CRYPTREC Report 2014

平成 27 年 3 月

独立行政法人 情報処理推進機構

独立行政法人 情報通信研究機構

「暗号技術活用委員会報告」

目次

はじめに

本報告書は、総務省及び経済産業省が主催する暗号技術検討会の下に設置され、独立行政法人情報処理推進機構及び独立行政法人情報通信研究機構によって共同で運営されている暗号技術活用委員会の 2014 年度の活動を報告するものである。

暗号技術活用委員会は、セキュリティ対策の推進、暗号技術の利用促進及び産業化を中心とした暗号利用に関する検討課題を主に担当する委員会として 2013 年度に設置された。

本委員会では、2013 年度に続き、暗号の普及促進・セキュリティ産業の競争力強化に係る検討、暗号技術の利用状況に係る調査及び必要な対策の検討、暗号政策の中長期的視点からの取組の検討（暗号人材育成等）など、従来から重要な検討課題として度々指摘はなされていたものの、我が国において本格的には検討がなされていなかった項目について審議し、「暗号普及促進・セキュリティ産業の競争力強化に向けた課題分析と見解」として取りまとめた。課題とした事項に関し今後解決策を検討するためには、CRYPTREC 外の組織との連携等も含めて考慮しなければならない点が多い。そこでそのような検討の際に有用と思われる視点について、検討を実施する体制についても含めて、留意点としてまとめている。

また、本委員会では、運用ガイドラインワーキンググループを設け、SSL/TLS 通信の安全性と可用性（相互接続性）のバランスをとった適用のために有用な事項を検討し、「SSL/TLS 暗号設定ガイドライン」として取りまとめて、公開した。このガイドラインは具体的な製品の設定方法に加えチェックリストも用意し、サーバ構築者やサーバ管理者にとって非常に使いやすいものとすることができた。

本委員会はまた、標準化推進ワーキンググループを設け、暗号アルゴリズム提案を実施または検討している企業・機関にとって有益な情報を取りまとめ、「暗号技術参照関係の俯瞰図」及び「標準化提案における交渉ノウハウ・課題及び参考情報」を作成した。暗号技術の提案に関して標準化活動の横展開を議論する場が存在しない状況下で始めた手探りの作業であったが、本課題に対する第一歩を踏み出せたのではないかと考えている。

今年度の活動成果が、今後の暗号の普及促進・セキュリティ産業の競争力強化や、より安全な情報化社会の実現に役立つことを期待している。

末筆ではあるが、本活動に様々な形でご協力下さった委員の皆様、関係者の皆様に対して深く謝意を表する次第である。

暗号技術活用委員会 委員長　松本 勉

本報告書の利用にあたって

本報告書の想定読者は、一般的な情報セキュリティの基礎知識を有している方である。例えば、電子署名や GPKI[1] システム等、暗号関連の電子政府関連システムに関係する業務に従事している方などを想定している。しかしながら、個別テーマの調査報告等については、ある程度の暗号技術の知識を備えていることが望まれる。

本報告書は、第 1 章には 2014 年度の暗号技術活用委員会の活動内容と成果概要を記述した。第 2 章には暗号技術活用委員会での 2 年間の検討結果となる「暗号普及促進・セキュリティ産業の競争力強化に向けた課題分析と見解」をまとめた。

Appendix には、運用ガイドライン WG の成果である「SSL/TLS 暗号設定ガイドライン」、標準化推進 WG の成果である「暗号技術参照関係の俯瞰図」と「標準化提案における交渉ノウハウ・課題及び参考情報」をそれぞれ掲載した。

2013 年度以前の CRYPTREC Report は、CRYPTREC 事務局（総務省、経済産業省、独立行政法人情報通信研究機構、及び独立行政法人情報処理推進機構）が共同で運営する下記の Web サイトから参照できる。

http://www.cryptrec.go.jp/report.html

本報告書ならびに上記 Web サイトから入手した CRYPTREC 活動に関する情報の利用に起因して生じた不利益や問題について、本委員会及び事務局は一切責任をもっていない。

本報告書に対するご意見、お問い合わせは、CRYPTREC 事務局までご連絡いただけると幸いである。

【問合せ先】 info@cryptrec.go.jp

[1] GPKI：Government Public Key Infrastructure（政府認証基盤）

委員会構成

暗号技術活用委員会（以下「活用委員会」）は、図 1 に示すように、総務省と経済産業省が共同で共催する暗号技術検討会の下に設置され、独立行政法人情報処理推進機構（IPA）と国立研究開発法人情報通信研究機構（NICT）が共同運営している。

活用委員会は、電子政府の安全性及び信頼性を確保し国民が安心して電子政府を利用できる環境を整備するため、セキュリティ対策の推進、暗号技術の利用促進及び産業化を中心とした暗号利用に関する検討課題を主に担当する委員会として、2013 年度に新たに設置された。具体的には、2012 年度にあった暗号運用委員会の担当業務の大半を引き継ぐ形で、暗号の普及促進・セキュリティ産業の競争力強化に係る検討、暗号技術の利用状況に係る調査及び必要な対策の検討、暗号政策の中長期的視点からの取組の検討（暗号人材育成等）などを実施する。

活用委員会と連携して活動する「暗号技術評価委員会」も、活用委員会と同様、暗号技術検討会の下に設置され、IPA と NICT が共同で運営している。

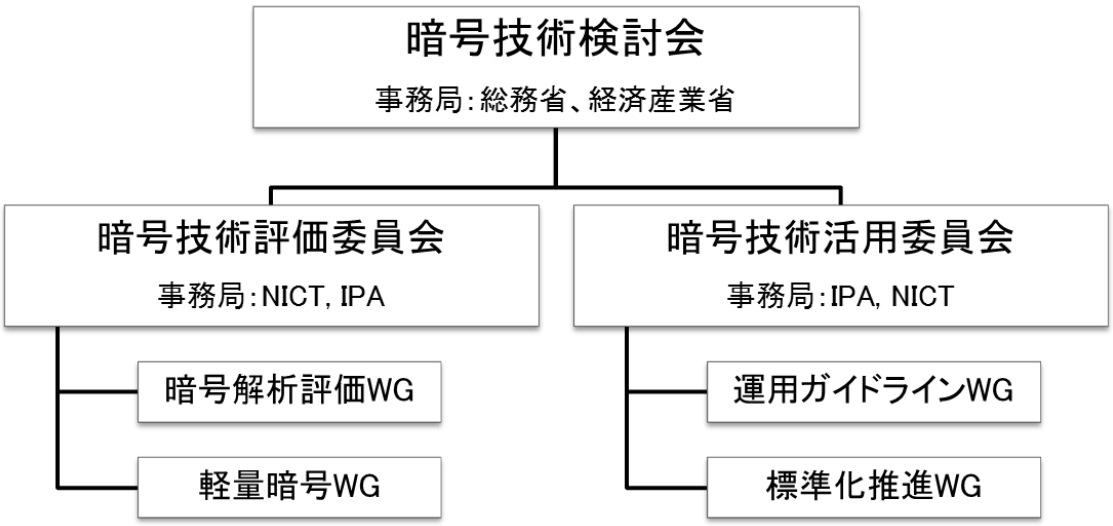


図 1　2014 年度の CRYPTREC の体制

委員名簿

暗号技術活用委員会

委員長	松本 勉	国立大学法人横浜国立大学 大学院 教授
委員	上原 哲太郎	立命館大学 教授
委員	遠藤 直樹	東芝ソリューション株式会社 技監
委員	川村 亨	日本電信電話株式会社 担当部長（チーフプロデューサ）
委員	菊池 浩明	明治大学 教授
委員	鈴木 雅貴	日本銀行 主査
委員	髙木 繁	株式会社三菱東京 UFJ 銀行 次長
委員	角尾 幸保	日本電気株式会社 主席技術主幹
委員	手塚 悟	東京工科大学 教授
委員	松井 充	三菱電機株式会社 技師長
委員	満塩 尚史	内閣官房 政府 CIO 補佐官
委員	八束 啓文	EMC ジャパン株式会社 部長
委員	山口 利恵	国立大学法人東京大学 特任准教授
委員	山田 勉	株式会社日立製作所 ユニットリーダ（主任研究員）
委員	山本 隆一	国立大学法人東京大学 特任准教授

運用ガイドラインワーキンググループ

主査	菊池 浩明	明治大学 教授
委員	阿部 貴	株式会社シマンテック マネージャー
委員	漆嶌 賢二	富士ゼロックス株式会社 マネージャー
委員	及川 卓也	グーグル株式会社 シニアエンジニアリングマネージャー
委員	加藤 誠	一般社団法人 Mozilla Japan テクニカルアドバイザ
委員	佐藤 直之	株式会社イノベーションプラス Director
委員	島岡 政基	セコム株式会社 主任研究員
委員	須賀 祐治	株式会社インターネットイニシアティブ シニアエンジニア
委員	高木 浩光	独立行政法人産業技術総合研究所 主任研究員
委員	村木 由梨香	日本マイクロソフト株式会社 セキュリティプログラムマネージャ
委員	山口 利恵	国立大学法人東京大学 特任准教授

標準化推進ワーキンググループ

主査	渡辺 創	独立行政法人産業技術総合研究所 研究グループ長
委員	江原 正規	東京工科大学 研究員

委員	河野 誠一	レノボ・ジャパン株式会社 主管研究員
委員	木村 泰司	一般社団法人日本ネットワークインフォメーションセンター
委員	坂根 昌一	シスコシステムズ合同会社
委員	佐藤 雅史	セコム株式会社 主務研究員
委員	武部 達明	横河電機株式会社 マネージャ
委員	廣川 勝久	ISO/IEC JTC 1/SC 17 国内委員長
委員	真島 恵吾	NHK 放送技術研究所 上級研究員
委員	真野 浩	コーデンテクノインフォ株式会社 代表取締役
委員	茗原 秀幸	三菱電機株式会社 担当課長

オブザーバー

大川 伸也	内閣官房内閣サイバーセキュリティセンター
石原 潤二	内閣官房内閣サイバーセキュリティセンター
森安 隆	内閣官房内閣サイバーセキュリティセンター
岡野 孝子	警察大学校
佐藤 健太	総務省 行政管理局[2014 年 12 月まで]
加藤 彰浩	総務省 行政管理局[2015 年 1 月から]
道喜 莉衣奈	総務省 行政管理局
白倉 侑奈	総務省 行政管理局
筒井 邦弘	総務省 情報流通行政局
近藤 直光	総務省 情報流通行政局
中村 一成	総務省 情報流通行政局
岩永 敏明	経済産業省 産業技術環境局
中谷 順一	経済産業省 商務情報政策局[2014 年 7 月まで]
中野 辰実	経済産業省 商務情報政策局[2014 年 8 月から]
室井 佳子	経済産業省 商務情報政策局
谷口 晋一	防衛省 運用企画局

事務局

独立行政法人 情報処理推進機構（伊藤毅志、近澤武、小暮淳、大熊建司、神田雅透、稲垣詔喬、吉川法子[2015 年 1 月まで]、加藤久美[2015 年 2 月から]）

独立行政法人 情報通信研究機構（平和昌、沼田文彦[2014 年 7 月まで]、中澤忠輝[2014 年 8 月から]、盛合志帆、野島良、大久保美也子、黒川貴司、金森祥子）

第 1 章　2014 年度の活動内容と成果概要

1.1　活動内容

暗号技術活用委員会では、今後の暗号に関する様々な課題解決に向けた政策立案等を行う際に役立てるために、2013 年度に引き続いて、以下の項目について検討を実施し、報告書に取りまとめることとなっていた。

① 暗号の普及促進・セキュリティ産業の競争力強化に係る検討

2013 年度と 2014 年度の 2 年間をかけて、「暗号政策上の課題の構造」や「暗号と産業競争力の関連性」など暗号の普及促進・セキュリティ産業の競争力強化に向けた具体的な課題分析や解決策の検討を取り扱う。

2014 年度は、2013 年度に引き続いて、議論を行ううえで有用な基礎データの収集を上期も継続して実施する。下期には、2013 年度及び 2014 年度上期に収集したデータをもとに、暗号の普及促進・セキュリティ産業の競争力強化に向けた具体的な課題分析や解決策の検討を実施し、報告書に取りまとめる。

② 暗号政策の中長期的視点からの取組の検討

上記の「暗号の普及促進・セキュリティ産業の競争力強化に係る検討」のなかで、様々なシステムを安全に動かしていくための暗号に関連する人材育成についても一緒に検討していくことにより、CRYPTREC として取り組むべき課題を明らかにし、報告書に取りまとめる。

③ 標準化推進

2013 年度の成果を踏まえ、今後、様々な組織が日本からの暗号アルゴリズムの提案を行う場合に、その成果が効果的に得られるようにするための、有望な標準化提案先の選定、当面必要とされる稼働見積もりや交渉方法、提案活動における課題等を、標準化推進 WG にて引き続き検討し、報告書に取りまとめる。

④ 運用ガイドライン作成

2013 年度にドラフト版を完成させた「SSL/TLS サーバ構築ガイドライン」について、引き続き運用ガイドライン WG にて作業を行い、成果物を暗号技術検討会に報告する。

1.2　今年度の委員会の開催状況

2014 年度暗号技術活用委員会は 3 回開催された。各回会合の概要は表 1 のとおり。

表 1　2014 年度暗号技術活用委員会概要

回	開催日	議案
―	メール審議	● WG 活動計画案の審議・承認
第 1 回	2014 年 10 月 30 日	● 活用委員会活動計画の確認 ● RC4 の注釈について ● 「暗号利用環境に関する動向調査」紹介 ● 最終報告書とりまとめに向けた論点整理 ● 各ワーキンググループからの報告・審議
第 2 回	2015 年 1 月 26 日	● RC4 の注釈について ● 標準化推進 WG からの報告・審議 ● SSL/TLS サーバ構築ガイドラインの審議 ● 最終報告書内容についての中間審議
第 3 回	2015 年 3 月 10 日	● 各ワーキンググループからの活動報告・審議 ● 課題解決に向けた分析結果・対策を取りまとめた最終報告書の審議

1.3　成果概要

1.3.1　暗号技術活用委員会の成果概要

暗号技術活用委員会では、2013 年度と 2014 年度の 2 年間をかけて、「暗号政策上の課題の構造」や「暗号と産業競争力の関連性」など、暗号アルゴリズムの普及促進・セキュリティ産業の競争力強化に向けた具体的な課題分析や解決策の検討を実施した。

2 年間の検討結果をもとに、各々の課題分析等の結果を第 2 章に取りまとめた。第 2 章の項目は以下のとおりである。

- ヒアリング調査結果
- 文献調査結果
- 暗号技術活用委員会での議論概要
- 今後の検討にあたっての留意点

また、RC4 の現行の注釈「128-bit RC4 は、SSL(TLS1.0 以上)に限定して利用すること」に対しては、暗号技術検討会事務局より要請された「CRYPTREC 暗号リストにおける RC4 の注釈について」の変更案の審議を行い、暗号技術活用委員会としては、以下の通りの活用委員会案を提案することとなった。

＜理由＞

- 早期に RC4 からの移行を進めることが好ましく、より明確に移行を促したほうがよい
- 暗号技術検討会から提示された変更案では、RC4 の利用可能範囲がどのように変化したのかが明確ではないため、「今後は極力利用すべきでない」という変更意図を明確化すべき

＜活用委員会案＞

「互換性維持のために継続利用をこれまで容認してきたが、今後は極力利用すべきでない。SSL/TLS での利用を含め、電子政府推奨暗号リストに記載された暗号技術への移行を速やかに検討すること。」

1.3.2 運用ガイドライン WG 概要報告

運用ガイドライン WG は、2015 年 3 月時点における、SSL/TLS 通信での安全性と可用性（相互接続性）のバランスを踏まえた暗号設定方法を、「SSL/TLS 暗号設定ガイドライン」と名称変更のうえ、ガイドラインとして取りまとめた。Appendix A に「SSL/TLS 暗号設定ガイドライン」を添付する。

本ガイドラインの主な対象読者は、主に SSL/TLS サーバを実際に構築するにあたって具体的な設定を行うサーバ構築者、実際のサーバ管理やサービス提供に責任を持つことになるサーバ管理者、並びに SSL/TLS サーバの構築を発注するシステム担当者としている。

本ガイドラインは 9 章で構成されており、章立ては以下のとおりである。

2 章では本ガイドラインを理解するうえで助けとなる技術的な基礎知識をまとめている。

3 章では、SSL/TLS サーバに要求される設定基準の概要について説明しており、4 章から 6 章で実現すべき要求設定の考え方を示している。

4 章から 6 章では、3 章で定めた設定基準に基づき、具体的な SSL/TLS サーバの要求設定について示している。

第 7 章では、SSL/TLS をより安全に使うために考えておくべきことをまとめている。

第 8 章は、クライアントの一つであるブラウザの設定に関する事項を説明しており、ブラウザの利用者に対して啓発するべき事項を取り上げている。

第 9 章は、そのほかのトピックとして、SSL/TLS を用いたリモートアクセス技術（“SSL-VPN”とも言われる）について記載している。

巻末には、4 章から 6 章までの設定状況を確認するためのチェックリストや、個別製品での具体的な設定方法例も記載している。

表 2　安全性と相互接続性との比較

設定基準	概要	安全性	相互接続性の確保
高セキュリティ型	扱う情報が漏えいした際、組織の運営や資産、個人の資産やプライバシー等に致命的または壊滅的な悪影響を及ぼすと予想される情報を極めて高い安全性を確保するSSL/TLSで通信するような場合に採用する設定基準 **※とりわけ高い安全性を必要とする明確な理由があるケースを対象としており、非常に高度で限定的な使い方をする場合の設定基準である。一般的な利用形態で使うことは想定していない**	本ガイドラインの公開時点において、標準的な水準を大きく上回る高い安全性水準を達成	最近提供され始めたバージョンのOSやブラウザが搭載されているPC、スマートフォンでなければ接続できない可能性が高い。 また、PC、スマートフォン以外では、最新の機器であっても一部の機器について接続できない可能性がある
推奨セキュリティ型	扱う情報が漏えいした際、組織の運営や資産、個人の資産やプライバシー等に何らかの悪影響を及ぼすと予想される情報を、安全性確保と利便性実現をバランスさせてSSL/TLSでの通信を行うための標準的な設定基準 **※ほぼすべての一般的な利用形態で使うことを想定している**	本ガイドラインの公開時点における標準的な安全性水準を実現	本ガイドラインで対象とするブラウザが搭載されているPC、スマートフォン等では問題なく相互接続性を確保できる。 バージョンが古いOSやブラウザ、一部の古い機器（フィーチャーフォンやゲーム機等）については接続できない可能性がある。
セキュリティ例外型	脆弱なプロトコルバージョンや暗号が使われるリスクを受容したうえで、安全性よりも相互接続性に対する要求をやむなく優先させてSSL/TLSでの通信を行う場合に許容しうる最低限度の設定基準 **※基本設定型への早期移行を前提として、暫定的に利用継続するケースを想定している**	推奨セキュリティ型への移行完了までの短期的な利用を前提に、本ガイドラインの公開時点において許容可能な最低の安全性水準を満たす	最新ではないフィーチャーフォンやゲーム機などを含めた、ほとんどのすべての機器について相互接続性を確保できる

3 章から 6 章が本ガイドラインの最大の特長ともいえ、「暗号技術以外の様々な利用上の判断材料も加味した合理的な根拠」を重視して現実的な利用方法を目指している。具体的には、実現すべき安全性と必要となる相互接続性とのトレードオフを考慮する観点から、安全性と可用性を踏まえたうえで設定すべき「要求事項」として 3 つの設定基準（表 2 参照）を提示している。

なお、7 章から 9 章は「情報提供」の位置づけとして記載している。

【開催日程】

第 1 回　2014 年 10 月 17 日
第 2 回　2014 年 12 月 16 日
第 3 回　2015 年　2 月 25 日

1.3.3　標準化推進 WG 概要報告

標準化推進 WG は、標準化機関に暗号アルゴリズム提案を検討している企業・機関にとって有益な情報について、2013 年度の成果を踏まえて以下のような議論を行い、「暗号技術参照関係の俯瞰図」と「標準化提案における交渉ノウハウ・課題及び参考情報」として取りまとめた。

本 WG では、暗号技術の提案に関して標準化活動の横展開を議論する場がなかった状況下の中、ファーストステップの作業として WG 委員の知見を集約して標準化活動に関する俯瞰図やノウハウをどのように取りまとめていくのがよいかを検討し、その方針に基づいた俯瞰図やノウハウを初めて取りまとめた。ファーストステップということで、俯瞰図の作成方法、ノウハウのとりまとめ方法も十分に固まったものではなく、また十分な網羅性を持っているわけではないので、今後の作業を進めるうえでのまとめ方のサンプル例として利用されたい。

課題として、網羅性の拡充をどのように進めるか、どのような知見を集めるべきか、俯瞰図の作成方法やメンテナンスをどのように行っていくか、ノウハウ・知見のメンテナンスをどのように行っていくか、アクティビティの結果をどのように展開するか、といった多くの点が残っている。今後の活動では、これらの課題をどのように解決するのかを踏まえて、どのようなやり方がよいかを見直して進めていくことが期待される。

(1) 暗号技術提案に当たっての俯瞰図の取りまとめ

今後暗号技術を提案する人が提案先を選定するために、参考となるように規格の参照関係について、「暗号技術参照関係の俯瞰図」（以下、俯瞰図）を作成した。主に今後暗号技術を提案する人が見ることを想定している。対象とした規格は、原則委員が関与している標準化団体の規格であるが、一部、暗号の標準化に影響の強い NIST、ANSI、ITU 等の規格も含む

こととした。

俯瞰図を作成することにより、暗号技術がどの規格で仕様として規定され、利用される技術がどの規格にて選ばれ、応用先としてどの規格に参照されているかについての現状を整理した。

(2) 暗号技術提案にあたっての交渉ノウハウ・課題等の整理

様々な標準化機関に対する日本提案の暗号アルゴリズム標準化を横断的に支援するため、標準化提案の際に知っていると、より提案が効率的に行えるようなノウハウや、標準化団体における基本的な情報、標準化活動における課題等について整理を行った。基本的な情報の中には、「提案できるタイミング」等、提案できる規格を探すために役立つ情報や、会議の年回数や電話会議の情報等のように稼動見積りの参考になる情報を含んでいる。

情報の整理の際に、まず団体間に共通する項目についてまとめることにより、標準化活動一般に利用できるような情報をとりまとめた。加えて、団体毎においても特有の情報をとりまとめた。

【開催日程】

第 1 回　2014 年 10 月 15 日
第 2 回　2014 年 12 月 11 日
第 3 回　2015 年　2 月 23 日

1.4　CRYPTREC シンポジウム 2015

【プログラムの概要】

日　時　　：2015 年 3 月 20 日（金）10：00～15：50
場　所　　：一橋大学一橋講堂
主　催　　：独立行政法人情報通信研究機構、独立行政法人情報処理推進機構
共　催　　：総務省、経済産業省
参加人数　：163 名
プログラム：表 3 のとおり

表 3　プログラム

時間	内容	
10:00	開会挨拶	情報処理推進機構　立石譲二　理事
	総務省挨拶・経済産業省挨拶	総務省・経済産業省
10:15	CRYPTREC 活動紹介	暗号技術検討会事務局
10:30	暗号技術評価委員会報告	今井秀樹　委員長　（東京大学　名誉教授）
10:45	暗号解析評価 WG 報告	高木剛　主査　（九州大学　教授）
11:05	軽量暗号 WG 報告	本間尚文　主査　（東北大学　准教授）
11:25	招待講演① プロトコルの形式検証と脆弱性発見の現実 - Case of CCS Injection -	林達也　様　（(株)レピダム　代表取締役）
12:10	昼休み	
13:40	暗号技術活用委員会報告	松本勉　委員長　（横浜国立大学　教授）
13:55	運用ガイドライン WG 報告	菊池浩明　主査　（明治大学　教授）
14:15	標準化推進 WG 報告	渡辺創　主査　（産業技術総合研究所　研究グループ長）
14:35	休憩	
15:00	招待講演② ISP から見た「暗号技術に期待したいこと・期待していないこと」	須賀祐治　様　（(株)インターネットイニシアティブ　シニアエンジニア）
15:45	閉会挨拶	情報通信研究機構　今瀬真　理事

第 2 章　暗号普及促進・セキュリティ産業の競争力強化に向けた課題分析と見解

2012 年度に改定した「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）」では、「安全性」及び「実装性」の観点に加え、「製品化、利用実績等」といった様々な視点で検討され、「電子政府推奨暗号リスト」、「推奨候補暗号リスト」、「運用監視暗号リスト」の 3 つのリストから構成される。この CRYPTREC 暗号リストの策定により、同リストに掲載されている暗号アルゴリズムの普及が促進し、ひいては日本のセキュリティ産業の競争力強化につながることが期待されている。

しかしながら、現実には「優れた暗号アルゴリズムがセキュリティ産業の競争力強化に直接的に繋がる」という関連性について、2012 年度の CRYPTREC のアクティビティである暗号運用委員会の委員ならびに CRYPTREC シンポジウム 2013 でのパネリストから極めて懐疑的な意見が多数出された。また、2012 年度の暗号技術の利用状況に係る調査結果からは、旧電子政府推奨暗号リスト策定から 10 年経過していたにもかかわらず、同リストに掲載されていた国産の暗号アルゴリズムの普及がほとんど進んでいない実態も明らかとなった。

そこで、我が国の暗号政策に係る中長期の視野に立って課題に引き続き取り組むため、暗号技術活用委員会において、「暗号政策上の課題の構造」や「暗号と産業競争力の関連性」など、暗号アルゴリズムの普及促進・セキュリティ産業の競争力強化に向けた具体的な課題分析等を行った。本章では、その結果を以下の通り取りまとめる。

- ヒアリング調査結果
- 文献調査結果
- 暗号技術活用委員会での議論概要
- 今後の検討にあたっての留意点

2.1　ヒアリング調査結果

2.1.1　ヒアリング調査の概要

「暗号政策上の課題の構造」や「暗号と産業競争力の関連性」などの課題に対する分析を行うにあたって幅広く現況を俯瞰することを目的として、ヒアリング（アンケート形式を含む）を 2013 年度下期から 2014 年度上期にかけて実施した。

ヒアリング先及びヒアリング項目の概要は以下のとおりである。

【ヒアリング先】

カテゴリ	対象
政府関係	政府 CIO
	X 省
	Y 省
業界団体	S 団体
	T 団体
暗号ライブラリ製造ベンダ	A 社
	B 社
	C 社
暗号製品製造ベンダ	D 社
	E 社
	F 社
セキュリティ製品製造ベンダ	G 社
	H 社
	I 社

【ヒアリング概要】

主な項目		概要
製品と暗号アルゴリズムとの関連性に関する事項	暗号アルゴリズムの選択に関連する事項	● 担当業務において、暗号を利用したり利用するように指示・取りまとめをした場面があったか ● どのような観点で利用する暗号アルゴリズムを決めているか ● 暗号アルゴリズムの選択に関してどのようなニーズがあるか ● 電子政府推奨暗号リストを活用しているか
	製品市場に関連する事項	● 製品市場（暗号ライブラリ・暗号製品・セキュリティ製品）はどのように変化しているか
国産暗号アルゴリズムの利用（普及阻害要因）に関連する事項		● 国産暗号アルゴリズムを利用しようと考えたことがあるか ● 国産暗号アルゴリズムを利用しようと考えたとき実際に大きな支障なく利用できたか
人材育成に関連する事項		● 暗号アルゴリズムの選択等に対する目利き人材としてどのような人材が必要か

2.1.2 ヒアリング調査の結果概要

【製品と暗号アルゴリズムとの関連性に関する事項】

実施したヒアリング結果に基づき、「A-1) 暗号アルゴリズムとセキュリティ製品との関係」と「A-2) 暗号アルゴリズムについての民間顧客からのニーズ」とに分けて整理した結果を以下に示す。

A-1) 暗号アルゴリズムとセキュリティ製品との関係

① セキュリティ製品の視点からみる暗号アルゴリズムの選択に関する現状について

一般的なベンダは、暗号ライブラリを使う際に、暗号機能を利用するための入出力インタフェースの仕様は理解していても、暗号アルゴリズムそのものはブラックボックスとして使っているのが現実である。また、暗号アルゴリズム自身の安全性だけでなく、実装難度が低く実装しやすいかとか、実用化のスケジュールとかといったことも含めて検討することになる。例えば、以下のような指摘があった。

ア) オープンになっている暗号アルゴリズムのなかからある水準以上のものを選べば、どれを選んでもセキュリティ製品からみて問題となるような技術的な差異は事実上ない。

イ) システムベンダは、パッケージベンダが作る（暗号以外の機能も様々に含んだ）パッケージライブラリを使ってシステムを構成していくので、システムベンダがどのパッケージライブラリを採用するか、そのパッケージライブラリがどんな下位の暗号ライブラリで構成されているかによって結局使える暗号アルゴリズムが絞られていく。その過程の中で、たいていの暗号アルゴリズムは滑り落ちて、AES くらいしか残っていないという状態になる。

ウ) 暗号アルゴリズムの選定では、安全性が優れているからというだけでなくて、利用実績があってこなれているものの方が、実装しやすく、当然コスト面も抑えられる。

エ) 国際的に販売するセキュリティ製品では、世界的に通用する暗号アルゴリズムを基本的に使うことになるので、国際標準化された暗号アルゴリズムしか使わない。

② ビジネスとしての暗号ライブラリ市場の成長鈍化について

暗号アルゴリズムの主な実装先として想定されているのは暗号ライブラリであるが、ヒアリングの結果からは、以前とは異なり暗号ライブラリ市場がビジネスとしては成立しにくくなっているのが現実である。例えば、以下のような指摘があった。

ア) ソフトウェアでは、OS やオープンソースに搭載されている暗号機能が使われ

るようになってきている。暗号ライブラリの利用先は、主にデバイス向け、特に複合機に移行してきている。

イ）暗号ライブラリを別途組み込んでアプリケーションを作り込むのは手間がかかるうえ、暗号ライブラリのメンテナンスも負担となるため、初めから搭載されている暗号機能を利用するほうが好まれる。

ウ）結果として、暗号ライブラリ市場はビジネスとしてほとんど成り立たず、現在では暗号ライブラリの研究開発を行っている国内メーカはほとんどないと考えられる。

なお、IPA「暗号利用環境に関する動向調査[2]」報告書でも、暗号ライブラリ市場の成長は 2008 年頃に止まり、現在横ばいになっていることが指摘されている。

③ 機能単体型セキュリティ製品市場の縮小について

情報セキュリティ製品の市場が拡大を続ける一方で、現在では様々なセキュリティ機能が搭載された統合型セキュリティ製品が主流であり、機能単体型セキュリティ製品の市場は横ばいまたは縮小の傾向になると予想される。例えば、ヒアリングの結果でも、以下のような指摘があった。

ア）暗号機能単体型のセキュリティ製品(VPN 等)の市場の伸びしろは大きくない。

イ）10 年位前には VPN 製品が単体で売れた時代もあったが、現在はネットワーク製品に組み込まれており、IPsec や SSL-VPN の製品単体では売れない。インフラの機能の一部として考えられている。

なお、上記のことは、IPA「暗号利用環境に関する動向調査」報告書でも指摘されている。

A-2) 暗号アルゴリズムについての民間顧客からのニーズ

④ 暗号アルゴリズムの違いは製品レベルでの差別化要因にならない

暗号アルゴリズムの違いは製品やシステムの購入に影響を与えるような差別化要因にはならず、採用している暗号アルゴリズムが理由で製品やシステムの購入が決まるケースはないのが現実である。例えば、以下のような指摘があった。

ア）現状では AES の安全性に問題がないため、暗号強度の違いが採用する暗号アルゴリズムの決め手とはならず、AES 以外の暗号アルゴリズムを採用しても製品の特長にはならない。それよりもユーザは柔軟性や効率等の使いやすさで製品を選択する。

[2] http://www.ipa.go.jp/security/products/products.html

イ）ユーザ側に暗号アルゴリズムを変えたいというニーズはない。

⑤ 日本でセキュリティ認証製品を出すモチベーションは高くない

セキュリティ認証を取得するもともとの目的は、製品に付加価値をつけるためではなく、国内外での調達要件に対応するためであるとの指摘があった。具体的には、以下のような指摘があった。

ア）グローバルにデバイスを展開するメーカは、米国での調達（政府・金融）を考えると FIPS140-2 を含めた形で製品を提供するケースが増えている。

イ）日本ではセキュリティ認証製品の重要性があまり知られておらず、セキュリティ認証を取得しても、調達要件上の優位性を持たず、また製品の付加価値としても認識してもらえない。

【国産暗号アルゴリズムの利用に関連する事項】

実施したヒアリング結果に基づき、国産暗号アルゴリズムの利用に関連して、国産暗号アルゴリズムの普及阻害要因を整理した結果を以下に示す。

⑥ 技術優位性以外の優位性の不足

製品やシステムでの暗号アルゴリズムの採用基準は、あくまでも調達・設計要件を満たしているかどうかであって、技術優位性はたくさんある比較項目のなかの一つに過ぎない。例えば、以下のような指摘があった。

ア）部品としてどれだけ強いかということのほかに、今までとの継続性はどうか、国際標準化はどうか、利用実績はどうか、といった点を見ている。

イ）実装のためのコスト面も無視できない。

ウ）日本以外の国に製品展開できるかどうかが重要であり、製品化するうえで必要な国際標準化がされていることは必須である。

⑦ 圧倒的なシェアを持つデファクトスタンダードの存在

ビジネス上は、基本的には国際標準化されていて広く採用されている暗号アルゴリズムを使うのが大前提であり、国内市場であっても、国産暗号アルゴリズムかどうかはほとんど関係がない。例えば、以下のような指摘があった。

ア）暗号アルゴリズムは接続する相手先の製品へも組み込まれている必要があるが、国産暗号アルゴリズムでは、製品の種類が圧倒的に少ない状況が改善されない限り、いくら技術的に優れていても国産暗号アルゴリズムは普及しない。

イ）デバイスを日本で作っているならば、そこへ国産暗号アルゴリズムを採用できたかもしれないが、近年は OEM の競争力も落ちてきているため、難しいと思

われる。

ウ）暗号アルゴリズムの採用には前例が求められるため、リーダーシップをとる企業の先進事例として取り上げられ、そこへ追従する企業に展開する、という形の普及展開の方法であっても難しい。

エ）最終的には、国産暗号アルゴリズムが搭載された製品自体が海外で広く販売されるか、国内で販売している外資系企業の製品に勝つことが必要な状況となっている。

⑧ 国産暗号アルゴリズムの利用促進策として「政府機関の情報セキュリティ対策の統一基準群（政府統一基準群)」を活用することの困難性

政府統一基準群を使って省庁の導入から紐づく組織や企業へピラミッド型に展開する普及策が考えられるが、現在の政府統一基準群では安全かつ実装性に優れている「電子政府推奨暗号リストの利用」を指定しているため、国産暗号アルゴリズムだけを明示的に指定して調達を行うことはできない。例えば、以下のような指摘があった。

ア）「電子政府推奨暗号リストを利用」との記述しかないため、国産暗号アルゴリズムを採用する動機付けにはならず、実態的にはデファクトスタンダードの暗号アルゴリズムが採用されている汎用市販品が多くの政府調達のベースとなっている。そのため、セキュリティ製品を作る企業にとっては、国産暗号アルゴリズムを導入するきっかけにはほとんどならない。

イ）行政規格としては必要最小限の要求事項の大枠だけを決め、暗号アルゴリズム名などの具体的な方式まで事細かに決めているわけではないものも多い。その場合、決めていない部分や詳細化・具体化する部分は民間規格に委ねることになるため、国産暗号アルゴリズムの普及策として政府統一基準群や電子政府推奨暗号リストがどの程度活用できるかはわからない。

ウ）仮に政府統一基準群で国産暗号アルゴリズムの利用を規定したとしても、製品化が伴わない、政府統一基準群だけに頼るだけの施策では、基準がガラパゴス化する懸念がある。

【人材育成に関連する事項】

実施したヒアリング結果に基づき、人材育成に関連して整理した結果を以下に示す。

⑨ 経営的観点と技術力を併せ持った人材の不足

技術力ばかりに注目するのではなく、経営層やオピニオンリーダへのロビー活動等も含め、標準化や普及展開を行う上で重要な技術以外の視点での的確な展開戦略

の検討・実施する人材が不足している。例えば、以下のような指摘があった。

ア）日本の技術者は経営的側面の重要性を知らなさすぎる。例えば、調達や経営上のデシジョンメイクがどのように行われているかといったことを理解している人材が不足している。
イ）日本の暗号の技術力やクオリティは非常に高いが展開戦略がない。ロビー活動等も含め、技術以外の部分の視点があまりにも弱い。

⑩ 暗号アルゴリズムとシステム構築・運用との間をつなぐ人材の不足

システム構築・運用にあたって、暗号アルゴリズムを適切に利用するためのノウハウをもつ人材が不足しており、意図しない使われ方や誤った使われ方をしたために安全な暗号アルゴリズムを使っていてもシステムとしては脆弱であったり、問題発生時に適切な対処がされていなかったりといったことが少なからず発生している。

2.2 文献調査結果

2.2.1 日本における動向

【組織体制（所管官庁・法制度・権限等）】

日本では、サイバーセキュリティ戦略本部の事務局である NISC と、暗号技術評価である CRYPTREC プロジェクトを行っている経済産業省・総務省が、主に暗号関連の施策を担っている。

CRYPTREC が発足した 2001 年ごろは、国際的に厳格な輸出規制下で暗号アルゴリズムが管理されており、また ISO/IEC などの国際標準規格も定められていなかったため、国際的に広く使われる暗号アルゴリズム（いわゆるデファクト暗号アルゴリズム）がなかった。そのため、国内においても、様々な企業が暗号アルゴリズムを自ら開発・販売する状況になっていたが、その中には安全な暗号アルゴリズムであるかが疑わしいものも少なくなかった。

このような状況下において、安全な暗号アルゴリズムで電子政府システムを構築できるようにするため、総務省と経済産業省は、安全であると評価された暗号アルゴリズムを選定しリスト化する目的で CRYPTREC を発足させ、2003 年に最初の電子政府推奨暗号リストを取りまとめた。その後は、電子政府推奨暗号リストに記載された暗号アルゴリズムについて安全性低下などの問題（暗号危殆化）が起きていないかを監視しており、必要に応じて関係各所に注意喚起を行っている（例えば、NISC が策定した SHA-1 及び RSA1024 に係る移行指針は、CRYPTREC からの注意喚起が契機となって指針が策定された）。

一方、最近の 10 年間で、ISO/IEC などの国際標準規格の策定や、輸出規制の大幅緩

和とワッセナーアレンジメントへの移行などの要因により、ビジネスの世界では国際的に利用できるデファクト暗号アルゴリズムの集約が進んでいる。このような外部環境の変化も踏まえ、暗号アルゴリズムの危殆化及び移行対策等を含めた適切な暗号アルゴリズムの選択を支援するため、入手しやすさや導入コスト、相互運用性、普及度合い等の観点も取り入れて電子政府推奨暗号リストの見直しが行われ、2012 年度末に「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）」が公表された。

また、2013 年から 2014 年にかけて、日本ではサイバーセキュリティ対策に関わる体制の見直しが以下の通り行われた。

① サイバーセキュリティ対策が国家安全保障戦略の一部を担うことが明確化された。

- サイバーセキュリティ戦略（2013 年 6 月情報セキュリティ政策会議決定）
  リスクの深刻化の進展に対応した国家安全保障・危機管理・産業競争力強化等の観点からの取組みを強化
  - 統一基準群の改定
  - GSOC（政府機関情報セキュリティ横断監視・即応調整チーム）機能強化
  - 重要インフラの範囲拡大・行動計画見直し
  - 情報セキュリティ普及啓発プログラムの改訂
  - 人材育成プログラムの改訂
  - 研究開発戦略の見直し
  - 国際戦略の策定
  - NISC 機能強化（組織体制の見直し）
- 情報セキュリティ研究開発戦略（2014 年 7 月情報セキュリティ政策会議決定）
  サイバーセキュリティ戦略に基づき、情報セキュリティ研究開発戦略を改定
  - 情報セキュリティのコア技術の保持
    暗号等のコア技術の保持は、我が国の新規産業創出や安全保障等の観点から重要であり維持・強化

② 政府は、サイバーセキュリティ戦略・国家安全保障戦略・日本再興戦略に基づき、セキュリティの機能強化を図った。また、国会においても 2014 年にサイバーセキュリティ基本法が成立した。

- サイバーセキュリティ基本法（2014 年 11 月成立（議員立法）、2015 年 1 月施行）
  - 法令上、初めて「サイバーセキュリティ」が明記された
  - サイバーセキュリティ戦略本部を設置。IT 総合戦略本部配下の情報セキュリティ政策会議が担ってきた機能は、サイバーセキュリティ戦略本部が担

う

- サイバーセキュリティ戦略本部の所管事務は以下のものが規定されている
  1. サイバーセキュリティ戦略案の作成
  2. 政府機関等の防御施策評価（監査を含む）
  3. 重大事象の施策評価（原因究明調査を含む）
  4. 各府省の施策の総合調整（経費見積り方針の作成等を含む）
- 内閣官房情報セキュリティセンター（旧 NISC; National Information Security Center）を改組し、サイバーセキュリティ戦略本部の事務局として法令組織（内閣官房組織令）となる「内閣サイバーセキュリティセンター（新 NISC; National center of Incident readiness and Strategy for Cybersecurity）」を設置
- 新 NISC は 2015 年 1 月 9 日付で発足。所管事務は以下のものが規定されている
  1. GSOC に関する事務
  2. 原因究明調査に関する事務
  3. 監査等に関する事務
  4. サイバーセキュリティに関する企画・立案、総合調整

このように、2015 年以降、同法などに基づき、情報セキュリティに対する組織体制が大幅に刷新される計画である。

【暗号アルゴリズムの位置づけ】

暗号技術検討会 2011 年度報告書によれば、CRYPTREC 暗号リストに求める役割として「国際標準化・製品化促進の手段として電子政府推奨暗号リストを活用」する目標が掲げられている。その趣旨は、国産暗号アルゴリズムにおいては「米国政府標準暗号アルゴリズム以外の暗号アルゴリズムは国際標準化や規格化、製品化からも排除される流れが強まっている点を考慮し、提案暗号である国産暗号アルゴリズムに対する国としてのバックアップの明確化を検討」するように求めるものであった。

一方、政府のセキュリティ政策における暗号の位置づけが語られておらず、暗号をどう活用するのか不明確である。サイバーセキュリティ年次計画においても「政府機関における安全な暗号利用の推進」以外の記述はない。政府統一基準においても「電子政府推奨暗号リストを参照」との記述がある程度であり、暗号が使用可能な場合には電子政府推奨暗号リストの中から暗号アルゴリズムを選択して使わせることとしか確認できない。

また、情報セキュリティ研究開発戦略では「暗号等のコア技術の保持は、我が国の新規産業創出や安全保障等の観点から重要」とされているが、研究開発以外の政府調達や産業政策、国家安全保障、情報保全といった観点での具体的な施策においては、暗号普及策が取り扱われていないことが多い。

2.2.2 米国の IT セキュリティ

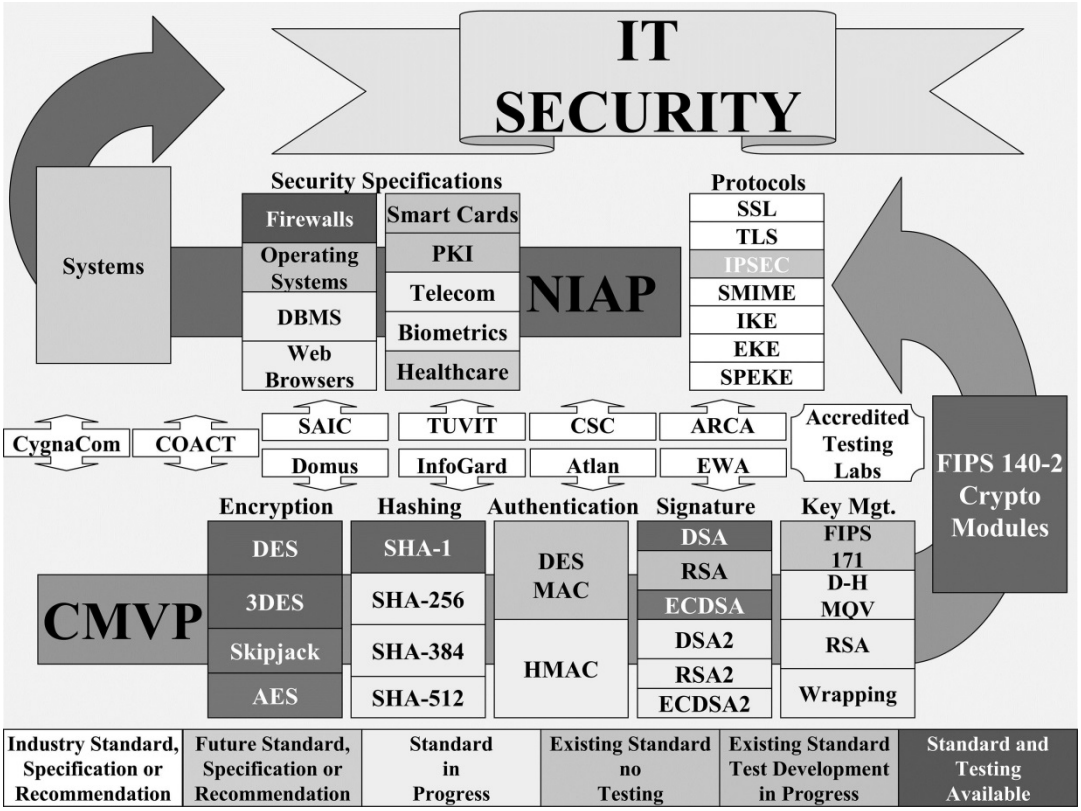
暗号アルゴリズムが IT セキュリティに寄与するまでには、目的や設計方針、想定する利用環境等といった製品を実現するために要求される考え方・思想が異なる複数の階層が存在する。これらの階層が上がる（システムに近づく）ほど、暗号アルゴリズム以外の要素の重要度がより高くなる。

米国では、CMVP Conference 2002 での資料からもわかるように、暗号アルゴリズムが製品・システムとして IT セキュリティに寄与するまでには要求される考え方・思想が異なる複数の階層が存在していることを 10 年以上前から認識しており、それぞれの階層で役割を分担しつつ、有機的に連携して暗号アルゴリズムから製品・システムのビジネスまでをつなげている。

その結果として、米国政府標準として定めた暗号アルゴリズムを採用した製品が生産される環境が整っている。具体的には、

ア）NIST や NSA の管理下に置いて、暗号アルゴリズム仕様、暗号モジュール、プロトコル、セキュリティ仕様、システムの 5 つの階層に分け、それぞれが有機的に連携するようにしている。特に、中間にあるプロトコルは国際的に影響力を持つ外部の産業標準を使う前提で、その前後を有機的に連携し、かつ標準化活動を直接支援することで、実態的にプロトコル標準においても強い影響力を与えるような形になっている

イ）実施したヒアリング結果でも、「暗号アルゴリズムを普及させるということと、暗号アルゴリズムでビジネスをすることは異なる。米国でも暗号アルゴリズムを標準化することで儲かっているわけではない。標準化されインフラ化された暗号アルゴリズムを実装した製品・システムの階層がビジネスになっている」との指摘があった



[出典] NIST, CMVP Status and FIPS 140-1&2, CMVP Conference 2002 Presentation

NIAP： セキュリティ認証製品の利用促進を通じて利用者のセキュリティ向上を図るための政府プログラム。 NSA と NIST により設立	システム	システムに必要な要件すべてを実装し、セキュリティ認証を受けたもの。この認証を受けたものが政府調達の対象となる（例：CC 認証、プロテクションプロファイル作成など）
	セキュリティ仕様	具体的なセキュリティ機能を実現するために必要となる仕様を規定するもの（例：ファイアウォール、OS、データベース、ブラウザ、バイオメトリクス、IC カード、ヘルスケアシステム等に必要なセキュリティ機能の規定など）
産業標準： 標準規格を策定する団体（IETF や IEEE 等）の活動を NIST や NSA が直接支援	プロトコル	基本的に外部の産業標準を流用する（例： IETF 標準プロトコル、IEEE 標準プロトコル、など）
CMVP： 暗号アルゴリズムを中心とした安全な暗号モジュールの提供を実現するための政府プログラム。 NIST が実施	暗号モジュール	暗号アルゴリズムを含め、実装された暗号モジュールが安全に使えるための必要な要件を実装し、セキュリティ認証を受けたもの（CMVP 認証）
	暗号アルゴリズム仕様	CMVP 認証の対象となる暗号アルゴリズムの仕様を規定したもの

2.2.3 IPA「暗号利用環境調査」報告書 － 海外動向

【組織体制（所管官庁・法制度・権限等）】

報告書では、暗号政策に関連する組織体制において、以下の点が指摘されている。

ア）欧米諸国などでは、暗号政策を国家安全保障もしくは情報保全と位置づけている国が多いうえ、暗号政策を議論しているのは、上位の組織体（国家安全保障会議、大統領府、閣議、セキュリティ戦略委員会など）となっている

イ）暗号政策として決められた目的を実現するための具体的な実務執行機関としての所管官庁及び権限が決められている。例えば、米国では、大統領令や OMB などが決定した政策方針に基づき、必要な標準・ガイドラインを作成したり、セキュリティ認証制度を運営したりする実務執行権限を NIST に与えている

ウ）暗号政策の遂行に当たっては、米国・ドイツ等は財務省などの予算権限を持つ省庁も直接関与しており、暗号政策の施策実行における財政面についても議論されているものと思われる

【暗号アルゴリズムの位置づけ】

報告書では、暗号アルゴリズムの位置づけについて、以下の点が指摘されている。

エ）欧米諸国の多くの国では、技術的な意味での暗号アルゴリズム単体のみに注目しているのではなく、あくまで国家安全保障や情報保全に関する文書の中での１つの構成要素として言及されている

オ）国家安全保障や情報保全のための高セキュリティ調達製品で利用する暗号アルゴリズムは、デファクトスタンダードの暗号アルゴリズムとは異なるものを指定している国（米・英・仏・露・中・韓など）もある

カ）政府の情報保全のために強力な暗号アルゴリズムを必要とする一方、国家安全保障・テロ対策の観点から暗号アルゴリズムの利用制限や司法権の行使による強制解除といった項目を含めて、暗号アルゴリズムの位置づけを決めている国（米・仏・露・中など）もある

【暗号アルゴリズムについての政府調達からのニーズ】

報告書では、米国以外でも、国家安全保障や情報保全などに関わる高セキュリティシステムや製品の政府調達においては、セキュリティ認証製品を使うように義務付けている国（英・仏・露・中・韓など）も多いことが指摘されている。例えば、デファクトスタンダードの暗号アルゴリズムが採用されている汎用市販品と、国家安全保障や情報保全のための高セキュリティ調達製品を明確に区別している国（英・仏・独・韓など）があるなど、具体的には以

下のような記述がある。

キ）取り扱う情報の重要性に基づいて、どの程度の安全性を持つ製品を調達させるかの政府調達基準を変えている国（英・仏・韓など）がある

ク）高セキュリティ調達製品では、何らかの製品認証（CC または CMVP 相当）を要求する場合、そこで利用される暗号アルゴリズムも明示的に決められている。なお、ここで利用される暗号アルゴリズムは非公開とされる場合もある（米・英・仏・独・露・中・韓など）

2.3 暗号技術活用委員会での議論概要

2.3.1 製品と暗号アルゴリズムとの関連性についての論点

製品と暗号アルゴリズムとの関連性について、「アルゴリズム、プロトコル、製品というレイヤのわけ方は NIST が 10 年以上前に作った考え方なので、最近の状況を踏まえて整理し直すのがいいと考える。特に、サービス（とりわけクラウドサービス）と暗号アルゴリズムの関係についても、製品と暗号アルゴリズムの関係と同じように体系として検討していく必要がある」との指摘が委員からあった。

また、「例えば、米国の認証系の標準では、ライバル企業間でも共通化する部分と独自技術として競争する部分が合意されていて、共通化部分では各社共通のモジュール化が行われている一方、日本の企業は各社が全てを独自技術で勝負しようとしており、本来は業界で共通化したほうがよい部分についてまでモジュール化ができてないために、結果として業界での技術が統一されていないようにみえる。特に暗号やセキュリティの分野で、企業間のうまい組み方ができない理由がどこにあるのかの分析が必要」との指摘が委員からあった。

その理由の一つには、例えば通信プロトコルでは標準に従わないと通信が行えないが、セキュリティ技術の標準化においては、ある技術を標準化したからといって他の技術が利用できないわけではないので、お互いに技術を共存させてもよいのではないか、との認識をもっている可能性が考えられる。

2.3.2 暗号アルゴリズムの位置づけについての論点

CRYPTREC において暗号アルゴリズムの安全性や利用方法については議論しているが、「わが国における暗号アルゴリズムの位置づけや戦略についての方針がはっきりしていないことを危惧するので、暗号アルゴリズムの位置づけや戦略について議論する場を体系として持っておいたほうがよい」との指摘が委員からあった。

2.3.3　政府主導の暗号アルゴリズムの標準化についての論点

企業が開発する技術を自ら標準化する理由の一つとして、他社企業へ特許ライセンスを許諾し、ライセンス料（ロイヤリティ）を得るというものがある。暗号アルゴリズムの分野においても、過去にはライセンス料の支払いを必要とするものも少なくなかった。

しかし、RSA のように特許期間が満了しライセンス料が不要になったり、AES や Camellia のように当初から無償ライセンス許諾をしたりするなど、現在、国際的に主流となっている暗号アルゴリズムのほとんどすべてがライセンス料無料（ロイヤリティフリー）で利用可能になっている。このため、最近の暗号アルゴリズムの標準化ではロイヤリティフリーを要求されることが一般的になった。

このような状況下では、企業が暗号アルゴリズムを自ら開発し、標準化を行うメリットやモチベーションは大きく削がれることになる。むしろ、どこの国の暗号アルゴリズムであれ、標準化されインフラ化された暗号アルゴリズムを採用して製品・システムを作れば十分ではないか、と考えることもできる。

ところが、世界的にみれば、暗号アルゴリズムの標準化を国家主導で進める国が少なからずある。

今後、暗号アルゴリズムの位置づけを検討するに当たっては、なぜ自国の暗号アルゴリズムの標準化を国家主導で進めている国があるのか、それによってどのような利益が当該国にあると考えればよいか、世界にどのような影響を与えることを期待しているのか、などを分析していく必要があるのではないかと思われる。

一つの仮説としては、政府と企業との win-win 構造を作り、うまく政府と企業が役割分担しつつ、それぞれの目的を達成するという意味で、当該国の政府が自国の暗号アルゴリズムの標準化を主導することにメリットを感じているということが考えられる。

具体的には、政府としては、

- 信頼できるかどうかわからない他国の暗号アルゴリズムを使わざるを得なくなる危険性を除去
- 自国の暗号アルゴリズムを搭載した製品がたくさん出回るようになれば、価格競争が働くため、結果として安く調達可能
- セキュリティ認証制度と合わせることで、信頼できない実装物（製品）が海外から入ってくるリスクを軽減

といったメリットが、企業としては、

- 自国政府という購入者が確実にいることで、安心して製品化に踏み切れる
- 自国政府から情報を先行して入手しやすい立場にあるので、他国に先行して製品を出せる

- 自国の暗号アルゴリズムが世界標準になるとわかっていれば、後で余分な暗号アルゴリズムを実装しなくて済むうえ、そのまま輸出もできるようになる

といったメリットがあると考えられる。

2.3.4 標準化活動に関連する論点

標準化活動について、標準化推進 WG から以下の項目についての指摘があった。

① 強い信頼関係に基づく人脈形成の重要性

標準化提案を受け入れてもらうためには、標準化に関わる他の「人」との関わりが必要不可欠である。周りの人たちとの協力関係を築き、ネゴシエーションしながら標準化を行うと提案がスムーズに受け入れられやすい。例えば、以下のような指摘があった。

ア）提案が受け入れられている理由は、様々な人たちとの人的なつながりや出来上がった信頼関係と、規格策定における協力において貸し借りをうまく行っているからである

イ）過去の審議経緯や利害関係等を十分に把握していると審議を進めるのに有利である。継続的に規格策定の場に関わっている人とコミュニケーションをとり、審議の経緯等を知っておくとよい

ウ）技術的な問題だけでなく、標準化会議に出席するメンバ（特に皆から一目置かれるキーパーソンとなるような人物）との信頼関係が提案交渉に大きく影響する

② 過去の経緯などを把握した継続的な標準化活動の重要性

欧米の場合、個人が標準化の仕事として長年参加しているので、企業を移っても所属企業名が変わるだけでその人物は引き続き参加するケースが多い。このような人物の場合、標準化作業についての過去の経緯などを熟知し、交渉ノウハウにも長けることが多いので、一目置かれる存在として強い発言力を持って優位に標準化作業を進めることができる。例えば、以下のような指摘があった。

エ）海外のコンサルタントは、継続的に規格策定の「現場」に関わっており、過去の審議経緯や利害関係者等を十分に把握しているため、審議がスムーズに進むことが多い

オ）日本の場合、企業として標準化団体に参加しており、担当者が異動してしまうと新たな担当者が割り当てられることが多いため、他のメンバたちから信頼を

得るようになるのに時間がかかる

上記の点について、暗号技術活用委員会としても検討した結果、「標準化活動を担当する人材の重要性」について、以下のとおり見解を取りまとめた。

- 日本では企業が標準化の旅費を出す等のサポートを行っているが、海外では標準化専門のコンサルタントが数多くいるように見受けられる。海外のコンサルタントは技術的に優れているとは限らないが、政治力があり、標準化を優位に進めることができるのは事実である。
- 海外のコンサルタント等と交渉を進めるうえでも、過去の経緯を知っていることや信頼関係が重要である。しかし、日本のように、人事異動等で担当者が交代するというのを続けているといつまでたっても信頼を成熟できないのは事実である。出張費等の資金援助など、担当者が継続的に標準化活動をしやすくなる仕組みが必要と考える。
- 欧米ともに、自らの優位性を活用できるやり方での標準化に力を入れている。具体的には、デファクト標準やフォーラム標準では、技術的に優れていることよりも、早く周囲の人を説得できる人が有利であるため、米国では標準化専門のコンサルタントを活用して、すばやくデファクト標準を作り上げていく手法を取る。また、欧州は国の数が多く賛成票が集められる優位性を踏まえて、各国投票で標準を決めるデジュール標準を推し進めていく手法を取る。日本も、標準化の進め方の枠組みといったものにも留意すべきである。
- 以前は日本が商品シェアを持っている業界が業界単位で標準化に持っていくことができたが、現状そのような業界があまりないことが懸念される。

2.3.5　人材育成に関連する論点

ヒアリング調査では「⑨ 経営的観点と技術力を併せ持った人材の不足」との指摘があったが、委員からは「経営的観点と技術力を併せ持った人材の不足よりも、経営的な決定権を持っている人が技術に対するケアをできていないことのほうが問題である。デシジョンメイクがうまくいっていない解決策として下から上に人材を育てるのは非常に時間がかかり、あまりにも回り道過ぎる」との指摘があった。

また、制御システムやIoT型サービスを始めとする様々な分野のプロジェクトに企業の暗号研究者が組み入れられ、本来の暗号研究が阻害されつつあり、企業による暗号研究の人材育成の困難になってきている結果、暗号アルゴリズムを評価できる人材が減ってきている懸念があるとの委員からの指摘があった。

2.4　今後の検討にあたっての留意点

2.1 節から 2.3 節までの結果を踏まえ、今後さらなる検討を行う際には、以下の点に留意して検討を行うことが望ましい。

I　暗号アルゴリズムの普及策を検討する場合には、暗号アルゴリズムのみでの議論でなく、プロトコルや製品、サービスレベルでの議論を図っていく必要があるが、プロトコル、製品、サービス以外の観点でのレベルが存在する可能性もあるため、どのようなレベルでの議論が適切かという観点も含めて議論をしていくことが望ましい。その際の留意点としては以下のとおり。

- 製品レベルの議論では、暗号アルゴリズムの実装先として「暗号ライブラリの開発」を期待することが困難になっている。
- ビジネスとして成立するのは製品レベルとなっており、プロトコルレベル以下の暗号アルゴリズムのみでの標準化・普及活動はビジネスとしては難しいため、一般に企業活動として主体的に行う暗号アルゴリズムの標準化・普及活動の対象は自社ビジネスの製品化に必要な範囲内にとどまる。

II　上記 I の議論と併せて、各社の自主技術として競争する部分と、各社が共通技術として共同でモジュール化する部分とを区別し、共通技術については各社が連携して活動する枠組みを作ることで各社の活動の効率化と製品市場の活性化を図る視点を取り入れることも考慮に値する。

III　サイバーセキュリティ基本法の制定を踏まえ、暗号に関して「こうあるべき、こうしていくべき」という戦略の部分をしっかりと議論して決めて実行していくヘッドクォーターが必要であり、新 NISC が発足したことを機にどこがヘッドクォーターになるのか、どこがどのような役割を担っているのかを整理することが望ましい。その際には、国産暗号アルゴリズムをどのように位置づけるかや、暗号による重要インフラや情報システムにおける安全性向上策を議論するための枠組みも併せて検討していくことも例として挙げられる。

IV　上記Ⅲの議論を受け、国産暗号アルゴリズムの普及策を検討する場合には、世界的にみれば暗号アルゴリズムの標準化を国家主導で進める国が少なからずあることを認識した上で検討することが望ましい。その際、技術優位性以外の優位性や項目が重要視されるといった暗号技術全般の特殊性を踏まえ、市場競争で国産暗号アルゴリズムの普及実現を図ることは相当困難であることを考慮する必要がある。

V　暗号アルゴリズムの標準化活動について検討する場合には、活動が長期にわたることを踏まえると、企業に任せ切るのではなく、実際に標準化活動を担当する人物が長期にわ

たって安定的に活動を継続できるような支援の在り方などを検討することが望ましい。例えば、その支援の一つとして、日本における標準化専門コンサルタントの育成について、その是非や実現可能性について検討することも一案である。

VI 人材育成を検討する場合には、CRYPTREC における暗号監視の維持のための人材育成という観点と、暗号に関する人材のステップアップを図る人材育成という観点は分けて検討することが望ましい。特に前者については、企業による暗号研究の人材育成が困難になってきていることを踏まえると、暗号監視の維持に必要な CRYPTREC での暗号評価作業や監視作業が継続できる体制・仕組みを検討することなどが考えられる。

VII スキルアップを図る人材育成の観点では、システム構築者・運用者、技術者、経営者のどれか一つに偏るのではなく、それぞれに対して育成方針を検討していくことが望ましい。

Appendix A
SSL/TLS 暗号設定ガイドライン

SSL/TLS 暗号設定ガイドライン

独立行政法人 情報処理推進機構
国立研究開発法人 情報通信研究機構

目次

1. はじめに

1.1 本書の内容及び位置付け

本ガイドラインは、2015 年 3 月時点における、SSL/TLS 通信での安全性と可用性（相互接続性）のバランスを踏まえた SSL/TLS サーバの設定方法を示すものである。

本ガイドラインは 9 章で構成されており、章立ては以下のとおりである。

2 章では、本ガイドラインを理解するうえで助けとなる技術的な基礎知識をまとめている。特に高度な内容は含んでおらず、SSL/TLS 及び暗号についての技術的な基礎知識を有している読者は本章を飛ばしてもらって構わない。

3 章では、SSL/TLS サーバに要求される設定基準の概要について説明しており、4 章から 6 章で実現すべき要求設定の考え方を示す。

4 章から 6 章では、3 章で定めた設定基準に基づき、具体的な SSL/TLS サーバの要求設定について示す。本章の内容は、安全性と可用性を踏まえたうえで設定すべき「要求事項」である。

第 7 章では、チェックリストの対象には含めていないが、SSL/TLS を安全に使うために考慮すべきことをまとめている。本章の内容は、「情報提供」の位置づけとして記載している。

第 8 章は、クライアントの一つであるブラウザの設定に関する事項を説明しており、ブラウザの利用者に対して啓発するべき事項を取り上げている。本章の内容は、第 7 章と同様、「情報提供」の位置づけのものである。

第 9 章は、そのほかのトピックとして、SSL/TLS を用いたリモートアクセス技術（“SSL-VPN”とも言われる）について記載している。本章の内容も「情報提供」の位置づけのものである。

3 章から 6 章が本ガイドラインの最大の特長ともいえ、「暗号技術以外の様々な利用上の判断材料も加味した合理的な根拠」を重視して現実的な利用方法を目指している。具体的には、実現すべき安全性と必要となる相互接続性とのトレードオフを考慮する観点から、安全性と可用性を踏まえたうえで設定すべき「要求設定項目」として 3 つの設定基準（「高セキュリティ型」「推奨セキュリティ型」「セキュリティ例外型」）を提示している。

Appendix には、4 章から 6 章までの設定状況を確認するためのチェックリストや、個別製品での具体的な設定方法例も記載している。

チェックリストの目的は、「選択した設定基準に対応した要求設定項目の設定忘れの防止」と「サーバ構築の作業受託先が適切に要求設定項目を設定したことの確認」を行うための手段となるものである。

1.2 本書が対象とする読者

対象読者は、主に SSL/TLS サーバを実際に構築するにあたって具体的な設定を行うサーバ構築者、実際のサーバ管理やサービス提供に責任を持つことになるサーバ管理者、並びに SSL/TLS サ

ーバの構築を発注するシステム担当者としている。
一部の内容については、ブラウザを使う一般利用者への注意喚起も対象とする。

1.3 ガイドラインの検討体制

本ガイドラインは、CRYPTREC 暗号技術活用委員会の配下に設置された運用ガイドラインワーキンググループに参加する委員の知見を集約したベストプラクティスとして作成されたものであり、暗号技術活用委員会及び暗号技術検討会の承認を得て発行されたものである。
運用ガイドラインワーキンググループは表 1 のメンバーにより構成されている。

表 1　運用ガイドラインワーキンググループの構成

主査	菊池 浩明	明治大学　総合数理学部 先端メディアサイエンス学科 教授
委員	阿部 貴	株式会社シマンテック　SSL 製品本部 SSL プロダクトマーケティング部　マネージャー
委員	漆嶌 賢二	富士ゼロックス株式会社　CS 品質本部 品質保証部 マネージャー
委員	及川 卓也	グーグル株式会社　エンジニアリング シニアエンジニアリングマネージャー
委員	加藤 誠	一般社団法人 Mozilla Japan　技術部 テクニカルアドバイザ
委員	佐藤 直之	株式会社イノベーションプラス Director
委員	島岡 政基	セコム株式会社 IS 研究所 コミュニケーションプラットフォームディビジョン 暗号・認証基盤グループ　主任研究員
委員	須賀 祐治	株式会社インターネットイニシアティブ サービスオペレーション本部　セキュリティ情報統括室 シニアエンジニア
委員	高木 浩光	独立行政法人産業技術総合研究所 セキュアシステム研究部門 主任研究員
委員	村木 由梨香	日本マイクロソフト株式会社 セキュリティレスポンスチーム セキュリティプログラムマネージャ
委員	山口 利恵	東京大学　大学院　情報理工学系研究科 ソーシャル ICT 研究センター　　特任准教授
執筆 とりまとめ	神田　雅透	情報処理推進機構　技術本部　セキュリティセンター

2. 本ガイドラインの理解を助ける技術的な基礎知識

2.1 SSL/TLS の概要

2.1.1 SSL/TLS の歴史

Secure Sockets Layer (SSL)はブラウザベンダであった Netscape 社により開発されたクライアントーサーバモデルにおけるセキュアプロトコルである。SSL には 3 つのバージョンが存在するがバージョン 1.0 は非公開である。SSL2.0 が 1995 年にリリースされたが、その後すぐに脆弱性が発見され、翌 1996 年に SSL3.0 [RFC6101] が公開されている。

標準化団体 Internet Engineering Task Force (IETF)は非互換性の問題を解決するために、Transport Layer Security Protocol Version 1.0 (TLS1.0) [RFC2246] を策定した。TLS1.0 は SSL3.0 をベースにしている。TLS1.0 で定められているプロトコルバージョンからも分かるように TLS1.0 は SSL3.1 とも呼ばれる。

TLS1.1 [RFC4346] は、TLS1.0 における暗号利用モードの一つである CBC[1]モードで利用される初期ベクトルの選択とパディングエラー処理に関連する脆弱性に対処するために仕様策定が行われた。具体的には BEAST[2]攻撃を回避することができる。

さらに TLS1.2 [RFC5246] は特にハッシュ関数 SHA-2 family (SHA-256 や SHA-384)の利用など、より強い暗号アルゴリズムの利用が可能になった。例えばメッセージ認証コード（MAC[3]）や擬似乱数関数にて SHA-2 family が利用可能になっている。また認証付暗号利用モードが利用可能な暗号スイートのサポートがなされており、具体的には GCM[4]や CCM[5]モードの利用が可能になった。

表 2 に SSL/TLS のバージョンの概要をまとめる。最近では、IETF において、TLS1.3 の規格化の議論が進んでいる。

なお、SSL/TLS に対する攻撃方法の技術的な詳細については、「CRYPTREC 暗号技術ガイドライン（SSL/TLS における近年の攻撃への対応状況）[6]」を参照されたい。

[1] CBC: Ciphertext Block Chaining
[2] BEAST: Browser Exploit Against SSL/TLS
[3] MAC: Message Authentication Code
[4] GCM: Galois/Counter Mode
[5] CCM: Counter with CBC-MAC
[6] http://www.cryptrec.go.jp/report/c13_kentou_giji02_r2.pdf

表 2　SSL/TLS のバージョン概要

バージョン	概要
SSL2.0 （1994）	● いくつかの脆弱性が発見されており、なかでも「ダウングレード攻撃（最弱のアルゴリズムを強制的に使わせることができる）」と「バージョンロールバック攻撃（SSL2.0 を強制的に使わせることができる）」は致命的な脆弱性といえる ● SSL2.0 は利用すべきではなく、実際に 2005 年頃以降に出荷されているサーバやブラウザでは SSL2.0 は初期状態で利用不可となっている
SSL3.0 （RFC6101） （1995）	● SSL2.0 での脆弱性に対処したバージョン ● 2014 年 10 月に POODLE[7]攻撃が発表されたことにより特定の環境下での CBC モードの利用は危険であることが認識されている。POODLE 攻撃は、SSL3.0 におけるパディングチェックの仕方の脆弱性に起因しているため、この攻撃に対する回避策は現在のところ存在していない ● POODLE 攻撃の発表を受け、必要に応じてサーバやブラウザの設定を変更し、SSL3.0 を無効化にするよう注意喚起されている [8]
TLS1.0 （RFC2246） （1999）	● 2015 年 3 月時点では、もっとも広く実装されているバージョンであり、実装率はほぼ 100% ● ブロック暗号を CBC モードで利用した時の脆弱性を利用した攻撃（BEAST 攻撃など）が広く知られているが、容易な攻撃回避策が存在し、すでにセキュリティパッチも提供されている。また、SSL3.0 で問題となった POODLE 攻撃は、プロトコルの仕様上、TLS1.0 には適用できない ● 暗号スイートとして、より安全なブロック暗号の AES と Camellia、並びに公開鍵暗号・署名に楕円曲線暗号が利用できるようになった ● 秘密鍵の生成などに擬似乱数関数を採用 ● MAC の計算方法を HMAC に変更
TLS1.1 （RFC4346） （2006）	● ブロック暗号を CBC モードで利用した時の脆弱性を利用した攻撃（BEAST 攻撃等）への対策を予め仕様に組み入れるなど、TLS1.0 の安全性強化を図っている ● 実装に関しては、規格化された年が TLS1.2 とあまり変わらなかったため、TLS1.1 と TLS1.2 は同時に実装されるケースも多く、2015 年 3 月時点でのサーバやブラウザ全体における実装率は約 50%
TLS1.2 （RFC5246） （2008）	● 暗号スイートとして、より安全なハッシュ関数 SHA-256 と SHA-384、CBC モードより安全な認証付暗号利用モード（GCM、CCM）が利用できるようになった。特に、認証付暗号利用モードでは、利用するブロック暗号が同じであっても、CBC モードの脆弱性を利用した攻撃（BEAST 攻撃等）がそもそも適用できない ● 必須の暗号スイートを TLS_RSA_WITH_AES_128_CBC_SHA に変更 ● IDEA と DES を使う暗号スイートを削除 ● 擬似乱数関数の構成を MD5/SHA-1 ベースから SHA-256 ベースに変更 ● 本格的に実装が始まったのは最近であり、2015 年 3 月時点でのサーバやブラウザ全体における実装率は約 55%

[7] POODLE: Padding Oracle On Downgraded Legacy Encryption
[8] SSL3.0 の脆弱性対策について、http://www.ipa.go.jp/security/announce/20141017-ssl.html

2.1.2 プロトコル概要

SSL/TLS はセッション層に位置するセキュアプロトコルで、通信の暗号化、データ完全性の確保、サーバ（場合によりクライアント）の認証を行うことができる。セッション層に位置することで､アプリケーション層ごとにセキュリティ確保のための仕組みを実装する必要がなく､HTTP、SMTP、POP など様々なプロトコルの下位に位置して、上記のような機能を提供することができる。

SSL/TLS では、暗号通信を始めるに先立って、ハンドシェイクが実行される（図 1 参照)。

ハンドシェイクでは、①ブラウザ（クライアント）とサーバが暗号通信するために利用する暗号アルゴリズムとプロトコルバージョンを決定し､②サーバ証明書によってサーバの認証を行い、③そのセッションで利用するセッション鍵を共有する、までの一連の動作を行う。

その際、SSL/TLS では相互接続性の確保を優先してきたため、一般には複数の暗号アルゴリズムとプロトコルバージョンが実装されている。結果として、暗号通信における安全性強度は、ハンドシェイクの①の処理でどの暗号アルゴリズムとプロトコルバージョンを選択したかに大きく依存する。

サイトの身分証明ともいえるサーバ証明書は、Trusted Third Party である認証局（CA[9]）によって発行されるのが一般的であり、特に Web Trust for CA などの一定の基準を満たしている代表的な認証局の証明書はルート CA として予めブラウザに登録されている。(4)の検証では、ブラウザに登録された認証局の証明書を信頼の起点として、当該サーバ証明書の正当性を確認する。

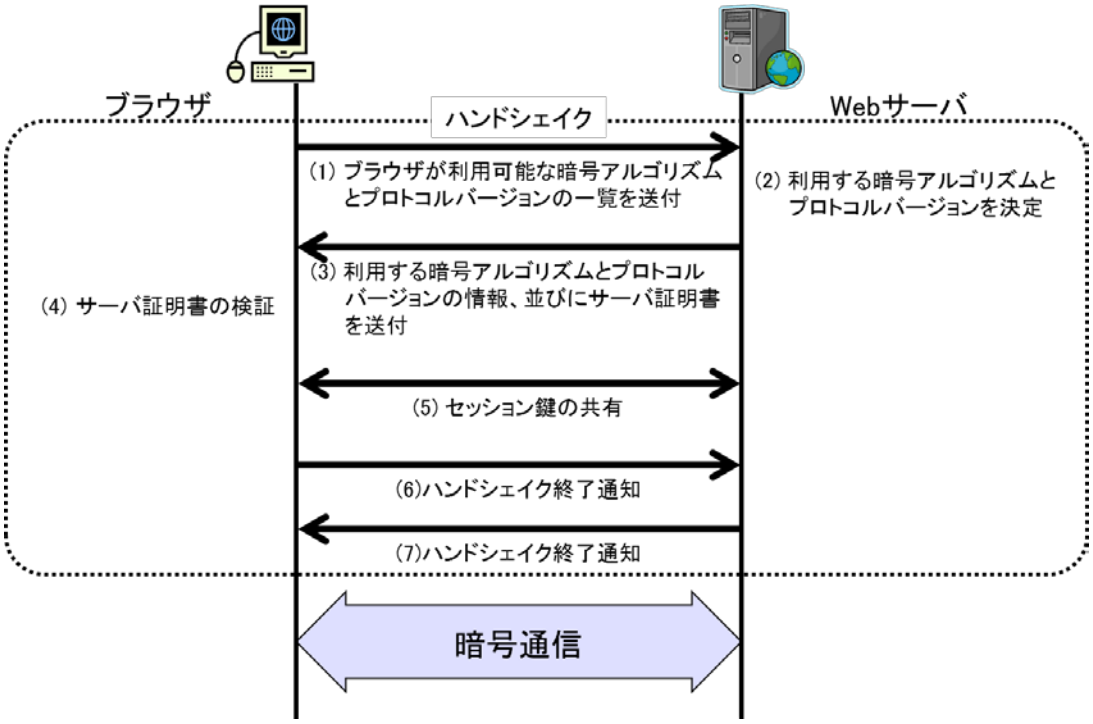


図 1　SSL/TLS プロトコル概要

[9] Certificate Authority

2.2 暗号アルゴリズムの安全性

2.2.1 CRYPTREC 暗号リスト

総務省と経済産業省は、暗号技術に関する有識者で構成される CRYPTREC 活動を通して、電子政府で利用される暗号技術の評価を行っており、2013 年 3 月に「電子政府における調達のために参照すべき暗号のリスト(CRYPTREC 暗号リスト)」 を策定した [10]。CRYPTREC 暗号リストは、「電子政府推奨暗号リスト」、「推奨候補暗号リスト」及び「運用監視暗号リスト」で構成される。

「政府機関の情報セキュリティ対策のための統一基準（平成 26 年度版）」（平成 26 年 5 月 19 日、情報セキュリティ政策会議） では以下のように記載されており、 政府機関における情報システムの調達及び利用において、CRYPTREC 暗号リストのうち「電子政府推奨暗号リスト」が原則的に利用される。

政府機関の情報セキュリティ対策のための統一基準（抄）

6.1.5 暗号・電子署名－遵守事項(1)(b)

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

（ア） 行政事務従事者が暗号化及び電子署名に対して使用するアルゴリズムについて、「電子政府推奨暗号リスト」に記載された暗号化及び電子署名のアルゴリズムが使用可能な場合には、それを使用させること。

（イ） 情報システムの新規構築又は更新に伴い、暗号化又は電子署名を導入する場合には、やむを得ない場合を除き、「電子政府推奨暗号リスト」に記載されたアルゴリズムを採用すること。

（以下、略）

2.2.2 異なる暗号アルゴリズムにおける安全性の見方

異なる技術分類の暗号アルゴリズムを組合せて利用する際、ある技術分類の暗号アルゴリズムの安全性が極めて高いものであっても、別の技術分類の暗号アルゴリズムの安全性が低ければ、結果として、低い安全性の暗号アルゴリズムに引きずられる形で全体の安全性が決まる。逆に言えば、異なる技術分類の暗号アルゴリズムであっても、同程度の安全性とみなされている暗号アルゴリズムを組合せれば、全体としても同程度の安全性が実現できることになる。

異なる技術分類の暗号アルゴリズムについて同程度の安全性を持つかどうかを判断する目安として、“ビット安全性（等価安全性ということもある)”という指標がある。具体的には、評価対象とする暗号アルゴリズムに対してもっとも効率的な攻撃手法を用いたときに、どの程度の計算量があれば解読できるか（解読計算量 [11]）で表現され、鍵長 [12]とは別に求められる。表記上、解

[10] http://www.cryptrec.go.jp/images/cryptrec_ciphers_list_2013.pdf

[11] 直感的には、基本となる暗号化処理の繰り返し回数のことである。例えば、解読計算量 2^{20} といえば、暗号化処理 2^{20} 回相当の演算を繰り返し行えば解読できることを意味する

読計算量が 2^x である場合に“x ビット安全性”という。例えば、共通鍵暗号においては、全数探索する際の鍵空間の大きさで 2^x（x は共通鍵のビット長）、ハッシュ関数の例としては、一方向性で 2^x、衝突困難性で $2^{(x/2)}$（x は出力ビット長）が解読計算量の（最大）理論値である。

“ビット安全性”による評価では、技術分類に関わらず、どの暗号アルゴリズムであっても、解読計算量が大きければ安全性が高く、逆に小さければ安全性が低い。また、解読計算量が実現可能と考えられる計算量を大幅に上回っていれば、少なくとも現在知られているような攻撃手法ではその暗号アルゴリズムを破ることは現実的に不可能であると予測される。

そこで、暗号アルゴリズムの選択においては、“x ビット安全性”の“x ビット”に着目して、長期的な利用期間の目安とする使い方ができる。例えば、NIST SP800-57 Part 1 revision 3[13]では、表 3 のように規定している。

なお、表記の便宜上、本ガイドラインでは以下の表記を用いる。

- AES-xxx：鍵長が xxx ビットの AES のこと
- Camellia-xxx：鍵長が xxx ビットの Camellia のこと
- RSA-xxx：鍵長が xxx ビットの RSA のこと
- DH-xxx：鍵長が xxx ビットの DH のこと
- ECDH-xxx：鍵長が xxx ビット（例えば NIST 曲線パラメータ P-xxx を利用）の ECDH のこと
- ECDSA-xxx：鍵長が xxx ビット（例えば NIST 曲線パラメータ P-xxx を利用）の ECDSA のこと
- HMAC-SHA-xxx：メッセージ認証子を作る HMAC において利用するハッシュ関数 SHA-xxx のこと。SSL/TLS では、暗号スイートで決めるハッシュ関数は HMAC として利用される。
- SHA-xxx：デジタル署名を作成する際に利用するハッシュ関数 SHA-xxx のこと。

[12] ハッシュ関数の場合はダイジェスト長に相当する

[13] NIST SP800-57, Recommendation for Key Management – Part 1: General (Revision 3)

表 3　NIST SP800-57 でのビット安全性の取り扱い方針（WG で加工）

<table>
<tr><td rowspan="2">ビット安全性</td><td rowspan="2">SSL で利用する
代表的な暗号
アルゴリズム</td><td rowspan="2">利用上の条件</td><td colspan="2">長期的な利用期間</td></tr>
<tr><td>2030 年まで</td><td>2031 年以降</td></tr>
<tr><td rowspan="2">80 ビット</td><td rowspan="2">RSA-1024
DH-1024
ECDH-160
ECDSA-160
SHA-1</td><td>新規に処理をする場合</td><td>利用不可</td><td>利用不可</td></tr>
<tr><td>過去に処理したものを利用する場合</td><td colspan="2">過去のシステムとの互換性維持の利用だけを容認</td></tr>
<tr><td rowspan="2">112 ビット</td><td rowspan="2">3-key Triple DES
RSA-2048
DH-2048
ECDH-224
ECDSA-224</td><td>新規に処理をする場合</td><td>利用可</td><td>利用不可</td></tr>
<tr><td>過去に処理したものを利用する場合</td><td>利用可</td><td>過去のシステムとの互換性維持の利用だけを容認</td></tr>
<tr><td>128 ビット</td><td>AES-128
Camellia-128
ECDH-256
ECDSA-256
SHA-256</td><td>特になし</td><td>利用可</td><td>利用可</td></tr>
<tr><td>128 ビットから
192 ビットの間</td><td>RSA-4096
DH-4096
HMAC-SHA-1</td><td>特になし</td><td>利用可</td><td>利用可</td></tr>
<tr><td>192 ビット</td><td>ECDH-384
ECDSA-384
SHA-384</td><td>特になし</td><td>利用可</td><td>利用可</td></tr>
<tr><td>256 ビット</td><td>AES-256
Camellia-256
ECDH-521
ECDSA-521
HMAC-SHA256</td><td>特になし</td><td>利用可</td><td>利用可</td></tr>
<tr><td>256 ビット以上</td><td>HMAC-SHA384</td><td>特になし</td><td>利用可</td><td>利用可</td></tr>
</table>

PART I：

サーバ構築における設定要求項目について

3. 設定基準の概要

本章では、SSL/TLS サーバの構築時に、主に暗号通信に関わる設定に関しての要求事項を決めるために考慮すべきポイントについて取りまとめる。

3.1 実現すべき設定基準の考え方

SSL/TLS は、1994 年に SSL2.0 が実装されて以来、その利便性から多くの製品に実装され、利用されている。一方、プロトコルの脆弱性に対応するため、何度かプロトコルとしてのバージョンアップが行われている影響で、製品の違いによってサポートしているプロトコルバージョンや暗号スイート等が異なり、相互接続性上の問題が生じる可能性がある。

本ガイドラインでは、安全性の確保と相互接続の必要性のトレードオフにより、「高セキュリティ型」「推奨セキュリティ型」「セキュリティ例外型」の 3 段階の設定基準に分けて各々の要求設定を定める。それぞれの設定基準における、安全性と相互接続性についての関係は表 4 のとおりである。

実際にどの設定基準を採用するかは、安全性の確保と相互接続の必要性の両面を鑑みて、サーバ管理やサービス提供に責任を持つ管理者が最終的に決定すべきことではあるが、本ガイドラインでは、安全性もしくは相互接続性についての特段の要求がなければ「推奨セキュリティ型」の採用を強く勧める。本ガイドラインの発行時点では、「推奨セキュリティ型」がもっとも安全性と可用性（相互接続性）のバランスが取れている要求設定であると考えている。

「セキュリティ例外型」は、システム等の制約上、脆弱なプロトコルバージョンである SSL3.0 の利用を全面禁止することのほうが現時点ではデメリットが大きく、安全性上のリスクを受容してでも SSL3.0 を継続利用せざるを得ないと判断される場合にのみ採用すべきである。なお、セキュリティ例外型であっても、SSL3.0 の無期限の継続利用を認めているわけではなく、近いうちに SSL3.0 を利用不可に設定するように変更される可能性がある。

また、SSL3.0 を利用する関係から、利用可能な暗号スイートの設定においても、脆弱な暗号アルゴリズムである RC4 の利用を認めている。ただし、本来的には RC4 は SSL3.0 に限定して利用すべきであるが、TLS1.0 以上のプロトコルバージョンで RC4 の利用を不可にする設定を行うことが難しいため、TLS1.0 以上であっても RC4 が使われる可能性が排除できないことにも注意されたい。

したがって、セキュリティ例外型を採用する際は、推奨セキュリティ型への早期移行を前提として、移行計画や利用終了期限を定めたりするなど、今後への具体的な対処方針の策定をすべきである。また、金融サービスや電子商取引サービスなど、不特定多数に公開されるサービス等において使用される SSL/TLS サーバであって、やむなくセキュリティ例外型を採用している場合では、利用者に対して「SSL3.0 の利用を許可しており、脆弱な暗号方式が使われる場合がある」等の注意喚起を行うことが望ましい。

表 4　安全性と相互接続性についての比較

設定基準	概要	安全性	相互接続性の確保
高セキュリティ型	扱う情報が漏えいした際、組織の運営や資産、個人の資産やプライバシー等に致命的または壊滅的な悪影響を及ぼすと予想される情報を、極めて高い安全性を確保して SSL/TLS で通信するような場合に採用する設定基準 **※とりわけ高い安全性を必要とする明確な理由があるケースを対象としており、非常に高度で限定的な使い方をする場合の設定基準である。一般的な利用形態で使うことは想定していない** ＜利用例＞ 政府内利用（G2G 型）のなかでも、限定された接続先に対して、とりわけ高い安全性が要求される通信を行う場合	本ガイドラインの公開時点（2015 年 5 月）において、標準的な水準を大きく上回る高い安全性水準を達成	最近提供され始めたバージョンの OS やブラウザが搭載されている PC、スマートフォンでなければ接続できない可能性が高い また、PC、スマートフォン以外では、最新の機器であっても一部の機器について接続できない可能性がある
推奨セキュリティ型	扱う情報が漏えいした際、組織の運営や資産、個人の資産やプライバシー等に何らかの悪影響を及ぼすと予想される情報を、安全性確保と利便性実現をバランスさせて SSL/TLS での通信を行うための標準的な設定基準 **※ほぼすべての一般的な利用形態で使うことを想定している** ＜利用例＞ • 政府内利用（G2G 型）や社内システムへのリモートアクセスなど、特定された通信相手との安全な通信が要求される場合 • 電子申請など、企業・国民と役所等との電子行政サービスを提供する場合 • 金融サービスや電子商取引サービス、多様な個人情報の入力を必須とするサービス等を提供する場合 • 既存システムとの相互接続を考慮することなく、新規に社内システムを構築する場合	本ガイドラインの公開時点（2015 年 5 月）における標準的な安全性水準を実現	本ガイドラインで対象とするブラウザ（8.1.2 節）が搭載されている PC、スマートフォンなどでは問題なく相互接続性を確保できる 本ガイドラインが対象としない、バージョンが古い OS やブラウザの場合や発売開始からある程度の年月が経過している一部の古い機器（フィーチャーフォンやゲーム機など）については接続できない可能性がある

安全性と相互接続性についての比較（続）

設定基準	概要	安全性	相互接続性の確保
セキュリティ例外型	脆弱なプロトコルバージョンや暗号が使われるリスクを受容したうえで、安全性よりも相互接続性に対する要求をやむなく優先させて SSL/TLS での通信を行う場合に許容しうる最低限度の設定基準 **※推奨セキュリティ型への早期移行を前提として、暫定的に利用継続するケースを想定している** ＜利用例＞ • 利用するサーバやクライアントの実装上の制約、もしくは既存システムとの相互接続上の制約により、推奨セキュリティ型（以上）の設定が事実上できない場合	推奨セキュリティ型への移行完了までの短期的な利用を前提に、本ガイドラインの公開時点（2015 年 5 月）において許容可能な最低限の安全性水準を満たす	最新ではないフィーチャーフォンやゲーム機などを含めた、ほとんどのすべての機器について相互接続性を確保できる

3.2 要求設定の概要

SSL/TLS における暗号通信に関わる設定には以下のものがある。

- プロトコルバージョンの設定（第 4 章）
- サーバ証明書の設定（第 5 章）
- 暗号スイートの設定（第 6 章）
- SSL/TLS を安全に使うために考慮すべきこと（第 7 章）

本ガイドラインでは、上記の設定項目のうち、プロトコルバージョン、サーバ証明書、暗号スイートの 3 つの項目について、3.1 節に記載した設定基準に対応した詳細な要求設定を該当章に各々まとめている。表 5 に要求設定の概要を記す。

表 5　要求設定の概要

<table>
<tr><th colspan="2">要件</th><th>高セキュリティ型</th><th>推奨セキュリティ型</th><th>セキュリティ例外型</th></tr>
<tr><td colspan="2">想定対象</td><td>G2G</td><td>一般</td><td>レガシー携帯電話含む</td></tr>
<tr><td colspan="2">暗号スイートの
（暗号化の）セキュリティ
レベル</td><td>①256 bit
②128 bit</td><td>①128 bit
②256 bit</td><td>① 128 bit
② 256 bit
③ RC4, Triple DES</td></tr>
<tr><td rowspan="5">暗号アルゴリズム</td><td rowspan="2">鍵交換</td><td>鍵長 2048 ビット以上の DHE または 鍵長 256 ビット以上の ECDHE</td><td colspan="2">鍵長 1024 ビット以上の DHE または鍵長 256 ビット以上の ECDHE</td></tr>
<tr><td></td><td colspan="2">鍵長 2048 ビット以上の RSA
鍵長 256 ビット以上の ECDH</td></tr>
<tr><td rowspan="2">暗号化</td><td colspan="3">鍵長 128 ビット及び 256 ビットの AES または Camellia</td></tr>
<tr><td></td><td></td><td>RC4
Triple DES</td></tr>
<tr><td>モード</td><td>GCM</td><td colspan="2">GCM, CBC</td></tr>
<tr><td>ハッシュ関数</td><td>SHA-384, SHA-256</td><td colspan="2">SHA-384, SHA-256, SHA-1</td></tr>
<tr><td colspan="2">プロトコルバージョン</td><td>TLS1.2 のみ</td><td>TLS1.2 ～ TLS1.0</td><td>TLS1.2～1.0, SSL3.0</td></tr>
<tr><td colspan="2">証明書鍵長</td><td colspan="3">鍵長 2048 ビット以上の RSA または 鍵長 256 ビット以上の ECDSA</td></tr>
<tr><td colspan="2">証明書でのハッシュ関数</td><td colspan="2">SHA-256</td><td>SHA-256, SHA-1</td></tr>
</table>

3.3 チェックリスト

図 2 に高セキュリティ型のチェックリストのイメージを示す。
チェックリストの目的は、

- 選択した設定基準に対応した要求設定項目をもれなく実施したことを確認し、設定忘れを防止すること
- サーバ構築の作業受託先が適切に要求設定項目を設定したことを、発注者が文書として確認する手段を与えること

である。したがって、選択した設定基準に応じたチェックリストを用い、すべてのチェック項目について、該当章に記載の要求設定に合致していることを確認して「済」にチェックが入ることが求められる。

Appendix A には、各々の設定基準に対応したチェックリストを載せる。

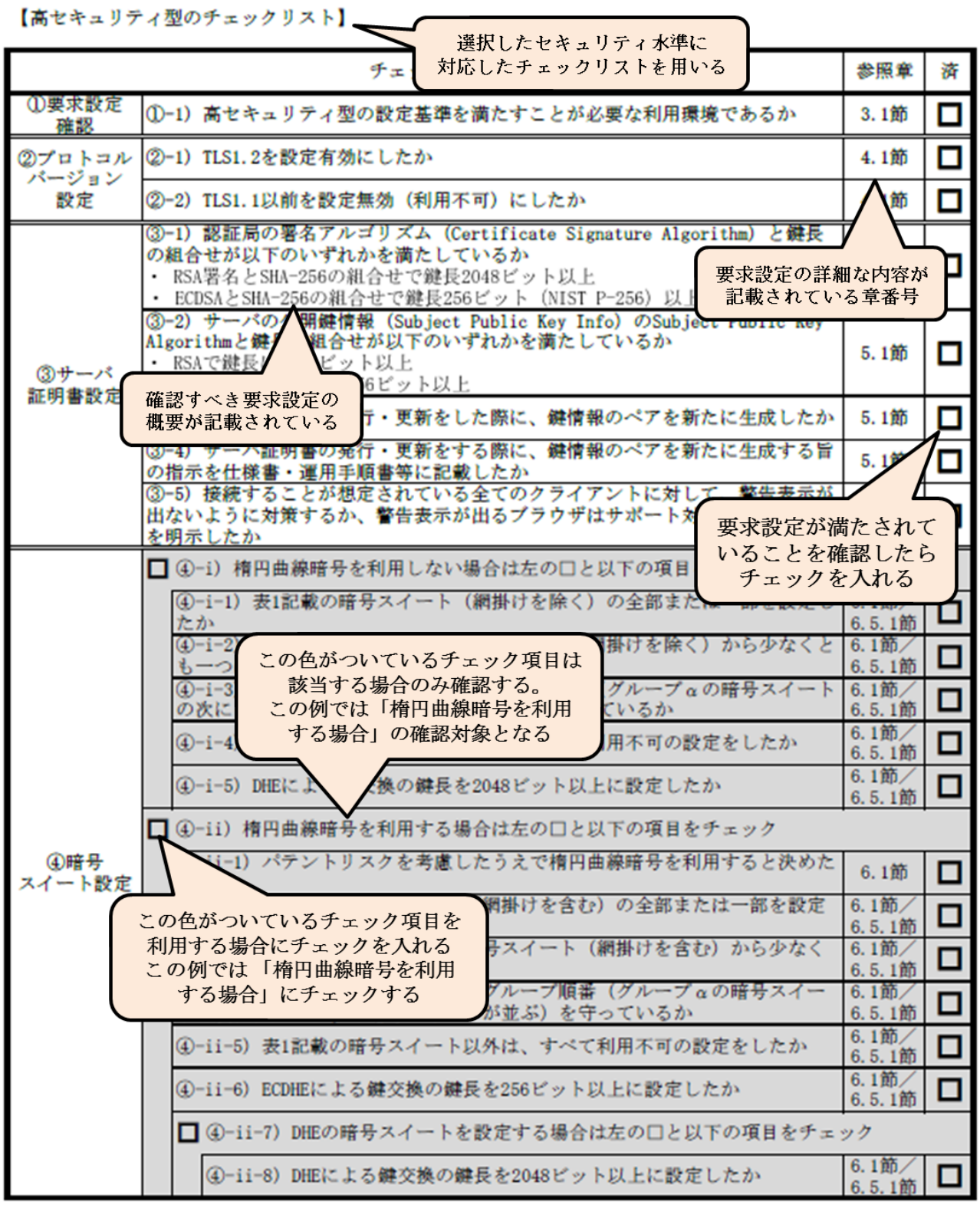


図 2　チェックリスト（高セキュリティ型の例）

4. プロトコルバージョンの設定

SSL/TLS は、1994 年に SSL2.0 が実装され始めた後、2014 年 3 月現在の最新版となる TLS1.2 まで 5 つのプロトコルバージョンが実装されている。4.1 節にプロトコルバージョンについての要求設定をまとめる。4.2 節にプロトコルバージョンごとの安全性の違いを記す。

4.1 プロトコルバージョンについての要求設定

基本的に、プロトコルのバージョンが後になるほど、以前の攻撃に対する対策が盛り込まれるため、より安全性が高くなる。しかし、相互接続性も確保する観点から、多くの場合、複数のプロトコルバージョンが利用できるように実装されているので、プロトコルバージョンの選択順位を正しく設定しておかなければ、予想外のプロトコルバージョンで SSL/TLS 通信を始めることになりかねない。

そこで、SSL2.0 から TLS1.2 までの安全性の違い（4.2 節 表 6 参照）を踏まえ、SSL/TLS サーバがサポートするプロトコルバージョンについての要求設定を以下のように定める。なお、高セキュリティ型の要求設定ではサーバとクライアントの両方が TLS1.2 をサポートしていることが必須となることに注意されたい。

【高セキュリティ型の要求設定】

- TLS1.2 を設定有効にする
- TLS1.1 以前を設定無効（利用不可）にする

TLS1.2	TLS1.1	TLS1.0	SSL3.0	SSL2.0
◎	×	×	×	×

◎：設定有効　　×：設定無効化　　－：実装なし

【推奨セキュリティ型の要求設定】

- SSL3.0 及び SSL2.0 を設定無効（利用不可）にする
- TLS1.1、TLS1.2 については、実装されているのであれば、設定有効にする
- プロトコルバージョンの優先順位が設定できる場合には、設定有効になっているプロトコルバージョンのうち、最も新しいバージョンによる接続を最優先とし、接続できない場合に順番に一つずつ前のプロトコルバージョンでの接続するように設定することが望ましい

TLS1.2	TLS1.1	TLS1.0	SSL3.0	SSL2.0
◎	○	○	×	×
－	◎	○	×	×
－	－	◎	×	×

○：設定有効（◎：優先するのが望ましい）　　×：設定無効化　　－：実装なし

【セキュリティ例外型の要求設定】

- SSL2.0 を設定無効（利用不可）にする
- TLS1.1、TLS1.2 については、実装されているのであれば、設定有効にする
- プロトコルバージョンの優先順位が設定できる場合には、設定有効になっているプロトコルバージョンのうち、最も新しいバージョンによる接続を最優先とし、接続できない場合に順番に一つずつ前のプロトコルバージョンでの接続するように設定することが望ましい

	TLS1.2	TLS1.1	TLS1.0	SSL3.0	SSL2.0
3 つのうちのいずれか	◎	○	○	○	×
	ー	◎	○	○	×
	ー	ー	◎	○	×

○：設定有効（◎：優先するのが望ましい）　×：設定無効化　ー：実装なし

4.2 プロトコルバージョンごとの安全性の違い

SSL2.0 から TLS1.2 までの各プロトコルバージョンにおける安全性の違いを表 6 にまとめる。

表 6　プロトコルバージョンでの安全性の違い

SSL/TLS への攻撃方法に対する耐性	TLS1.2	TLS1.1	TLS1.0	SSL3.0	SSL2.0
ダウングレード攻撃(最弱の暗号アルゴリズムを強制的に使わせることができる)	安全	安全	安全	安全	脆弱
バージョンロールバック攻撃（SSL2.0 を強制的に使わせることができる）	安全	安全	安全	安全	脆弱
ブロック暗号の CBC モード利用時の脆弱性を利用した攻撃(BEAST/POODLE 攻撃など)	安全	安全	パッチ適用要	脆弱	脆弱
利用可能な暗号アルゴリズム	**TLS1.2**	**TLS1.1**	**TLS1.0**	**SSL3.0**	**SSL2.0**
128 ビットブロック暗号（AES, Camellia）	可	可	可	不可	不可
認証付暗号利用モード（GCM, CCM）	可	不可	不可	不可	不可
楕円曲線暗号	可	可	可	不可	不可
SHA-2 ハッシュ関数（SHA-256, SHA-384）	可	不可	不可	不可	不可

【コラム①】　SSL3.0 への大打撃となった POODLE 攻撃

POODLE 攻撃 は BEAST 攻撃に似た攻撃方法であり、SSL3.0 にてブロック暗号を CBC モードで利用する場合の脆弱性を利用した攻撃方法である。BEAST 攻撃同様、例えば、中間者攻撃や攻撃対象に大量の通信を発生させるなど、攻撃には複数の条件が必要であり、ただちに悪用可能な脆弱性ではない。

ただ、BEAST 攻撃に対しては脆弱性を回避するためのセキュリティパッチが公開されており、技術的にもプロトコルそのものを変更しなくても平文を 1 対(N-1)の分割を行うことで回避できる可能性が示されている。これに対して、POODLE 攻撃は SSL3.0 のパディングチェックの仕組み自体の脆弱性に起因している。具体的には、SSL3.0 はパディングの最終 1 バイト分だけをチェックして正しければメッセージ全体が正しいと判断する仕様であるため、攻撃者が作った偽のメッセージであっても 1/256 の確率で正しいものとしてサーバが受理してしまうことを利用した攻撃方法である。

つまり、POODLE 攻撃は SSL3.0 の仕様上の脆弱性に起因することから、脆弱性を回避するためのセキュリティパッチが公開されていない。このため、SSL3.0 自体が古いプロトコルバージョンであることもあり、ブラウザベンダは順次 SSL3.0 をデフォルトで利用不可とする対策を取っている（詳細は 8.3.2 節参照）。また、SSL/TLS サーバ構築者に対しては、SSL3.0 を無効化するための手順を IPA が公表している。

その後、POODLE again ということで「TLS1.x でも POODLE 攻撃が可能」との情報[14]が公開された。ただし、これは、SSL3.0 とは違い、TLS1.x の仕様でも POODLE 攻撃が可能ということではない。実際の TLS1.x の仕様では、パディングの全データをチェックしなければならないことになっており、仕様通りに実装されていれば、攻撃者が作った偽のメッセージをサーバが受理する確率は極めて小さい（具体的には 2^（パディング長）分の 1）。

ところが、実際の製品においては、TLS1.x の仕様に反して、パディングチェックを SSL3.0 と同じ最終 1 バイト分しか行っていないものが数多く見つかり[15]、TLS1.x を使っていても POODLE 攻撃と同じ手法が使えてしまった実装上の問題が発覚した。これが、POODLE again の正体であり、すぐに該当する製品についてはセキュリティパッチが提供された。

[14] https://www.imperialviolet.org/2014/12/08/poodleagain.html
[15] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-8730

5. サーバ証明書の設定

サーバ証明書は、①クライアントに対して、情報を送受信するサーバが意図する相手（サーバの運営組織等）によって管理されるサーバであることを確認する手段を提供することと、②SSL/TLS による暗号通信を行うために必要なサーバの公開鍵情報をクライアントに正しく伝えること、の 2 つの役割を持っている。

これらの役割を正しく実行するために、5.1 節にサーバ証明書についての要求設定をまとめる。5.2 節以降にはサーバ証明書の設定を決める際の検討項目の概要を記す。

5.1 サーバ証明書についての要求設定

後述する 5.2 節以降の内容を踏まえ、サーバ証明書についての要求設定を以下のように定める。なお、本ガイドライン公開時点（2015 年 5 月）においては、推奨セキュリティ型の要求設定は高セキュリティ型と同様とする。

高セキュリティ型（推奨セキュリティ型）の要求設定では、少なくともハッシュ関数として SHA-256 が使えることが必須条件となることに注意されたい。例えば、SHA-256 が使えないブラウザ（クライアント）では、サーバ証明書の検証ができず、警告表示が出るか、当該サーバとの接続は不能となる。このことは、DSA や ECDSA を使う場合についても同様である。

一方、セキュリティ例外型の要求設定では、ハッシュ関数として SHA-1 の利用も許容しており、過去のシステムとの相互接続性は高い。ただし、最新のブラウザでは SHA-1 を使うサーバ証明書に対して警告表示を出すようになってきていることに注意すること。

この他、非技術的要因として、ECDSA を採用する際にはパテントリスクの存在 [16]が広く指摘されているので、十分な検討のうえで採用の可否を決めることが望ましい。

DSA については、5.3 節で示すように電子政府推奨暗号リストに選定されており、安全性上の問題はない。しかし、サーバ証明書としては現時点でほとんど利用されておらず、技術的にも RSA や ECDSA と比較して大きなメリットがあるとは言えないことから、本ガイドラインでは積極的には DSA の利用を勧めない [17]。

[16] 楕円曲線暗号の標準化・規格化を推進するコンソーシアム SECG に対して、Certicom 社から特許レター（RAND 条件でのライセンス許諾）が通知されている。詳細は以下を参照のこと
http://www.secg.org/certicom_patent_letter_SECG.pdf

[17] 本ガイドラインでは、DSA は利用しないことを要求設定の前提としているため、6 章の暗号スイートの設定からも DSA を利用する暗号スイートが除外されていることに留意されたい

【高セキュリティ型の要求設定】

- 本ガイドライン公開時点（2015 年 5 月）で、多くの認証局から入手可能なサーバ証明書のうち、安全性が高いものを利用する。

サーバ証明書のアルゴリズムと鍵長	サーバ証明書の発行・更新を要求する際に生成する鍵情報（Subject Public Key Info）でのアルゴリズム（Subject Public Key Algorithm）と鍵長としては、以下のいずれかを必須とする。 ● RSA（OID = 1.2.840.113549.1.1.1）で鍵長は 2048 ビット以上 ● 楕円曲線暗号で鍵長 256 ビット以上（NIST P-256 の場合の OID = 1.2.840.10045.3.1.7） また、認証局が発行するサーバ証明書での署名アルゴリズム（Certificate Signature Algorithm）と鍵長については、以下のいずれかを必須とする。 ● RSA 署名と SHA-256 の組合せ（sha256WithRSAEncryption; OID = 1.2.840.113549.1.1.11）で鍵長 2048 ビット以上 ● ECDSA と SHA-256 の組合せ（ecdsa-with-SHA256; OID = 1.2.840.10045.4.3.2）で鍵長 256 ビット（NIST P-256）以上
サーバ証明書の発行・更新時の鍵情報の生成	● サーバ証明書の発行・更新を要求する際には、既存の鍵情報は再利用せず、必ず新たに公開鍵と秘密鍵の鍵ペアを生成しなければならない ● 上記の指示をサーバ管理者への仕様書、運用手順書、ガイドライン等に明示しなければならない
クライアントでの警告表示の回避	● 当該サーバに接続することが想定されている全てのクライアントに対して、以下のいずれかの手段を用いて警告表示が出ないようにしなければならない ➢ パブリック認証局からサーバ証明書を入手する ➢ 警告表示が出るクライアントの利用を禁ずる措置を取る ➢ 5.4.2 節の例外規定に従って、信頼できるプライベート認証局のルート CA 証明書をインストールする

【推奨セキュリティ型の要求設定（高セキュリティ型の要求設定と同じ)】

- 本ガイドライン公開時点（2015 年 5 月）で、多くの認証局から入手可能なサーバ証明書のうち、安全性が高いものを利用する。

サーバ証明書の暗号アルゴリズムと鍵長	サーバ証明書の発行・更新を要求する際に生成する鍵情報（Subject Public Key Info）でのアルゴリズム（Subject Public Key Algorithm）と鍵長としては、以下のいずれかを必須とする。 ● RSA（OID = 1.2.840.113549.1.1.1）で鍵長は 2048 ビット以上 ● 楕円曲線暗号で鍵長 256 ビット以上（NIST P-256 の場合の OID = 1.2.840.10045.3.1.7） また、認証局が発行するサーバ証明書での署名アルゴリズム（Certificate Signature Algorithm）と鍵長については、以下のいずれかを必須とする。 ● RSA 署名と SHA-256 の組合せ（sha256WithRSAEncryption; OID = 1.2.840.113549.1.1.11）で鍵長 2048 ビット以上 ● ECDSA と SHA-256 の組合せ（ecdsa-with-SHA256; OID = 1.2.840.10045.4.3.2）で鍵長 256 ビット（NIST P-256）以上
サーバ証明書の発行・更新時の鍵情報の生成	● サーバ証明書の発行・更新を要求する際には、既存の鍵情報は再利用せず、必ず新たに公開鍵と秘密鍵の鍵ペアを生成しなければならない ● 上記の指示をサーバ管理者への仕様書、運用手順書、ガイドライン等に明示しなければならない
クライアントでの警告表示の回避	● 当該サーバに接続することが想定されている全てのクライアントに対して、以下のいずれかの手段を用いて警告表示が出ないようにするか、警告表示が出るブラウザはサポート対象外であることを明示しなければならない ➢ パブリック認証局からサーバ証明書を入手する ➢ 警告表示が出るクライアントの利用を禁ずる措置を取る ➢ 5.4.2 節の例外規定に従って、信頼できるプライベート認証局のルート CA 証明書をインストールする

【セキュリティ例外型の要求設定】

- 本ガイドライン公開時点（2015 年 5 月）で、多くの認証局から入手可能なサーバ証明書のうち、許容可能な水準以上の安全性を確保しているサーバ証明書で、最も相互接続性が高いものを利用する。具体的には、ハッシュ関数について、①SHA-256 では相互接続できないブラウザが一定程度存在する可能性が否定できないこと、②MD5 のような証明書偽造につながる可能性がある致命的な脆弱性が発見されていないこと、から SHA-1 の利用を許容する。
- セキュリティ例外型においては、楕円曲線暗号を利用したサーバ証明書の場合、十分な相互接続性が確保できるとは必ずしも言えないため、RSA の利用を勧める。

サーバ証明書の暗号アルゴリズムと鍵長	サーバ証明書の発行・更新を要求する際に生成する鍵情報（Subject Public Key Info）でのアルゴリズム（Subject Public Key Algorithm）と鍵長としては、以下のいずれかを必須とする。 ● RSA（OID = 1.2.840.113549.1.1.1）で鍵長は 2048 ビット以上 また、認証局が発行するサーバ証明書での署名アルゴリズム（Certificate Signature Algorithm）と鍵長については、以下のいずれかを必須とする。なお、SHA-256 との組合せのほうが望ましいが、状況によっては SHA-1 との組合せを選んでもよい。 ● RSA 署名と SHA-256 の組合せ（sha256WithRSAEncryption; OID = 1.2.840.113549.1.1.11）で鍵長 2048 ビット以上 ● RSA 署名と SHA-1 の組合せ（sha1WithRSAEncryption; OID = 1.2.840.113549.1.1.5）で鍵長 2048 ビット以上 ※ 過去のシステム・ブラウザ等との相互接続性の確保を優先するなら SHA-1 を利用したサーバ証明書のほうがよいが、最新のブラウザでは SHA-1 を使うサーバ証明書に対して警告表示を出すようになってきていることに注意すること。詳細については 8.3.1 節を参照のこと
サーバ証明書の発行・更新時の鍵情報の生成	● サーバ証明書の発行・更新を要求する際には、既存の鍵情報は再利用せず、必ず新たに公開鍵と秘密鍵の鍵ペアを生成しなければならない ● 上記の指示をサーバ管理者への仕様書、運用手順書、ガイドライン等に明示しなければならない
クライアントでの警告表示の回避	● 当該サーバに接続することが想定されている全てのクライアントに対して、以下のいずれかの手段を用いて警告表示が出ないようにするか、警告表示が出るブラウザはサポート対象外であることを明示しなければならない ➢ パブリック認証局からサーバ証明書を入手する ➢ 警告表示が出るクライアントの利用を禁ずる措置を取る ➢ 5.4.2 節の例外規定に従って、信頼できるプライベート認証局のルート CA 証明書をインストールする

5.2 サーバ証明書に記載されている情報

サーバ証明書には、表 7 に示す情報が記載されている。これらの情報は、証明書プロパティの「詳細」で見ることができる。

これらのうち、当該サーバ証明書を発行した認証局が「Issuer（発行者）」となり、当該認証局がサーバ証明書に施すアルゴリズムが「Certificate Signature Algorithm（署名アルゴリズム）」、実際の署名値が「Certificate Signature Value」として記載される。

SSL/TLS サーバを運用するものは「Subject（サブジェクトー発行対象）」となり、当該サーバ自身が利用する公開鍵の情報が「Subject Public Key Info（公開鍵情報）」として記載される。公開鍵情報には「Subject Public Key Algorithm（公開鍵を使う時の暗号アルゴリズム）」と「Subject's Public Key（実際の公開鍵の値）」が含まれており、その公開鍵をどのように使うかは「Certificate Key Usage（キー使用法）」に記載される。

例えば、Subject Public Key Algorithm に「RSA」、Certificate Key Usage に「Signing, Key Encipherment」とある場合には、Subject's Public Key に書かれた公開鍵は、対応する秘密鍵で作られた RSA 署名（Signing）の検証用途にも、セッション鍵を送付する RSA 暗号化（Key Encipherment）用途にも使えることを意味する。

表 7　サーバ証明書に記載される情報

証明書のバージョン	Version
シリアル番号	Serial Number
署名アルゴリズム	Certificate Signature Algorithm
発行者	Issuer
有効期間（開始～終了）	Validity (Not Before ～ Not After)
サブジェクト（発行対象）	Subject
（サブジェクトが使う）公開鍵情報 [18]	Subject Public Key Info (Algorithm, Public Key Value)
拡張情報	Extensions
キー使用法	Certificate Key Usage
署名	Certificate Signature Value

5.3 サーバ証明書で利用可能な候補となる暗号アルゴリズム

本ガイドラインにおいて「サーバ証明書で利用可能な候補となる暗号アルゴリズム」とは、サ

[18] Windows の証明書プロパティでは『公開キー』と表記されているが、本文中では『公開鍵』で表記を統一する。

ーバ証明書の仕様に合致するものに採用されている「署名」と「ハッシュ関数」のうち、CRYPTREC 暗号リスト（2.2.1 節参照）にも掲載されているものとする。具体的には、表 8 に示した「署名」と「ハッシュ関数」である。

現在発行されているサーバ証明書は、大多数が RSA と SHA-256 との組合せによるものか、RSA と SHA-1 との組み合わせによるものである。特に最近では、安全性向上が必要との観点から SHA-1 から SHA-256 への移行も急速に進みだしている。また、RSA でも鍵長が 1024 ビットから 2048 ビットへ移行している一方、処理性能の低下を避けるために鍵長 256 ビットの ECDSA を採用するケースも増えてきている。

実際に、従来 RSA と SHA-1 の組合せでしかサーバ証明書を発行しなかった認証局でも、SHA-256 や ECDSA に対応したサーバ証明書を発行するようになってきている。このような動きに対応し、比較的新しいブラウザやクライアント機器では SHA-256 や ECDSA を使ったサーバ証明書でも問題なく検証できるようになっている。

ただし、本ガイドライン公開時点（2015 年 5 月）では、古い機器などを中心に、SHA-256 や ECDSA を使ったサーバ証明書の検証ができないクライアントも相当数存在していると考えられるため、古い機器との相互接続性の確保を考慮するのであれば、一定の配慮が必要となる。

表 8　サーバ証明書で利用可能な候補となる暗号アルゴリズム

技術分類	リストの種類	アルゴリズム名
署名	電子政府推奨暗号リスト	RSASSA PKCS#1 v1.5（RSA）
		DSA
		ECDSA
ハッシュ関数	電子政府推奨暗号リスト	SHA-256
	運用監視暗号リスト	SHA-1

5.4 サーバ証明書で考慮すべきこと

5.4.1 信頼できないサーバ証明書の利用は止める

ブラウザなどをはじめとするサーバ証明書を検証するアプリケーションには、一定の基準に準拠した認証局の証明書（ルート CA 証明書）があらかじめ登録されており、これらの認証局（とその下位認証局）はパブリック認証局と呼ばれている。一般に、パブリック認証局が、第三者の立場から確認したサーバの運営組織等の情報を記載したサーバ証明書を発行し、ブラウザに予め搭載されたルート CA 証明書と組合せて検証を行うことで、サーバ証明書の信頼性を確保する。これにより、当該サーバ証明書の正当性が確認できれば、ブラウザは警告表示することなく、当該サーバへの接続を行う。

一方、証明書の発行プログラムさえあれば誰もがサーバ証明書を作ることができるが、これではサーバ構築者が“自分は正当なサーバ”であると自己主張しているに過ぎない。このようなサーバ証明書は“オレオレ証明書”ともいわれ、ブラウザでは当該サーバ証明書の正当性が確認で

きない“危険なサーバ”として警告が表示される。

本来、SSL/TLS における重要な役割の一つが接続するサーバの認証であり、その認証をサーバ証明書で行う仕組みである以上、“危険なサーバ”との警告表示が出るにもかかわらず、その警告を無視して接続するよう指示しなければならないサーバ構築の仕方をすべきではない。

5.4.2 ルート CA 証明書の安易な手動インストールは避ける

5.4.1 節のようにして発行されたサーバ証明書を利用した SSL/TLS サーバを“危険なサーバ”として認識させない手段として、当該サーバ証明書の正当性を確認するためのルート CA 証明書を、ブラウザ（クライアント）の「信頼できるルート CA」に手動でインストールする方法がある。

しかし、安易に「信頼できるルート CA」として手動インストールをすることは、“危険なサーバ”との警告を意図的に無視することにつながる。また、5.4.1 節に記載したパブリック認証局のルート CA 証明書とは異なり、これら手動インストールしたルート CA 証明書はブラウザベンダによって管理されていない。このため、万が一、当該ルート CA 証明書の安全性に問題が生じた場合でも、ブラウザベンダによって自動的に無効化されることはなく、インストールした当該ルート CA 証明書を利用者自身が手動で削除する必要がある。もし、削除を怠ると不正なサーバ証明書を誤信するリスクが増大する。

したがって、ルート CA 証明書の手動インストールは原則として避けるべきであり、ましてや利用者に対して手動インストールを求めるような運用をすべきではない。

例外的にルート CA 証明書の手動インストールを行う必要がある場合には、ルート CA 証明書の安全性に問題が生じた場合にインストールを勧めた主体によって、利用者のブラウザから当該ルート CA 証明書の無効化や削除ができるようにする等、インストールした利用者に対して具体的に責任を負うことができる体制を整えるべきである。

例えば、企業・団体等が自身の管理する端末に対して、当該組織が自前で構築した、言わばプライベートなルート CA 証明書をシステム管理部門等の管理下でインストールし、また当該ルート CA 証明書の安全性に問題が生じた場合には、速やかに当該部門が各端末に対して当該ルート CA 証明書を無効化する措置を講ずることができるような体制である。具体的には、組織等において一定のポリシーに基づいて端末管理を行っている場合、管理システムなどから各端末にルート CA 証明書を自動更新（インストールおよび削除）する仕組みを提供するなどである。一例として Windows クライアントに対して Active Directory から自動更新する場合の構成例を Appendix D.2 に示す。

このような仕組みを用いて各端末にインストールされたルート CA 証明書の安全性に問題が生じた場合には、当該組織の責任において、当該ルート CA 証明書を速やかに自動削除するなどの無効化する措置を講じなければならない。

5.4.3 サーバ証明書で利用すべき鍵長

署名の安全性は鍵長にも大きく影響される。通常、同じアルゴリズムであれば、鍵長が長いほ

ど安全性を高くすることができる。ただし、あまりにも長くしすぎると処理時間が過大にかかるようになり、実用性を損なうことにもつながる。

CRYPTREC では、素因数分解問題の困難性に関する調査研究に基づいて RSA の安全性に関する見積りを作成している。これによれば、RSA 2048 ビットを素因数分解するのにおおむね 10^{25}～10^{27} FLOPS 程度の計算量が必要との見積もりを出しており、劇的な素因数分解手法の発見がない限り、計算機性能の向上を考慮しても世界最速の計算機が 1 年かけて解読可能となるのは 2035 年以降と予想している。また、楕円曲線上の離散対数問題の困難性に関する調査研究も行われており、ECDSA 192 ビットを解くのにおおむね 10^{24}～10^{25} FLOPS 程度の計算量が必要と見積もっている。詳細については、CRYPTREC Report 2013[19] 図 3.1、図 3.2 を参照されたい。

以上の報告と、内閣官房情報セキュリティセンター（現 : 内閣サイバーセキュリティセンター）が公表している「政府機関の情報システムにおいて使用されている暗号アルゴリズム SHA-1 及び RSA1024 に係る移行指針 [20]」を踏まえれば、本ガイドライン公開時点（2015 年 5 月）でサーバ証明書が利用すべき鍵長は、RSA は 2048 ビット以上、ECDSA は 256 ビット以上が妥当である。

5.4.4 サーバ証明書を発行・更新する際に新しい鍵情報を生成する重要性

サーバ証明書を取得する際に、公開鍵と秘密鍵の鍵ペアの生成・運用・管理が正しく行われないと、暗号化された通信データが第三者に復号されてしまうなどの問題が発生するリスクがある。例えば、とりわけ危険なのは、サーバ機器の出荷時に設定されているデフォルト鍵や、デフォルト設定のまま生成した鍵ペアを利用した場合、意図せず第三者と同じ秘密鍵を共有してしまう恐れがある。

また、何らかの理由により秘密鍵が漏えいした恐れがあり、サーバ証明書を再発行する必要性に迫られた時に、前回使用した CSR（Certificate Signing Request : サーバ証明書を発行するための署名要求）を使い回すと、同じ公開鍵と秘密鍵の鍵ペアのまま新しいサーバ証明書が作られるので、古いサーバ証明書を失効させ、新しいサーバ証明書を再発行することの意味がなくなる。

こうしたリスクを防ぐためには、サーバ管理者は、サーバ証明書を取得・更新する際に既存の鍵ペアを使い回すことを厳に慎み、毎回新しく生成した鍵ペアを使った CSR でサーバ証明書を取得・更新しなければならない。

[19] http://www.cryptrec.go.jp/report/c13_eval_web_final.pdf
[20] http://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf

【コラム②】実際にあった！漏えいしたかもしれない鍵ペアを再利用した証明書の再発行

2014 年 4 月、OpenSSL Heartbleed と呼ばれる脆弱性が大きく報道された。これは、サーバの動静を簡易に確認するための TLS1.2 の拡張機能 Heartbeat を実装した OpenSSL の脆弱性を利用した攻撃である。具体的には、OpenSSL の Heartbeat 実装においてメモリサイズのチェックをしなかったため、当該 OpenSSL で構築した SSL/TLS サーバでは、返信すべきデータに隣接するメモリ領域のデータまでも返信してしまうという脆弱性を利用している。

この脆弱性が大きな問題となったのは、仮に攻撃が行われたとしてもサーバ側に攻撃の痕跡が残っていないため、いつ攻撃されたかを特定すること自体ができなかったことに加え、漏えいしたデータは攻撃時にメモリ上に展開されていたデータの一部であったことから、どのデータが漏えいしたのかを特定することが事実上できなかったことである。

このため、SSL/TLS サーバ運用上の最悪ケースとして「サーバの秘密鍵自体が漏えいした可能性が否定できない」とし、対策として OpenSSL のバージョンアップを行うとともに、

対策1.　運用中のサーバ証明書を失効させ、

対策2.　新しい秘密鍵を用いて、

対策3.　新しいサーバ証明書を再取得・再設定する、

ようにとのアナウンスが出された。ところが、実際には、このアナウンスの趣旨がサーバ運用者には正しく伝わらなかった可能性が大きい。

例えば、英 Netcraft 社の調査結果 [21]によれば、Heartbleed の公表後 1 か月間で 43%のサーバ証明書が再設定（対策 3）されたが、3 つの対策全てを実施した（図 3 中の A の部分）のは 14%にすぎなかった。一方、運用中のサーバ証明書を失効（対策 1）させたにも関わらず、漏えいした可能性がある同じ秘密鍵のまま（対策 2 を実施しなかった）でサーバ証明書を再設定（対策 3）したケース（図 3 中の B の部分）が全体の 5%もあったという。

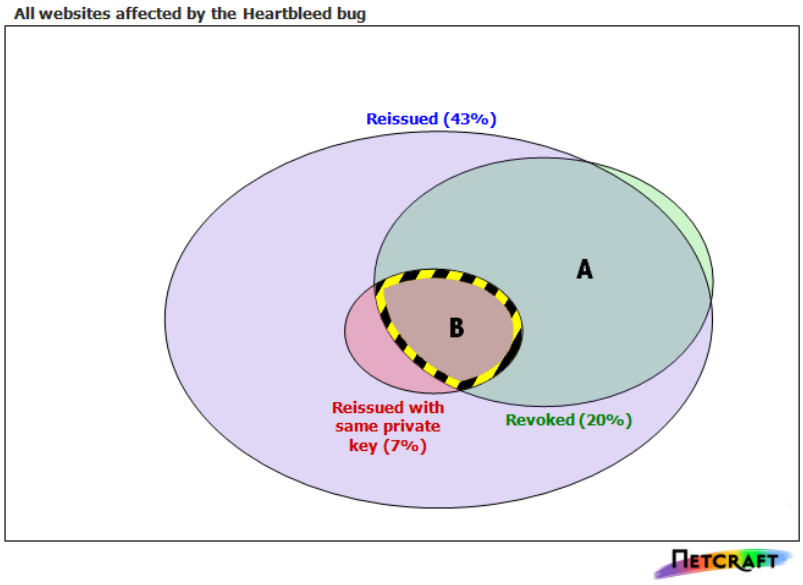


図 3　英 Netcraft 社の調査結果より

21

http://news.netcraft.com/archives/2014/05/09/keys-left-unchanged-in-many-heartbleed-replacement-certificates.html

6. 暗号スイートの設定

暗号スイートは「鍵交換__署名__暗号化__ハッシュ関数」の組によって構成される。

例えば、「TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384」であれば、鍵交換には「DHE」、署名には「RSA」、暗号化には「鍵長 256 ビット GCM モードの Camellia（CAMELLIA_256_GCM）」、ハッシュ関数には「SHA-384」が使われることを意味する。「TLS_RSA_WITH_AES_128_CBC_SHA」であれば、鍵交換と署名には「RSA」、暗号化には「鍵長 128 ビット CBC モードの AES（AES_128_CBC）」、ハッシュ関数には「SHA-1」が使われることを意味する。

実際の SSL/TLS 通信においては、サーバとクライアント間での暗号化通信前の事前通信（ハンドシェイク）時に、両者の合意により一つの暗号スイートを選択する。暗号スイートが選択された後は、選択された暗号スイートに記載の鍵交換、署名、暗号化、ハッシュ関数の方式により SSL/TLS における各種処理が行われる。つまり、SSL/TLS における安全性にとって、暗号スイートをどのように設定するかが最も重要なファクタであることを意味する。

6.1 節に暗号スイートについての要求設定をまとめる。6.2 節から 6.4 節では暗号スイートの設定を決めるうえでの重要な検討項目の概要を記す。

6.1 暗号スイートについての要求設定

一般に、暗号スイートの優先順位の上位から順にサーバとクライアントの両者が合意できる暗号スイートを見つけていく。このため、暗号スイートの選択のみならず、優先順位の設定が重要となる。

その際、多くのブラウザ（クライアント）との相互接続性を確保するためには、多くの製品に共通して実装されている暗号スイートを設定することが不可欠である点に注意する必要がある。一方、高い安全性を実現するためには、比較的新しい製品でしか実装されていないが、高い安全性を持つ暗号アルゴリズムで構成される暗号スイートを設定する必要がある。

上記の点と 6.2 節～6.4 節での内容を踏まえ、本ガイドラインでは、暗号スイートについての要求設定を以下のように定める。なお、本節では、要求設定の概要についてのみ記載する。詳細な要求設定については、各々の該当節を参照すること。

【高セキュリティ型の要求設定】

高セキュリティ型の要求設定の概要は以下の通り。詳細な要求設定は 6.5.1 節を参照のこと。

- 以下の条件を満たす暗号スイートを選定する。
  - CRYPTREC 暗号リストに掲載されているアルゴリズムのみで構成される。
  - 暗号化として 128 ビット安全性以上を有する。
  - 安全性向上への寄与が高いと期待されることから、認証付暗号利用モードを採用する。
  - Perfect Forward Secrecy（後述）の特性を満たす。

- ただし、本ガイドラインではサーバ証明書で DSA を利用しないことを要求設定の前提としている（5.1 節参照）ため、DSA を含む暗号スイートは選定しない。

- 暗号スイートの優先順位は以下の通りとする。
  - 選定した暗号スイートをグループ α とグループ β に分類し、安全性の高いグループを優先する。グループ分けの基準はブロック暗号の鍵長によるものとする。
- 上記以外の暗号スイートは利用除外とする。
- 鍵交換で DHE を利用する場合には鍵長 2048 ビット以上、ECDHE を利用する場合には鍵長 256 ビット以上の設定を必須とする。

【推奨セキュリティ型の要求設定】

推奨セキュリティ型の要求設定の概要は以下の通り。詳細な要求設定は 6.5.2 節を参照のこと。

- 以下の条件を満たす暗号スイートを選定する。
  - CRYPTREC 暗号リストに掲載されているアルゴリズムのみで構成される。
  - 暗号化として 128 ビット安全性以上を有する。
  - ただし、本ガイドラインではサーバ証明書で DSA を利用しないことを要求設定の前提としている（5.1 節参照）ため、DSA を含む暗号スイートは選定しない。
- 暗号スイートの優先順位は以下の通りとする。
  - 選定した暗号スイートを、安全性と実用性とのバランスの観点に立って、グループ A、グループ B、・・・、グループ F とグループ分けをする。
  - 以下の条件でグループごとの優先順位を付ける。
    - 本ガイドライン公開時点（2015 年 5 月）では、通常の利用形態において、128 ビット安全性があれば十分な安全性を確保できることから 128 ビット安全性を優先する。
    - 鍵交換に関しては、Perfect Forward Secrecy の特性の有無と実装状況に鑑み、DHE、次いで RSA の順番での優先順位とする。
- 上記以外の暗号スイートは利用除外とする。
- 鍵交換で DHE を利用する場合には鍵長 1024 ビット以上 [22]、ECDHE/ECDH を利用する場合には鍵長 256 ビット以上、RSA を利用する場合には鍵長 2048 ビット以上の設定を必須とする。

【セキュリティ例外型の要求設定】

セキュリティ例外型の要求設定の概要は以下の通り。詳細な要求設定は 6.5.3 節を参照のこと。

- 以下の条件を満たす暗号スイートを選定する。

[22] ①暗号解読以外の様々な秘密鍵の漏えいリスクを考えれば PFS の特性を優先させるほうが望ましい、②6.3.3 節に示すように DHE を利用する場合、多くの場合で 1024 ビットが選択される環境である、②DHE であれば秘密鍵漏えいの影響が当該セッション通信のみに限定される、ことを踏まえ、本ガイドラインの発行時点での DHE の推奨鍵長は 1024 ビット以上とする

  - CRYPTREC 暗号リストに掲載されているアルゴリズムのみで構成される。
  - ただし、今までほとんど使われていない楕円曲線暗号と Triple DES や RC4 の組合せを今後使っていく積極的な理由は見いだせないことから、楕円曲線暗号と Triple DES、RC4 の組み合わせは選定しない。
  - また、本ガイドラインではサーバ証明書で DSA を利用しないことを要求設定の前提としている（5.1 節参照）ため、DSA を含む暗号スイートも選定しない。
- 暗号スイートの優先順位は以下の通りとする。
  - 選定した暗号スイートを、安全性と実用性とのバランスの観点に立って、グループ A、グループ B、・・・とグループ分けをする。なお、グループ A からグループ F までは推奨セキュリティ型と同様であり、推奨セキュリティ型での優先順位のつけ方を適用する。
- 上記以外の暗号スイートは利用除外とする。
- 鍵交換で DHE を利用する場合には鍵長 1024 ビット以上、ECDHE/ECDH を利用する場合には鍵長 256 ビット以上、RSA を利用する場合には鍵長 2048 ビット以上の設定を必須とする。

6.2 暗号スイートで利用可能な候補となる暗号アルゴリズム

本ガイドラインにおいて「暗号スイートで利用可能な候補となる暗号アルゴリズム」とは、SSL/TLS 用の暗号スイートとして IETF で規格化されたものに採用されている暗号アルゴリズムのうち、CRYPTREC 暗号リスト（2.2.1 節参照）にも掲載されているものとする。具体的には、表 9 に示した暗号アルゴリズムである。

表 9　暗号スイートで利用可能な候補となる暗号アルゴリズム

暗号スイートでの標記	CRYPTREC 暗号リストでの標記		
	技術分類	リストの種類	アルゴリズム名
鍵交換	鍵共有・守秘	電子政府推奨暗号リスト	DH（Ephemeral DH を含む）
			ECDH（Ephemeral DH を含む）
		運用監視暗号リスト	RSAES PKCS#1 v1.5（RSA）
署名	署名	電子政府推奨暗号リスト	RSASSA PKCS#1 v1.5（RSA）
			DSA
			ECDSA
暗号化	128 ビットブロック暗号	電子政府推奨暗号リスト	AES（鍵長 128 ビット、256 ビット）
			Camellia（鍵長 128 ビット、256 ビット）
	暗号利用モード	電子政府推奨暗号リスト	CBC
			GCM
ハッシュ関数	ハッシュ関数	電子政府推奨暗号リスト	SHA-256
			SHA-384
		運用監視暗号リスト	SHA-1

暗号スイートで利用可能な候補となる暗号アルゴリズム（続）

以下は SSL3.0 でのみ利用可			
暗号化	64 ビット ブロック暗号	電子政府推奨暗号 リスト	3-key Triple DES
	ストリーム暗号	運用監視暗号リスト	128-bit RC4

なお、Triple DES は電子政府推奨暗号リストに、RC4 は運用監視暗号リストに掲載されているが、以下の理由を総合的に検討した結果、本ガイドラインでは TLS1.0 以上の場合には Triple DES と RC4 を採用しないことに決定した。

【TLS1.0 以上の場合での Triple DES の除外理由】

- TLS1.0 以上の場合には、Triple DES よりも安全でかつ高速な共通鍵暗号として AES や Camellia が利用可能である。

【TLS1.0 以上の場合での RC4 の除外理由】

- TLS1.0 以上の場合には、RC4 よりもはるかに安全な共通鍵暗号として AES や Camellia が利用可能である。
- ネットワーク環境等の利用状況も踏まえて総合的に判断すれば、RC4 の安全性の脆弱性を大きく優越するほどの実利用における速度優位性が認められない。このことは、RC4 の処理速度が速いという理由が、他の安全な暗号アルゴリズムを使わない理由にはならないことを意味する。
- NIST[23]や ENISA[24]などが最近発行している SSL/TLS での設定ガイドラインにおいても、RC4 は除外されている。

6.3 鍵交換で考慮すべきこと

SSL/TLS の仕様では、実際のデータを暗号化する際に利用する“セッション鍵”はセッションごとに（あるいは任意の要求時点で）更新される。したがって、何らかの理由により、ある時点でのセッション鍵が漏えいした場合でも、当該セッション以外のデータは依然として保護された状態にある。

一方、セッション鍵は暗号通信を始める前にサーバとクライアントとで共有しておく必要があるため、事前通信（ハンドシェイク）の段階でセッション鍵を共有するための処理が行われる。この処理のために使われるのが、表 9 での「鍵共有・守秘」に掲載されている暗号アルゴリズムである。

[23] NIST SP800-52 revision 1 (draft), Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations

[24] ENISA, “Algorithms, Key Sizes and Parameters Report - 2013 recommendations,”

6.3.1 秘密鍵漏えい時の影響範囲を狭める手法の採用（Perfect Forward Secrecy の重要性）

秘密鍵が漏えいする原因は暗号アルゴリズムの解読によるものばかりではない。むしろ、プログラムなどの実装ミスや秘密鍵の運用・管理ミス、あるいはサイバー攻撃やウイルス感染によるものなど、暗号アルゴリズムの解読以外が原因となって秘密鍵が漏えいする場合のほうが圧倒的に多い。

最近でも、OpenSSL Heartbleed Bug や Dual_EC_DRGB の脆弱性などが原因による秘密鍵の漏えいが懸念されており、“秘密鍵が漏えいする”リスクそのものは決して無視できるものではない。スノーデン事件でも話題になったように、秘密鍵の運用・管理そのものに問題がある場合も想定される。

上述した通り、SSL/TLS では、毎回変わるセッション鍵をサーバとクライアントが共有することでセッションごとに違った秘密鍵を使って暗号通信をしており、仮にある時点でのセッション鍵が漏えいした場合でも当該セッション以外のデータは依然として保護されている。

しかし、多くの場合、セッション鍵の交換には固定の鍵情報を使って行っている。このため、どんな理由であれ、もし仮に鍵交換で使う暗号アルゴリズムの“秘密鍵”が漏えいした場合、当該秘密鍵で復号できるセッション鍵はすべて漏えいしたことと同義となる。つまり、SSL/TLS での通信データをためておき、年月が経って、当時の鍵交換で使った暗号アルゴリズムの“秘密鍵”が入手できたならば、過去にさかのぼって、ためておいた通信データの中身が読み出せることを意味している。

そこで、過去の SSL/TLS での通信データの秘匿を確保する観点から、鍵交換で使った暗号アルゴリズムの“秘密鍵”に毎回異なる乱数を付加することにより、見かけ上、毎回異なる秘密鍵を使ってセッション鍵の共有を行うようにする方法がある。これによって、仮に鍵交換で使う暗号アルゴリズムの“秘密鍵”が何らかの理由で漏えいしたとしても、当該セッション鍵の共有のために利用した乱数がわからなければ、当該セッション鍵そのものは求められず、過去に遡及して通信データの中身が読まれる危険性を回避することができる。

このような性質のことを、Perfect Forward Secrecy、または単に Forward secrecy と呼んでいる。なお、本ガイドラインでは Perfect Forward Secrecy（あるいは PFS）に統一して呼ぶこととする。

現在の SSL/TLS で使う暗号スイートの中で、Perfect Forward Secrecy の特性を持つのは Ephemeral DH と Ephemeral ECDH と呼ばれる方式であり、それぞれ DHE、ECDHE と表記される。

6.3.2 鍵交換で利用すべき鍵長

5.4.3 節で述べたことと同様、鍵交換においても、鍵長を長くすれば処理時間や消費リソースなども増えるため、安全性と処理性能、消費リソースなどのトレードオフを考えて適切な鍵長を選択する必要がある。

例えば、処理性能や消費リソースの制約が厳しい組込み機器などの場合、鍵長 4096 ビットの RSA 暗号を利用して得られるメリットよりもデメリットの方が大きくなる可能性がある。CRYPTREC の見積もりでは、劇的な素因数分解手法の発見がない限り、計算機性能の向上を考慮しても世界最速の計算機が 1 年かけて鍵長 2048 ビットの RSA を解読可能となるのは 2035 年以降

と予想している。また、NIST SP800-57 では鍵長 2048 ビットは 2030 年までは利用可とされている（2.2.2 節 表 3 参照）。したがって、2030 年を超えて利用することを想定していないシステムやサービスであれば、2048 ビット以上の鍵長を使うメリットは乏しいといえる。

内閣官房情報セキュリティセンター（現：内閣サイバーセキュリティセンター）が公表している「政府機関の情報システムにおいて使用されている暗号アルゴリズム SHA-1 及び RSA1024 に係る移行指針」、並びに CRYPTREC が公開している公開鍵暗号についての安全性予測を踏まえれば、本ガイドライン公開時点（2015 年 5 月）での鍵交換で利用すべき鍵長は、RSA は 2048 ビット以上、ECDH/ECDHE は 256 ビット以上が妥当である。なお、RSA に関しては、サーバ証明書の申請段階で鍵長 2048 ビット以上を設定することで実現する。

6.3.3 DHE/ECDHE での鍵長の設定状況についての注意

鍵交換について、暗号スイート上は鍵長の規定がない。このため、同じ暗号スイートを使っていても、利用可能な鍵長は製品依存になっていることに注意する必要がある。特に、鍵交換で RSA を使う場合と、DHE や ECDHE/ECDH を使う場合とでは、鍵長の扱いが全く異なるので、それぞれについて適切な設定を行っておく必要がある。

RSA での鍵交換を行う場合にはサーバ証明書に記載された公開鍵を使うことになっており、本ガイドラインの発行時点では鍵長 2048 ビットの公開鍵がサーバ証明書に通常記載されている。このことは、RSA での鍵交換を行う場合、サーバ証明書を正当に受理する限り、どのサーバもブラウザも当該サーバ証明書によって利用する鍵長が 2048 ビットにコントロールされていることを意味する。例え鍵長 2048 ビットの RSA が使えないブラウザがあったとしても、鍵交換が不成立・通信エラーになるだけであり、2048 ビット以外の鍵長が使われることはない。

つまり、RSA での鍵交換に関しては、サーバ証明書の発行時に利用する鍵長を正しく決め、その鍵長に基づくサーバ証明書を発行してもらえばよく、ほとんどの場合、サーバやブラウザ等に特別な設定をする必要はない。

一方、DHE、ECDH/ECDHE については、利用する鍵長がサーバ証明書で明示的にコントロールされるのではなく、個々のサーバやブラウザでの鍵パラメータの設定によって決められる。このため、どの鍵長が利用されるかは、使用する製品での鍵パラメータの設定状況に大きく依存する。例えば、デフォルトで使用する鍵長が製品やバージョンによって異なることが知られており、2013 年夏頃までは鍵長 1024 ビットの DHE しか使えない製品やバージョンも少なくなかった。有名なところでは、Apache 2.4.6 以前、Java 7（JDK7）以前、Windows Server 2012 などがある。

図 4 の 2015 年 1 月の Alexa の調査結果 [25]によれば、約 47 万の主要なサイトについて、DHE が利用できるのは約 52.3%であり、そのうちの約 87.5%（全体では約 45.8%）が鍵長 1024 ビットを採用している。一方、ECDHE が利用できるのは約 62.7%であり、そのうちの約 98%（全体では約 61.5%）が鍵長 256 ビットを採用している。

このことは、DHE を利用した場合は鍵長 1024 ビットが、ECDHE を利用した場合は鍵長 256 ビットが採用される可能性が極めて高いことを意味している。

[25] https://securitypitfalls.wordpress.com/2015/02/01/january-2015-scan-results/

DHE で鍵長 2048 ビットとして使う場合には、鍵長 2048 ビットをサポートしているバージョンを使ったうえで、デフォルトで使用可となっているか、もしくは使用可のオプション設定を行うことが必要である。

【明示的に鍵長 2048 ビットを指定できる代表例】

- OpenSSL
- Apache 2.4.7 以降
- lighttpd 1.4.29 以降
- nginx
- Java 8 以降

これらについては Appendix B.3 に実際の設定例を記す。

【明示的に鍵長を指定できるが、鍵長 2048 ビットをサポートしていない代表例】

- Apache 2.4.6 以前
- Java 7 以前

例えば、Java 7 以前では DHE で扱える鍵長は 64 ビット刻みで 512 ビットから 1024 ビットまでである。これらの製品を利用する場合には、必ず鍵長を 1024 ビットに指定して利用すること。

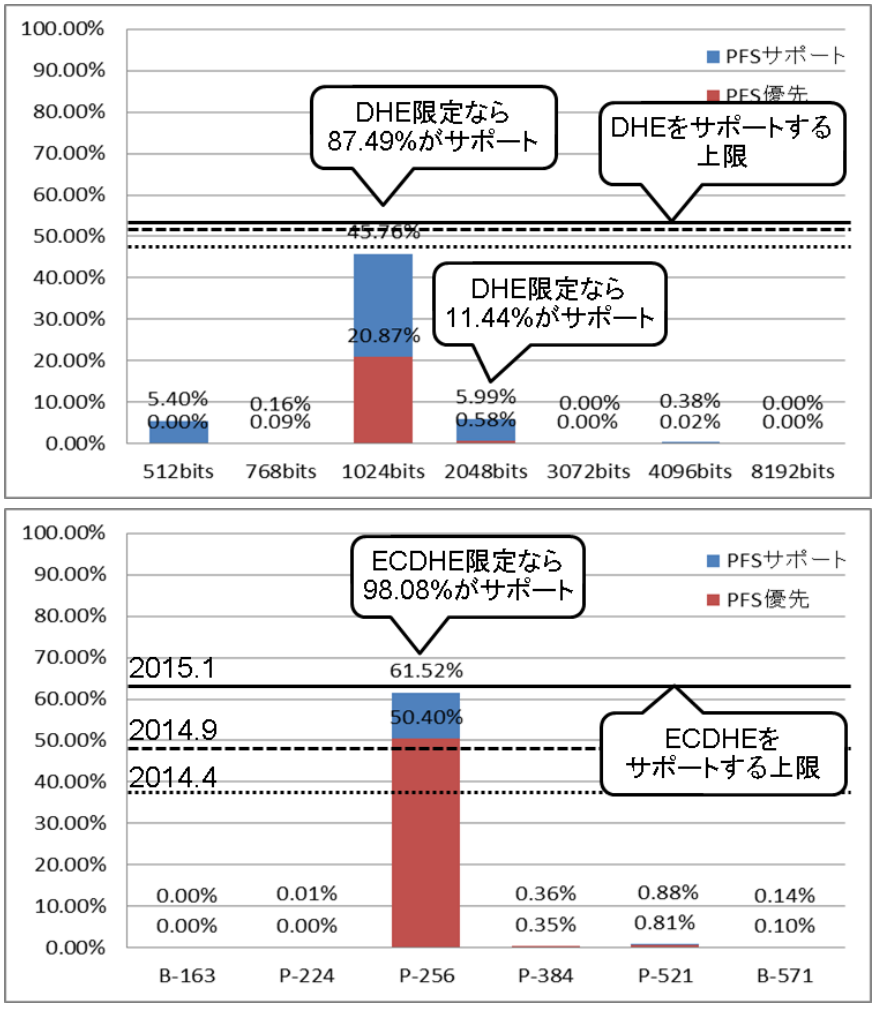


図 4　DHE/ECDHE の鍵長の設定状況（Alexa の調査結果を加工）

【明示的に鍵長を指定できない代表例】

- Apache Tomcat
- Microsoft IIS

これらについては、DHE の鍵長を指定することができず、クライアント側からの指定により 512 ビット、1024 ビット等の弱い鍵パラメータが使われる可能性がある。例えば、サーバ側の設定が鍵長 2048 ビット対応可能だったとしても、本ガイドライン公開時点（2015 年 5 月）では、ブラウザ（クライアント）側が鍵長 2048 ビットに対応していない可能性が十分に考えられる。その場合には、サーバ側は鍵長 1024 ビットを自動的に選択することに注意を要する。

この点は、RSA で鍵交換を行う場合とは大きく事情が異なるため、これらの製品を使う場合には、DHE を含む暗号スイートは選択せず、ECDHE または RSA を含む暗号スイートを使うように設定すべきである。

6.4 暗号スイートについての実装状況

SSL/TLS 用の暗号スイートは IETF で規格化されており、任意に暗号アルゴリズムを選択して「鍵交換＿署名＿暗号化＿ハッシュ関数」の組を自由に作れるわけではない。また、IETF で規格化されている暗号スイートだけでも数多くあるため、実際の製品には実装されていない暗号スイートも多い。

多くの製品に共通して実装されている暗号スイートを設定すれば、相互接続性を広く担保できる可能性が高まる。一方、特定の製品のみに実装されている暗号スイートだけを設定すれば、意図的に当該製品間での接続に限定することができる。

6.5 暗号スイートについての詳細な要求設定

本節では、6.1 節での要求設定の概要に基づき、各々の詳細な要求設定を以下に示す。

なお、鍵交換に PSK または KRB が含まれる暗号スイートは、サーバとクライアントの両方で特別な設定をしなければ利用することができないため、本ガイドラインの対象外とする。

また、非技術的要因として、ECDH や ECDSA を採用する際にはパテントリスクの存在が広く指摘されているので、十分な検討のうえで採用の可否を決めることが望ましい。

6.5.1 高セキュリティ型での暗号スイートの詳細要求設定

6.1 節の条件を踏まえて、表 10 の通り、選定した暗号スイートをグループ α とグループ β に分類する。グループ分けの基準はブロック暗号の鍵長によるものとし、安全性の高いグループをグループ α に割り当て、優先して設定する。

なお、グループ内での暗号スイートから全部または一部を選択して設定するが、その際の優先順位は任意に定めてよい。また、グループ β の暗号スイートについては選択しなくてもよい。

「除外事項」は設定で除外すべき暗号スイートを示したものである[26]。

表 10　高セキュリティ型での暗号スイートの要求設定（基本）

グループ α	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x00,0x9F)
	TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0, 0x7D)
グループ β	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x00,0x9E)
	TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x7C)
設定すべき鍵長	鍵交換で DHE を利用する場合には鍵長 2048 ビット以上の設定を必須とする。なお、DHE の鍵長を明示的に設定できない製品を利用する場合には、DHE を含む暗号スイートは選定すべきではない
高セキュリティ型での除外事項	グループ α、グループ β、表 11 以外のすべての暗号スイートを利用除外とする

パテントリスクについても検討したうえで ECDH や ECDSA を採用することを決めた場合には、表 11 の暗号スイートグループを追加してよい。

表 11　高セキュリティ型での暗号スイートの要求設定（楕円曲線暗号の追加分）

グループ α への追加または代替	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xC0,0x2C)
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xC0,0x30)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x87)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x8B)
グループ β への追加または代替	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xC0,0x2B)
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xC0,0x2F)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x86)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x8A)
設定すべき鍵長	鍵交換で ECDHE を利用する場合には鍵長 256 ビット以上の設定を必須とする

6.5.2 推奨セキュリティ型での暗号スイートの詳細要求設定

6.1 節の条件を踏まえて、表 12 の通り、選定した暗号スイートをグループ A、グループ B、・・・とグループ分けをする。グループ分けの基準は安全性と実用性とのバランスの観点に立って行い、優先設定する順番にグループ A から順に割り当てる。

グループ内での暗号スイートから全部または一部を選択して設定するが、その際の優先順位は任意に定めてよい。また、グループ C 以降の暗号スイートについては選択しなくてもよい。

（RFC 必須）は、TLS1.2 を規定する RFC においてサポートが必須と指定されている暗号スイートであり、不特定多数からのアクセスを想定する SSL/TLS サーバにおいては利用可に設定する

[26] 高セキュリティ型の暗号スイート設定では、TLS1.2 でのサポートが必須と指定されている暗号スイート AES128-SHA を利用した通信が接続不可となることに留意されたい

ことが推奨される暗号スイートである[27]。

また、「除外事項」は設定で除外すべき暗号スイートを示したものである。

表 12　推奨セキュリティ型での暗号スイートの要求設定（基本）

グループ A	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x00,0x9E)
	TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x7C)
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x00,0x67)
	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0x00,0xBE)
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x00,0x33)
	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x00,0x45)
グループ B	TLS_RSA_WITH_AES_128_GCM_SHA256 (0x00,0x9C)
	TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x7A)
	TLS_RSA_WITH_AES_128_CBC_SHA256 (0x00,0x3C)
	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0x00,0xBA)
	TLS_RSA_WITH_AES_128_CBC_SHA (0x00,0x2F) （RFC 必須）
	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x00,0x41)
グループ C	該当なし
グループ D	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x00,0x9F)
	TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0, 0x7D)
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x00,0x6B)
	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256 (0x00,0xC4)
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x00,0x39)
	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x00,0x88)
グループ E	TLS_RSA_WITH_AES_256_GCM_SHA384 (0x00,0x9D)
	TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x7B)
	TLS_RSA_WITH_AES_256_CBC_SHA256 (0x00,0x3D)
	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256 (0x00,0xC0)
	TLS_RSA_WITH_AES_256_CBC_SHA (0x00,0x35)
	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x00,0x84)
グループ F	該当なし
設定すべき鍵長	鍵交換で DHE を利用する場合には鍵長 1024 ビット以上 、RSA を利用する場合には鍵長 2048 ビット以上の設定を必須とする。なお、DHE の鍵長を明示的に設定できない製品を利用する場合には、DHE を含む暗号スイートは選定すべきではない
推奨セキュリティ型での除外事項	グループ A～グループ F 及び表 13 以外のすべての暗号スイートを利用除外とする

[27] TLS1.1 及び TLS1.0 でのサポートが必須と指定されている暗号スイートは Triple DES を利用するものである。しかし、推奨セキュリティ型を適用する SSL/TLS サーバが接続相手として対象とするブラウザは、BEAST 攻撃等に対するセキュリティパッチが適用されているブラウザであることを考慮すれば、AES が利用可能であり、6.5.2 節の設定であっても事実上問題がないと判断した

パテントリスクについても検討したうえでECDHやECDSAを採用することを決めた場合には、表 13 の暗号スイートグループを追加してよい。

表 13　推奨セキュリティ型での暗号スイートの要求設定（楕円曲線暗号の追加分）

グループ A への追加または代替	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xC0,0x2B)
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xC0,0x2F)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x86)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x8A)
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xC0,0x23)
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC0,0x27)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256 (0xC0,0x72)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0xC0,0x76)
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xC0,0x09)
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC0,0x13)
グループ C への追加	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 (0xC0,0x2D)
	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 (0xC0,0x31)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x88)
	TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256 (0xC0,0x8C)
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 (0xC0,0x25)
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 (0xC0,0x29)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256 (0xC0,0x74)
	TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256 (0xC0,0x78)
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xC0,0x04)
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xC0,0x0E)
グループ D への追加または代替	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xC0,0x2C)
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xC0,0x30)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x87)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x8B)
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xC0,0x24)
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC0,0x28)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384 (0xC0,0x73)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384 (0xC0,0x77)
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xC0,0x0A)
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC0,0x14)

推奨セキュリティ型での暗号スイートの要求設定（楕円曲線暗号の追加分）（続）

グループ F への追加	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 (0xC0,0x2E)
	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 (0xC0,0x32)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x89)
	TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384 (0xC0,0x8D)
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 (0xC0,0x26)
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 (0xC0,0x2A)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384 (0xC0,0x75)
	TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384 (0xC0,0x79)
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xC0,0x05)
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xC0,0x0F)
設定すべき鍵長	鍵交換で ECDHE または ECDH を利用する場合には鍵長 256 ビット以上の設定を必須とする

6.5.3 セキュリティ例外型での暗号スイートの詳細要求設定

6.1 節の条件を踏まえて、表 14 の通り、選定した暗号スイートをグループ A、グループ B、・・・とグループ分けをする。グループ分けの基準は安全性と実用性とのバランスの観点に立って行い、優先設定する順番にグループ A から順に割り当てる。

グループ A からグループ F までは推奨セキュリティ型と同様であるので、6.5.2 節を参照のこと。セキュリティ例外型では、推奨セキュリティ型に加え、グループ G とグループ H として、以下の暗号スイートグループを追加する。グループ内での暗号スイートから全部または一部を選択して設定するが、その際の優先順位は任意に定めてよい。

（RFC 必須）は、TLS1.2、TLS1.1 及び TLS1.0 を規定する RFC においてサポートが必須と指定されている暗号スイートであり、不特定多数からのアクセスを想定する SSL/TLS サーバにおいては利用可に設定すべき暗号スイートである。

また、「除外事項」は設定で除外すべき暗号スイートを示したものである。

表 14　セキュリティ例外型での暗号スイートの要求設定（基本）

グループ A～ グループ F	推奨セキュリティ型と同じ　（6.5.2 節参照）
グループ G	TLS_RSA_WITH_RC4_128_SHA (0x00,0x05)
グループ H	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x00,0x16)
	TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x00,0x0A)　（RFC 必須）
設定すべき鍵長	鍵交換で DHE を利用する場合には鍵長 1024 ビット以上 、RSA を利用する場合には鍵長 2048 ビット以上の設定を必須とする。なお、DHE の鍵長を明示的に設定できない製品を利用する場合には、DHE を含む暗号スイートは選定すべきではない
セキュリティ例外型での除外事項	グループ A～グループ G 及び表 13 以外のすべての暗号スイートを利用除外とする

【コラム③】　輸出規制時代の名残－FREAK 攻撃

FREAK[28]攻撃 は、中間者攻撃に分類されるなかでも特にダウングレード攻撃と呼ばれる攻撃手法の一種で、SSL/TLS で利用する暗号スイートを「RSA を利用する輸出規制対象の暗号スイート（RSA_EXPORT)」に強制的にダウングレードさせる攻撃である。

RSA_EXPORT は、2000 年前後まで続いていた輸出規制に対応するためのもので、あえて暗号強度を弱める処理を行う。具体的には、たとえサーバ証明書で鍵長 2048 ビットの RSA を使って鍵交換をするように記載されていても、強制的に暗号強度を大きく弱めた鍵長 512 ビットの RSA を利用して鍵交換をするように制御する。こうすることで、鍵交換での RSA が解読できればセッション鍵を取り出すことができるため、当該 SSL/TLS 通信を復号することが可能になる。

発見者によれば、鍵長 512 ビットの RSA は Amazon EC2 で 100 ドル出せば 12 時間以内に解読できると主張している。実際、鍵長 768 ビットの RSA の解読事例が 2010 年に発表されていることを考慮すれば、鍵長 512 ビットの RSA が簡単に解読されたとしてもおかしくはない。

FREAK 攻撃が成功するためには、少なくともサーバが RSA_EXPORT を受け付ける設定になっている必要がある。一方、本ガイドラインの要求設定では、「高セキュリティ型」「推奨セキュリティ型」「セキュリティ例外型」のいずれにおいても EXPORT を使う暗号スイートは利用除外とするようになっているため、RSA_EXPORT が選択されることはなく、FREAK 攻撃はもともと成功しない。

なお、今では RSA_EXPORT を必要とする機会はほとんどないことから、サーバ・ブラウザともに、デフォルトでは RSA_EXPORT を受け付けないようにするためのセキュリティパッチがベンダ各社から提供されている。

[28] Factoring RSA Export Keys

7. SSL/TLS を安全に使うために考慮すべきこと

プロトコルとしての脆弱性だけでなく、実装上の脆弱性が発見されることも時おり起きる。

そのような脆弱性が発見されると基本的にはベンダからセキュリティパッチが提供されるので、ベンダが提供するセキュリティパッチを入手可能な状態とし、常にセキュリティパッチを適用して最新の状態にしておくことが望ましい。

それ以外にも、SSL/TLS をより安全に使うために、以下の項目を参考にするとよい。

7.1 サーバ証明書の作成・管理について注意すべきこと

7.1.1 サーバ証明書での脆弱な鍵ペアの使用の回避

OpenSSL などの暗号モジュールにおいて擬似乱数生成機能のエントロピー不足などの脆弱性が存在する場合、これを用いて鍵配送・共有や署名で使う公開鍵と秘密鍵の鍵ペアを生成した際に、結果的に解読容易な鍵ペアが生成されてしまうリスクがある。

こうしたリスクを防ぐためには、サーバ管理者は、鍵ペアの生成時点で脆弱性が指摘されていない暗号モジュールを利用するよう注意すべきである。また、既知の解読可能な鍵ペアでないことを確認するサービスなども提供されている [29]。

7.1.2 推奨されるサーバ証明書の種類

ブラウザなどをはじめとするサーバ証明書を検証するアプリケーションには、一定の基準に準拠した認証局の証明書（ルート CA 証明書）があらかじめ登録されており、これらの認証局（とその下位認証局）はパブリック認証局と呼ばれている。一般に、パブリック認証局が、第三者の立場から確認したサーバの運営組織等の情報を記載したサーバ証明書を発行し、ブラウザに予め搭載されたルート CA 証明書と組合せて検証を行うことで、サーバ証明書の信頼性を確保する。これにより、当該サーバ証明書の正当性が確認できれば、ブラウザは警告表示することなく、当該サーバへの接続を行う。

パブリック認証局から発行されるサーバ証明書は、その用途や利用範囲に応じて表 15 に示す 3 種類に分類される。これらのサーバ証明書のうち、不特定多数の利用者がアクセスする一般的な Web サーバ用途であれば、運営サイトの法的実在性の確認やグリーンバーによる視認性の高さといった優位点がある EV 証明書が利用者にとって一番安心できるサーバ証明書といえる。しかし、本ガイドライン公開時点（2015 年 5 月）においては、スマートフォンなど一部の機器においてまだ十分にグリーンバーが機能しているとは言い難く、また入手コストにおいて OV 証明書とのギャップが大きい、といった課題もある。

そこで、本ガイドラインでは、不特定多数の利用者がアクセスする一般的なサーバ用途について、EV 証明書の利用を推奨するに留める。

[29] 例えば https://factorable.net/keycheck.html がある。ただし、安全性を 100%証明するものではないことに注意されたい

表 15　サーバ証明書の種類と違い

サーバ証明書の種類	内容の違い
DV 証明書 (Domain Validation)	サーバの運営組織が、サーバ証明書に記載されるドメインの利用権を有することを確認したうえで発行される証明書。 オンライン申請による短時間発行や低コストで入手できるものが多い、などのメリットがある。 一方、サーバの運営組織の実在性や、ドメイン名と運営組織の関係については確認されないので、不特定の利用者を対象とする一般的な Web サーバの用途には不向きである。
OV 証明書 (Organization Validation)	ドメイン名の利用権に加えて、サーバ運営組織の実在性の確認やドメイン名と運営組織との関係などについても確認した上で発行される証明書。 不特定多数の利用者がアクセスするような一般的な Web サーバの用途で利用されるが、①現状では利用者がブラウザで OV 証明書と DV 証明書を明確に識別することは難しい、②サーバ運営組織等の確認項目や確認方法は個々の認証局によって異なる、という課題もある。
EV 証明書 (Extended Validation)	OV 証明書と同様で、ドメイン名の利用権に加えて，サーバ運営組織の実在性等の確認やドメイン名と運営組織との関係などについても確認した上で発行される証明書。 3 つの証明書のなかでは発行コストが最もかかるが、以下の点で DV 証明書や OV 証明書に対して優位点を持つ。 ● 運営組織の法的実在性について、CA/Browser Forum が規定した国際的な認定基準にもとづいて確認が行われる。このため認証局に依らず一定レベルの確認が保証される ● ブラウザのアドレス表示部分等が緑色になる「グリーンバー」機能が有効に機能する場合には、利用者にとって EV 証明書であることの識別が容易 ● グリーンバーには運営組織も表示されるため，ドメイン名との関係が一目でわかる

7.1.3　サーバ証明書の有効期限

サーバ管理者は、サーバ証明書の更新漏れによって自社のサービスに障害を発生させることがないように、サーバ証明書の有効期間を管理し、更新作業のために必要なリードタイムを考慮した上で、適切な管理方法（例えば、更新作業開始時期の明文化など）を定めることが求められる。

市販されているサーバ証明書の有効期間は、半年や 1 年程度のものから、2 年、3 年程度のもの等様々である。一般に、有効期間が長いほど、サーバ証明書の更新頻度が少なく更新作業の工数を削減できる。しかし、その反面、単純なミスによる更新忘れ、組織改編・担当者異動時の引き継ぎ不備による更新漏れ、鍵危殆化（秘密鍵の漏えい）リスクの増大、サーバ証明書に記載されたサーバの運営組織情報が（組織名変更などにより）正確でなくなるリスクの増大、アルゴリズム Agility（セキュリティ強度の変化に対して、安全な側に移行するための対策に要する時間、迅

速さの程度）の低下などが危惧されるようになる。特に、2 年や 3 年など比較的長い間有効なサーバ証明書を利用する場合には、管理者がサーバ証明書の有効期限切れに気づかず、更新漏れによるサービス障害の発生が大きなリスクとなりえる。

これらを総合的に勘案し、特段の制約が存在しない限り、サーバ管理者は、1 年程度の有効期間を持つサーバ証明書を選択し、サーバ証明書の更新作業を、年次の定型業務と位置付けることが望ましい。

なお、SHA-1 を利用しているサーバ証明書に関しては、クライアントにおいて SHA-256 への対応が進み、SHA-1 でなくても運用上の支障がなくなった場合に、速やかに SHA-256 を利用しているサーバ証明書への移行ができるようにするため、有効期間をできるだけ短く設定することが望ましい。

7.1.4 サーバ鍵の適切な管理

サーバ管理者は、サーバ証明書に対応する秘密鍵について、紛失、漏えい等が発生しないように適切に管理しなければならない。秘密鍵の紛失（データ破壊を含む）に備えバックアップを作成し保管する場合には、秘密鍵の危殆化[30]（漏えいなど）が発生しないようにするために、バックアップの方法や保管場所、その他の保管の要件について注意深く設計することが求められる。

サーバ管理者は、秘密鍵が危殆化した際に遅滞なく適切な対処を行うため、必要に応じて、次のような事項について、あらかじめ、方針及び手順を整理し文書化することが推奨される。

- 秘密鍵の危殆化に対応するための体制（関係者と役割、委託先との連携を含む）
- 秘密鍵が危殆化した、またはその恐れがあると判断するための基準
- 秘密鍵の危殆化の原因を調べること、及び、原因の解消を図ること
- 当該サーバ証明書の利用を停止すること（実施の判断基準、手順を含む）
- 当該サーバ証明書を遅滞なく失効すること（実施の判断基準、手順を含む）
- 新しい鍵ペアを生成し、新鍵に対して新しくサーバ証明書の発行を行うこと
- 秘密鍵の危殆化についての情報の開示（通知先、通知の方法、公表の方針等）

7.1.5 複数サーバに同一のサーバ証明書を利用する場合の注意

負荷分散や冗長化による可用性向上などを目的として複数のサーバに同一のサーバ証明書をインストールして利用する場合、サーバ管理者は、以下の観点で注意が必要である。

- サーバ証明書の更新や再発行の際には、入替作業の対象となる全てのサーバについて漏れなく証明書をインストールしなおすこと
- サーバ証明書の入替えに伴って暗号通信に関わる設定（4 章から 7 章までを参照）の変更を行う場合は、対象となる全てのサーバに漏れなく適用すること

[30] 安全性上の問題が生じ、信用できなくなる状態のこと

サーバ管理者は、サーバ証明書の入替作業の対象となるサーバに漏れが発生しないよう、サーバ毎にペアとなる秘密鍵や暗号スイートなどの情報を一覧管理し、また外部からの監視／スキャンツールを用いたチェックと組合せるなど、管理方法を定め文書化することが推奨される。

7.1.6 ルート CA 証明書

サーバ証明書の安全性は、その証明書を発行する認証局自体の安全性はもとより、最終的には信頼の起点（トラストアンカー）となる最上位の認証局（ルート CA）の安全性に依拠している。

このことは、ルート CA の用いる暗号アルゴリズムおよび鍵長の安全性が十分でなければ、サーバ証明書の安全性も確保することができないことを意味している。例えば、ルート CA 証明書の安全性に問題が生じ、ブラウザベンダなどが当該ルート CA 証明書を失効させた場合、サーバ証明書自体には問題がなかったとしてもルート CA 証明書とともに失効することとなる。

このようなリスクを避けるためには、サーバ管理者は、信頼の起点（トラストアンカー）となるルート CA についても、当該サーバ証明書と同様の安全性を満たすルート CA 証明書を発行しているルート CA を選ぶべきである。ルート CA 証明書で利用している暗号アルゴリズムおよび鍵長の確認方法については、Appendix D.1 を参照されたい。

【コラム④】　DigiNotar 認証局事件

2011 年 8 月、オランダの認証局事業者 DigiNotar 社によって多数のサーバ証明書が不正に発行されていることが発覚した。本事件は、主要なブラウザベンダによって同社のルート CA 証明書を無効化する緊急パッチが提供された点、ならびに不正発行の規模が過去に類を見ないという 2 点において関心を集めた象徴的な事件となった。

同社はパブリックルート認証局として主にオランダ国内を市場として証明書を発行していたが、2011 年 6 月に同社の認証局システムが攻撃者によって侵入され、1 ヶ月以上に渡る遠隔操作により少なくとも 531 枚以上の不正な証明書が発行された。これらの証明書は、イラン国内から Google 社の Gmail サービスへのアクセスに対する中間者攻撃等に悪用された。このような事態を踏まえ、同年 9 月には同社ルート CA 証明書が主要なブラウザから無効化されることとなった。

ルート CA 証明書が無効化された場合、その認証局が発行する証明書を利用する Web サーバへの影響は避けられない。このような場合、サーバ管理者は他の認証局からサーバ証明書を早急に取得しなおすことが必要となる。

世界の認証局事業者は、サーバ証明書やルート CA 証明書に用いる暗号アルゴリズムのみならず、認証局システム自体のセキュリティを維持・向上させるための対策を含む業界基準を制定し（CA/ブラウザフォーラムによる「Baseline Requirement」等）、こうした事件の再発防止を図っている。

7.2 さらに安全性を高めるために

7.2.1 HTTP Strict Transport Security（HSTS）の設定有効化

例えばオンラインショッピングサイトのトップページが暗号化なしの HTTP サイトで、ショッピングを開始する際に HTTPS へリダイレクトされるような構成になっていた場合、リダイレクトを悪意のあるサイトに誘導し、情報を盗むといった中間者攻撃が SSL strip というツールを用いて可能であるという報告が Moxie Marlinspike によってなされた。

この攻撃に対して、HTTP で接続したら、すぐに強制的に HTTPS サイトへリダイレクトし、以降の通信は全て HTTPS とすることによって防御する技術が RFC 6797 で規定されている HTTP Strict Transport Security（HSTS）である。

HSTS に対応した SSL/TLS サーバに HTTPS でアクセスした場合、HTTPS 応答には以下のような HTTP ヘッダが含まれている。

Strict-Transport-Security:max-age=有効期間秒数;includeSubDomains

このヘッダを受け取った HSTS 対応のブラウザは、有効期間の間は当該サーバへは HTTP ではなく全て HTTPS で通信するように自動設定しておく。これにより、以前接続したときに HSTS が有効になっているサーバであれば、何らかの理由で、ブラウザが HTTP で接続しようとしても自動的に HTTPS に切り替えて接続する。

以上のように、HTTPS で安全にサービスを提供したい場合などでは、ユーザに意識させることなくミスを防止でき、ユーザの利便性を向上させることができるので、HSTS の機能を持っているならば有効にすることを推奨する。参考までに、いくつかの設定例を Appendix B.4 で紹介する。

ただし、HSTS が実際に機能するためには、サーバだけでなく、ブラウザも対応している必要があることに注意されたい。また、一度も接続したことがないサーバ（例外的に Firefox 17 以降ではあらかじめ登録されているサーバもある）や、HSTS の期限切れになったサーバの場合にも、HTTPS への変換は行われない。

2014 年 9 月時点での主要な製品の HSTS へのサポート状況は以下の通りである。

- サーバ
  - Apache 2.2.22 以降：設定により可能
  - Lighttpd 1.4.28 以降：設定により可能
  - nginx 1.1.19 以降：設定により可能
  - IIS：設定により可能
- クライアント（ブラウザ）
  - Chrome：4.0.211.0 以降でサポート
  - Firefox：Firefox 17 以降でサポート
  - Opera：Opera 12 以降でサポート
  - Safari：Mac OS X Mavericks 以降でサポート
  - Internet Explorer：Windows 10 IE 以降でサポート予定

7.2.2 リネゴシエーションの脆弱性への対策

リネゴシエーションとは、サーバとクライアントとの間で暗号アルゴリズムや暗号鍵の設定のために使われる事前通信（ハンドシェイク）において、一度確立したセッションに置き換わる新たなセッションを確立する際に、すでに確立したセッションを使って改めてハンドシェイクを行う機能である。

リネゴシエーションの脆弱性とは、クライアントとサーバの間に攻撃者が入る中間者攻撃によって、通信データの先頭部分に任意のデータを挿入することができるというものである（図 5）。これにより、例えば、攻撃者が挿入した HTTP リクエストを、あたかも正当なユーザから送られたリクエストであるかのようにサーバに誤認させるといったことができる。

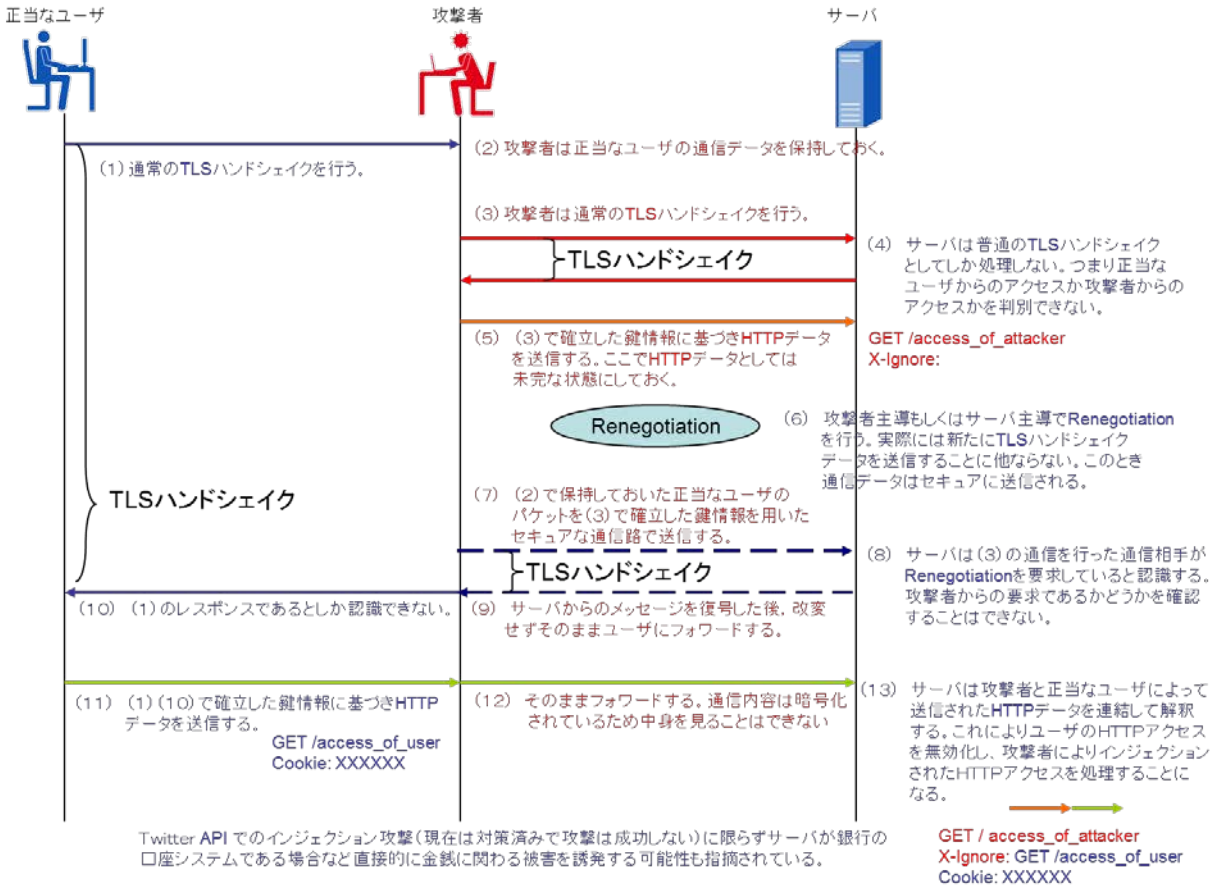


図 5　リネゴシエーションの脆弱性

この脆弱性のポイントは、リネゴシエーションが確立したセッションを使って行われることから、リネゴシエーションの前後の通信が同じ通信相手である、という前提で処理が行われる点にある。ところが、実際に図 5 の（10）で確立したセッションは、クライアントにとって一回目のハンドシェイクで確立したセッション（図 5 の（1）の要求に対するセッション）なのに対して、サーバはリネゴシエーションで確立したセッション（図 5 の（7）の要求に対するセッション）になっている。

それにも関わらず、両者がその食い違いを認識できないため、その結果として、サーバは、リネゴシエーション前の攻撃者からの通信（図 5 の（5）の通信）とリネゴシエーション後のクラ

イアントからの通信（図 5 の（11）（12）の通信）を、同一クライアントからの通信と誤認して受け付けて処理を行うことになり、予期せぬ事態を引き起こす可能性がある。

【推奨対策】

リネゴシエーションに関するプロトコル上の脆弱性であることから、対策としては以下のどちらかの設定とすることを推奨する。

- リネゴシエーションを利用不可とする
- リネゴシエーションの脆弱性対策（RFC5746）を反映したバージョンの製品を利用するとともに、対策が取られていないバージョンの製品からのリネゴシエーション要求は拒否する設定を行う

7.2.3 圧縮機能を利用した実装攻撃への対策

圧縮機能は、何度も出てくる同じ長い文字列を別の短い情報に置き換えることで全体のデータサイズを削減し、通信効率を向上させるために利用するものである。

しかしながら、圧縮対象となる文字列に秘密情報が含まれている場合、圧縮機能によって別の情報に置き換わることによるデータサイズの変動に着目することによって、どの文字列が圧縮されたのかが分かる可能性がある。しかも、着目しているのはデータサイズであるので、データが暗号化されているかどうかは関係がない。

実際にこのような圧縮機能を利用した実装攻撃として、CRIME、TIME、BREACH などがある。これらの攻撃は、SSL/TLS のプロトコル自体の脆弱性ではなく、圧縮機能の特性そのものを利用した攻撃方法である。したがって、根本的な対策としては「SSL/TLS では圧縮機能を利用しない」こと以外に方法はない。

このため、最近のバージョンの OpenSSL や Windows などでは、デフォルト設定がオフになっていたり、そもそもサポートを取りやめたりしている。

7.2.4 OCSP Stapling の設定有効化

サービス提供の終了やサーバの秘密鍵の漏えいなど、何らかの理由で、サーバ証明書の有効期間内であっても当該サーバ証明書が失効している場合がある。そのため、サーバ証明書の正当性を確認する時には、当該サーバ証明書が失効していないかどうかもあわせて確認すべきである。

サーバ証明書が失効されていないか確認する方法として、CRL[31]と OCSP[32]の二つの方法があるが、CRL はサイト数の増大に伴ってファイルサイズが増大しており、近年では OCSP のみに依存するブラウザが多くを占めている。

ただ、OCSP を使用した場合には 2 つの問題がある。

1) OCSP 実行時の通信エラー処理について明確な規定がなく、ブラウザの実装に依存する。このため、OCSP レスポンダの通信障害等で適切な OCSP 応答が得られない場合にサーバ証明書の失効検証を正しく行わないまま SSL 通信を許可してしまうブラウザも少なくな

[31] Certificate Revocation List
[32] Online Certificate Status Protocol

い。そのようなブラウザに対しては、あるサイトのサーバ証明書が失効していたとしても、DDoS 攻撃などにより意図的に OCSP レスポンダに接続させないことにより、当該サイトが有効であるとして SSL/TLS 通信をさせることができる

2) OCSP を使った場合には、あるサイトにアクセスがあったことを OCSP レスポンダも知り得てしまうため、プライバシー上の懸念がある。例えば、ある利用者が、ある会員制のサイトにアクセスした場合、ブラウザはサーバ証明書の失効検証のために当該サイトの OCSP 応答を取得する。そこで、OCSP レスポンダのアクセス履歴から、ある接続元 IP の利用者は、当該サイトの会員であると OCSP レスポンダが知り得ることになる

上記の問題を解決するために、RFC 6066 Transport Layer Security (TLS) Extension: Extension Definition の 8 節で、Certificate Status Request という TLS 拡張が規定されている。これを使うことにより、OCSP 応答を OCSP レスポンダからではなく、アクセス先サイトの Web サーバから取得して SSL/TLS 通信を開始することができる。

- OCSP レスポンダからの OCSP 応答を Web サーバがキャッシュしている限り、ブラウザは OCSP 応答による失効検証を行うことができる
- OCSP 応答を、OCSP レスポンダからではなく、Web サーバから取得するので、当該サイトへのアクセス履歴を OCSP レスポンダが知ることはない

なお、OCSP Stapling は 2014 年 9 月時点で以下の環境においてサポートされている。参考までに、いくつかの設定例を Appendix B.5 で紹介する。

- サーバ
  - Apache HTTP Server 2.3.3 以降
  - nginx 1.3.7 以降
  - Microsoft IIS on Windows Server 2008 以降

  など

- クライアント（ブラウザ）
  - Mozilla Firefox 26 以降
  - Microsoft Internet Explorer （Windows Vista 以降）
  - Google Chrome

  など

7.2.5 Public Key Pinning の設定有効化

近年、FLAME 攻撃や、DigiNotar、TURKTRUST などの認証局からのサーバ証明書の不正発行など、偽のサーバ証明書を使った攻撃手法が増加傾向にある。これらの攻撃により発行されたサーバ証明書は、認証局が意図して発行したものではないという意味で“偽物”であるが、動作そのものは“本物”と同じふるまいをする。

このため、この種の攻撃に対しては、従来の PKI による、信頼するルート証明書のリストと、

証明書チェーンの検証（認証パス検証）だけでは正当なサーバ証明書であるかどうかの判断がつかない。

これを補う目的で導入されつつあるのが、Public Key Pinning（もしくは Certificate Pinning）と呼ばれている技術である。従来の PKI による証明書チェーンの検証に加え、Public Key Pinning では、あるサイト用に期待されるサーバ証明書の公開鍵情報（SPKI; Subject Public Key Info）フィールドの情報のハッシュ値を比較することにより、当該サーバ証明書が正当なものであるかどうかを判断する。

2014 年 9 月時点で、Public Key Pinning をサポートしている環境は以下の通りである。

- サーバ
  - HTTP ヘッダを追加可能な任意のサーバ

- クライアント
  - Google Chrome 13 以降
  - Mozilla Firefox 32 以降（デスクトップ版）、34 以降（Android 版）
  - Internet Explorer： マイクロソフト脆弱性緩和ツール（EMET[33]） を導入することで設定可能（EMET バージョン 4.0 以降よりサポート）

期待されるハッシュ値の提供方法には 2 通りある。

1) ブラウザのソースコードに主要なサイトの SPKI フィールドの情報のハッシュ値リストを保持し、これと比較して SSL サーバ証明書が正当であるかを調べるもの。2014 年 9 月時点では Google Chrome や Mozilla Firefox がサポートしている。

2) サイトから送られる HTTP ヘッダに含まれる、SSL サーバ証明書の SPKI フィールドの情報のハッシュ値を元に正当性を比較するもの。IETF において、Public Key Pinning Extension for HTTP として発行された。参考までに、いくつかの設定例を Appendix B.6 で紹介する。

[33] http://technet.microsoft.com/ja-jp/security/jj653751

PART II：

ブラウザ＆リモートアクセスの利用について

8. ブラウザを利用する際に注意すべきポイント

8.1 本ガイドラインが対象とするブラウザ

8.1.1 対象とするプラットフォーム

ベンダがセキュリティホールに対する修正を行っている OS を利用すべきである。本ガイドラインの公開時点（2015 年 5 月）で、サポート対象となっているものは以下の通りである。

- デスクトップ向け OS
  - Windows Vista Service Pack 2 （2017 年 4 月 11 日サポート終了）
  - Windows 7 Service Pack 1 （2020 年 4 月 11 日サポート終了）
  - Windows 8 （2016 年 1 月 12 日サポート終了）
  - Windows 8.1 （2023 年 1 月 10 日サポート終了）
  - Mac OS X 10.9

- スマートフォン向け OS
  - 当該端末で利用できる最新の Android（もっとも古いもので Android4.x）
  - iOS 8

8.1.2 対象とするブラウザのバージョン

ブラウザは､少なくとも提供ベンダがサポートしているバージョンのものを利用すべきである。本ガイドラインの公開時点（2015 年 5 月）でサポートしている、8.1.1 節に挙げた OS 上で動作するブラウザのバージョンは以下のとおりである。

- Microsoft Internet Explorer
  2016 年 1 月 12 日以降は、サポートされるオペレーティングシステムで利用できる最新バージョンの Internet Explorer のみがテクニカルサポートとセキュリティ更新プログラムを提供されるようになる（表 16)。詳細は、以下を参照のこと。

  Microsoft Internet Explorer サポート ライフサイクル ポリシーに関する FAQ
  http://support2.microsoft.com/gp/microsoft-internet-explorer

- Microsoft Internet Explorer 以外のブラウザ
  - Apple Safari 最新版
  - Google Chrome 最新版
  - Mozilla Firefox 最新版
  - Mobile Safari（iOS）: iOS 8 に搭載する Mobile Safari

表 16　Internet Explorer のサポート期間

ブラウザバージョン	OSバージョン	サポート期間（ライフサイクルポリシー@2014年11月10日時点）
		2015 / 2016 / 2017 / 2018 / 2019 / 2020 / 2021 / 2022 / 2023
Internet Explorer 7	Windows Vista SP2	2016/1/12
Internet Explorer 8	Windows Vista SP2	2016/1/12
	Windows 7 SP1	2016/1/12
Internet Explorer 9	Windows Vista SP2	2017/4/11
	Windows 7 SP1	2016/1/12
Internet Explorer 10	Windows 7 SP1	2016/1/12
	Windows 8	2016/1/12
Internet Explorer 11	Windows 7 SP1	2020/1/14
	Windows 8.1	2023/1/10

8.2 設定に関する確認項目

8.2.1 基本原則

8.1 節で対象とするブラウザは、インストール時のデフォルト設定で利用することを各ベンダは推奨しているので、企業の情報システム担当からの特別な指示がある場合などを除き、原則としてデフォルト設定を変えずに利用することを強く推奨する。

【基本原則】

- ベンダがサポートしているブラウザであって、更新プログラムを必ず適用する（Internet Explorer の場合）、または最新バージョンのブラウザを利用する（Internet Explorer 以外）
- 自動更新を有効化しておく
- 企業の情報システム担当からの特別な指示がある場合などに限り、社内ポリシーに従う

8.2.2 設定項目

設定項目を標準機能で提供していないブラウザ

以下のブラウザは、設定変更オプションが提供されておらず、そもそも設定変更ができない。

- PC 版 Web ブラウザ
  - Apple Safari
  - Google Chrome

- スマートフォンに含まれる Web ブラウザ
  - Android 標準ブラウザ
  - Mobile Safari (iOS)

設定項目を標準機能で提供しているブラウザ

以下のブラウザは、設定変更オプションが提供されている。ただし、特別な指示がない限り、デフォルト設定を変更すべきではない。

- Microsoft Internet Explorer

  他のブラウザとは異なり、Internet Explorer では、

  “ツール”→“インターネットオプション”→“詳細設定”

  を選択すると多数の設定項目が表示され、ユーザが細かく設定できるようになってはいる。しかし、安全性を考慮してデフォルト設定が行われていることから、特段の理由がない場合には“プロトコルバージョンの設定を除いて”設定を変更することは推奨しない。

  なお、Internet Explorer のセキュリティ機能及びデフォルト設定については、以下に一覧としてまとめられている。

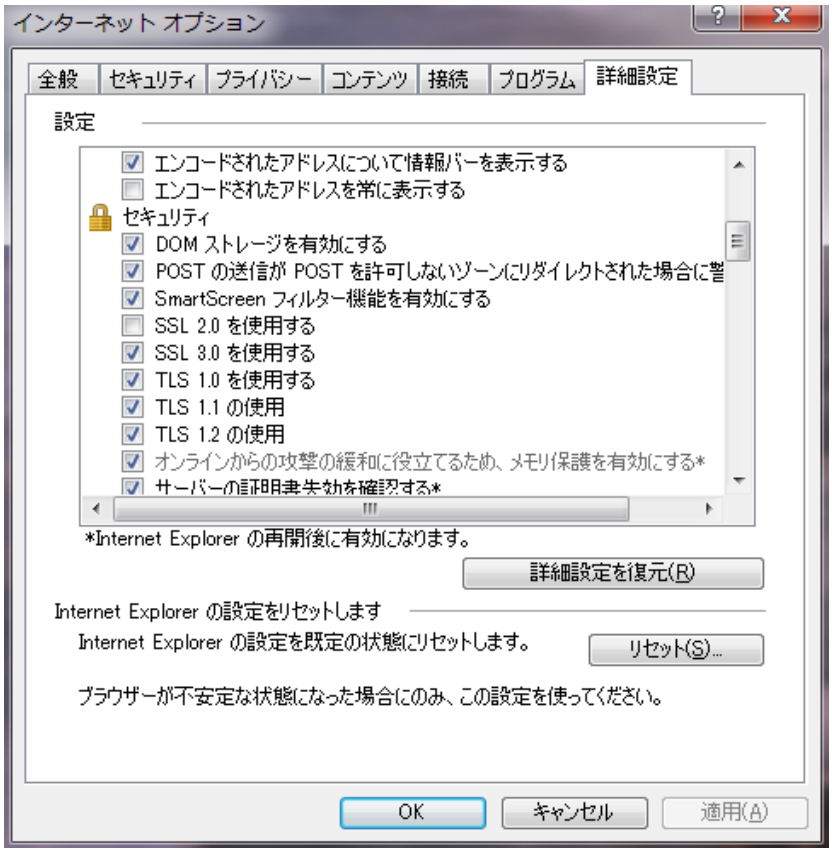
  バージョン別 IE のセキュリティ機能
  http://msdn.microsoft.com/ja-jp/ie/cc844005.aspx

  【プロトコルバージョンの設定】
  “ツール”→“インターネットオプション”→“詳細設定”を選択した後、設定項目を“セキュリティ”までスクロールさせると、「SSL2.0 を使用する」「SSL3.0 を使用する」「TLS1.0 を使用する」「TLS1.1 を使用」「TLS1.2 を使用」といったチェックボックスが表示される。ここでのチェックボックスにチェックが入っているプロトコルバージョンが、ブラウザが使うことができるプロトコルバージョンとなる。
  本ガイドライン公開時点（2015 年 5 月）のデフォルト設定では、IE6 では「SSL2.0 を使用する」にチェックが入っている一方、IE8 以降では TLS1.1 や TLS1.2 をサポートしているものの「TLS1.1 を使用」「TLS1.2 を使用」にはチェックが入っていない。

  このように、Internet Explorer は使うバージョンによって利用できるプロトコルバージョンが異なるので、プロトコルバージョンについてのみ、適切な設定になっているかを確認し、必要に応じて設定変更することを推奨する。

	TLS1.2	TLS1.1	TLS1.0	SSL3.0	SSL2.0
IE6（参考）	×	×	▲	○	○
IE7	×	×	○	○	▲
IE8	▲	▲	○	○	▲
IE9	▲	▲	○	○	▲
IE10	▲	▲	○	○	▲
IE11	▲	▲	○	○	▲

○：デフォルト設定 ON　▲：デフォルト設定 OFF　×：サポートしていない



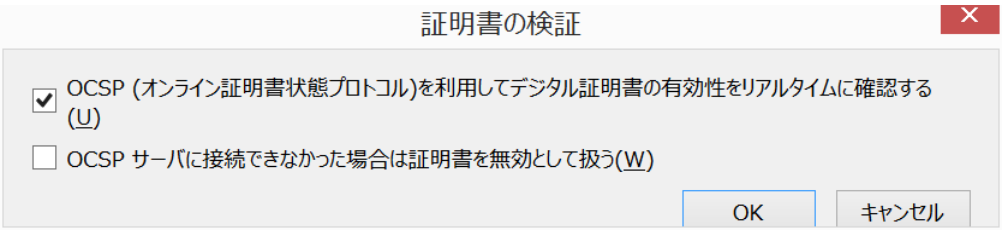
- Firefox

  Firefox では、サーバ証明書の検証、失効機能においてどのように処理するかの動作についてのみ設定方法を提供している。この設定については、

  “メニュー”→“オプション”→“詳細”→“証明書”→“検証(V)…”

  を選択することで設定方法へのダイアログが表示される。

  デフォルトの設定は以下のようになっており、特段の理由がない場合に変更することは推奨しない。



8.3 ブラウザ利用時の注意点

8.3.1 鍵長 1024 ビット、SHA-1 を利用するサーバ証明書の警告表示

CA/Browser Forum にて、サーバ証明書の有効期限が 2014 年 1 月 1 日以降の場合、RSA の鍵長

を最小 2048 ビットにすると決められている。このため、ブラウザベンダ各社では、RSA の鍵長が 2048 ビット未満のものは順次無効にする対処がされている。また、SHA-1 についても、順次無効化する対処が予定されている。

詳しくは以下のとおりである。

- Microsoft Internet Explorer
  2017 年 1 月 1 日より SHA-1 で署名されたサーバ証明書を受け付けない [34]。詳細は別途追記予定

- Google Chrome
  Chrome 39 より順次、SHA-1 で署名されたサーバ証明書については、アドレスバーの鍵アイコンが別表記になる [35, 36]。以下のようにサーバ証明書の有効期限によって表記は変化する。

バージョン	サーバ証明書の有効期限	アドレスバーの鍵アイコンの表記
39	2017 年 1 月 1 日以降	黄色い三角アイコン
40	2016 年 6 月 1 日～12 月 31 日	黄色い三角アイコン
	2017 年 1 月 1 日以降	HTTP と同様の表示
42	2016 年 1 月 1 日～12 月 31 日	黄色い三角アイコン
	2017 年 1 月 1 日以降	赤い×アイコン

- Firefox
  2014 年以降、SSL/TLS で利用される RSA の鍵長が 2048 ビットに満たないルート証明書は順次無効になり、2015 年の中頃までにはすべてで無効になる [37]。
  また SHA-1 で署名されたサーバ証明書についても、2015 年以降にリリースされる最新版の Firefox では、以下のように変更をする予定である [38]。

バージョン	サーバ証明書の有効期限	アドレスバーの鍵アイコンの表記
2015 年以降のバージョン	2017 年 1 月 1 日以降	警告表示をする UI を追加
2016 年以降のバージョン	2017 年 1 月 1 日以降	“接続の安全性を確認できません”と表示
2017 年以降のバージョン	すべて	“接続の安全性を確認できません”と表示

[34] http://blogs.technet.com/b/pki/archive/2013/11/12/sha1-deprecation-policy.aspx
[35] http://blog.chromium.org/2014/09/gradually-sunsetting-sha-1.html
[36] https://groups.google.com/a/chromium.org/forum/#!topic/security-dev/QNVVo4_dyQE
[37] https://wiki.mozilla.org/CA:MD5and1024
[38] https://blog.mozilla.org/security/2014/09/23/phasing-out-certificates-with-sha-1-based-signaturealgorithms/

8.3.2 SSL3.0 の取り扱い

POODLE 攻撃の公表を受け、各ブラウザベンダは順次 SSL3.0 を利用不可とする対処を取り始めている。

- Internet Explorer
  セキュリティ情報 MS15-032「Internet Explorer 用の累積的なセキュリティ更新プログラム（3038314)」により、Internet Explorer 11 では SSL3.0 がデフォルトで無効になっている。それ以外のバージョンの Internet Explorer では、設定を変更することにより、SSL3.0 を無効化することができる。詳しくは、下記 URL のマイクロソフトセキュリティアドバイザリを参照のこと。

  マイクロソフト セキュリティ アドバイザリ 3009008
  https://technet.microsoft.com/ja-jp/library/security/3009008.aspx

- Google Chrome
  Chrome 40 からデフォルトで SSL3.0 が無効化されている。

- Firefox
  Firefox 34 および Firefox ESR 31.3.0 からデフォルトで SSL3.0 が無効化されている。

9. その他のトピック

9.1 リモートアクセス VPN over SSL （いわゆる SSL-VPN）

SSL-VPN と呼ばれるものは、正確には SSL を使った“リモートアクセス VPN”の実現方法といえる。SSL-VPN 装置を介して SSL-VPN 装置の奥にあるサーバ（インターネットからは直接アクセスできないサーバ）とクライアント端末をつなぐ形での VPN であり、IPsec-VPN のような特定端末間だけで VPN を構成する、いわゆる拠点間 VPN とは異なる。

したがって、あくまでリモートアクセスでの通信経路上が SSL/TLS で保護されているにすぎないと考え、本ガイドラインの推奨セキュリティ型（または高セキュリティ型）の設定を適用することとし、Appendix A.3（または Appendix A.2）のチェックリストを用いて確認すべきである。

なお、一口に SSL-VPN といっても、実現形態が製品によって全く異なることに注意がいる。実現形態としては、大きく以下の 3 通りに分かれる。

- 通常のブラウザを利用する“クライアントレス型”
- 接続時に自動的に Java や Active X をインストールすることでブラウザだけでなく、アプリケーションでも利用できるようにした“on-demand インストール型”
- 専用のクライアントソフト（通信アダプタなどを含む）をインストール・設定してから利用する“クライアント型”がある。

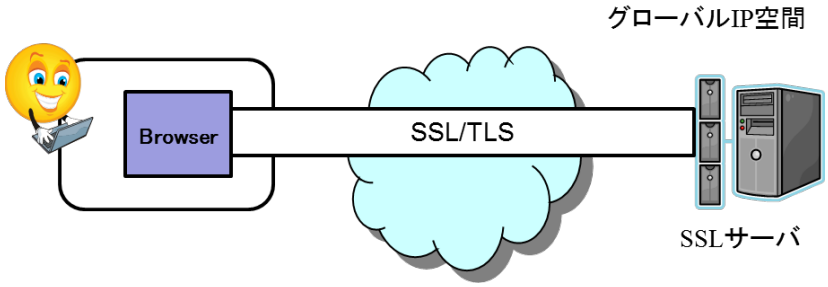
クライアントレス型は、ブラウザさえあればどの端末からでもアクセス可能であり、利便性に優れる一方、SSL との最大の差はグローバル IP をインターネットに公開しているか否か程度の違いといえる。結果として、最初のクライアント認証を SSL/TLS サーバが受け持つか、SSL-VPN 装置が受け持つか程度の差でしかなく、VPN というよりも、本質的には SSL/TLS と同じものとみるべきである。

On-demand インストール型も、接続時に自動的にインストールされることから、特に利用端末に制限を加えるものではなく、クライアントレス型と大きく異なるわけではない。むしろ、ブラウザでしか使えなかったクライアントレス型を、他のアプリケーションでも利用できるように拡張したという位置づけのものである。

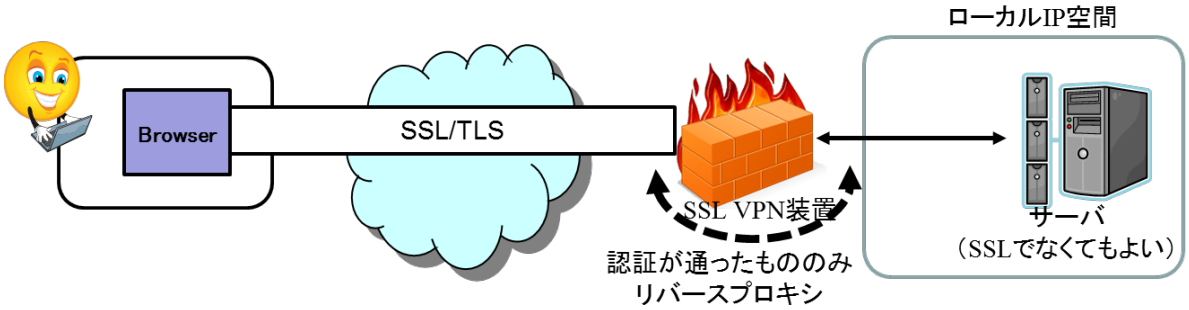
一方、クライアント型は上記の 2 つのタイプとは明らかに異なり、専用のクライアントソフトがインストールされた端末との間でのみアクセスする。つまり、誤って偽サーバに接続することがなく、また内部サーバにアクセスできる端末も厳格に制限できるため、端末に IPsec-VPN ソフトをインストールして構成するモバイル型の IPsec-VPN に近い形での運用形態となる。

機密度の高い情報を扱うのだとすれば、少なくともクライアント型での SSL-VPN を利用すべきである。

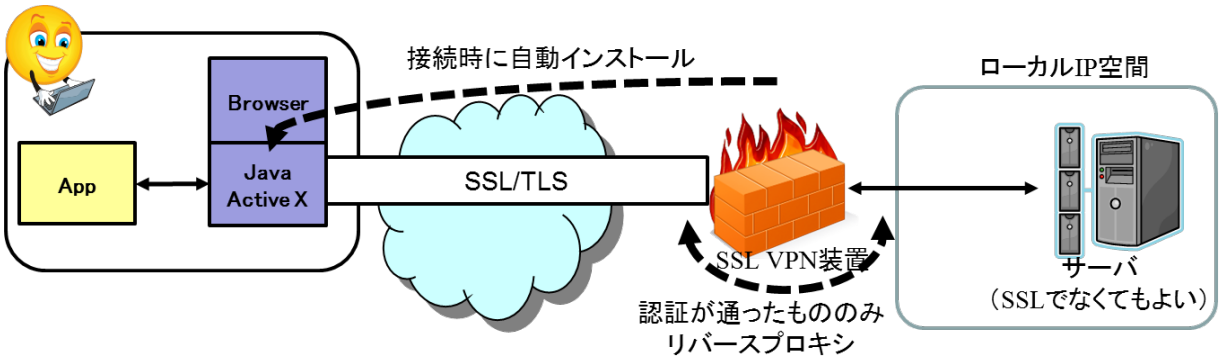
【参考：通常の SSL/TLS】



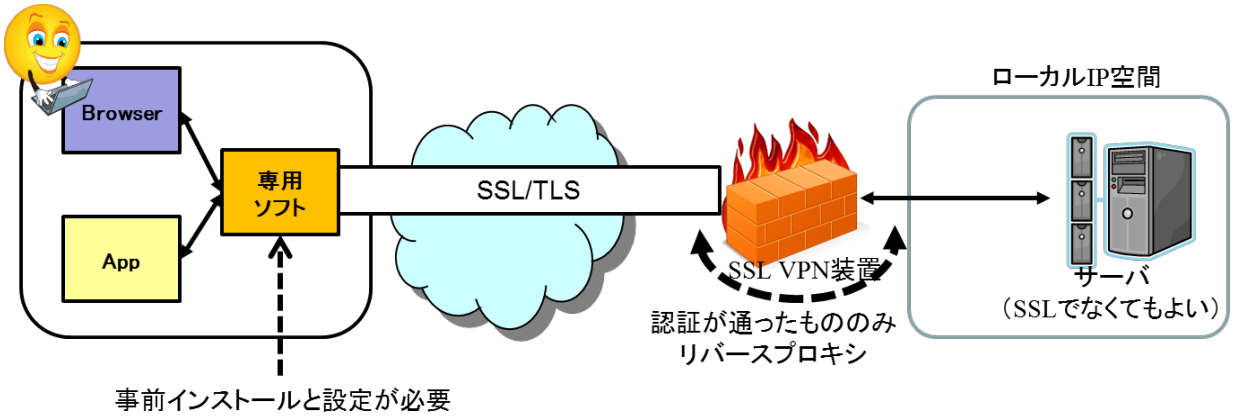
【クライアントレス型（ブラウザベース）】



【On-demand インストール型（Java や Active X を使ってブラウザ以外でも利用可能）】



【クライアント型（専用ソフトベース）】



Appendix：

付録

Appendix A：チェックリスト

チェックリストの原本は以下の URL からも入手可能である。

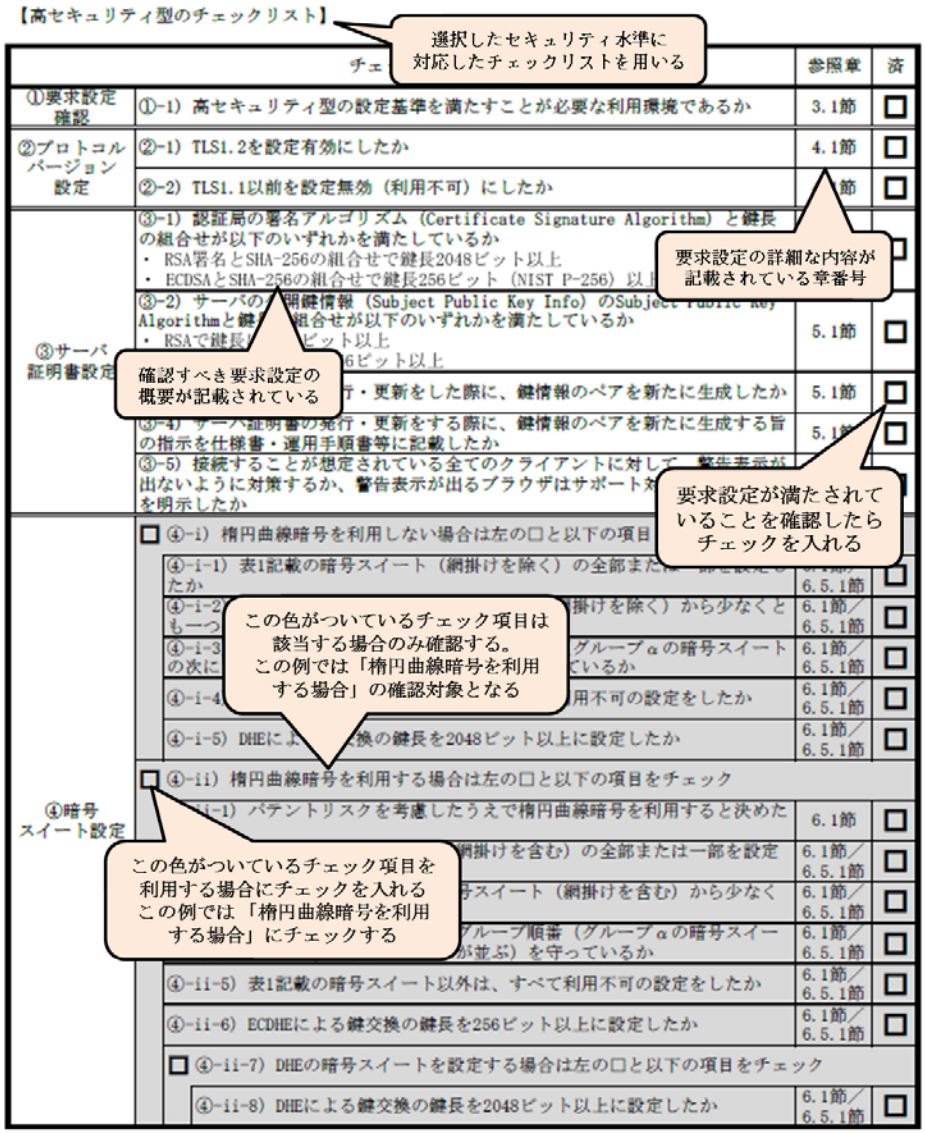
[pdf 版]　http://www.ipa.go.jp/files/000045652.pdf

[excel 版] http://www.ipa.go.jp/files/000045650.xlsx

A.1.　チェックリストの利用方法

本チェックリストは、以下の項目について、選択した設定基準に対応した要求設定をもれなく実施したことを確認するためのチェックリストである。選択した設定基準に応じたチェックリストを用い、すべてのチェック項目について、該当章に記載の要求設定に合致していることを確認して「済」にチェックが入ることが求められる。

＜チェックリストの例＞



A.2. 高セキュリティ型のチェックリスト

【高セキュリティ型のチェックリスト】

チェック項目		参照章	済
①要求設定確認	①-1）高セキュリティ型の設定基準を満たすことが必要な利用環境であるか	3.1節	☐
②プロトコルバージョン設定	②-1）TLS1.2を設定有効にしたか	4.1節	☐
	②-2）TLS1.1以前を設定無効（利用不可）にしたか	4.1節	☐
③サーバ証明書設定	③-1）認証局の署名アルゴリズム（Certificate Signature Algorithm）と鍵長の組合せが以下のいずれかを満たしているか ・ RSA署名とSHA-256の組合せで鍵長2048ビット以上 ・ ECDSAとSHA-256の組合せで鍵長256ビット（NIST P-256）以上	5.1節	☐
	③-2）サーバの公開鍵情報（Subject Public Key Info）のSubject Public Key Algorithmと鍵長の組合せが以下のいずれかを満たしているか ・ RSAで鍵長は2048ビット以上 ・ 楕円曲線暗号で鍵長256ビット以上	5.1節	☐
	③-3）サーバ証明書の発行・更新をした際に、鍵情報のペアを新たに生成したか	5.1節	☐
	③-4）サーバ証明書の発行・更新をする際に、鍵情報のペアを新たに生成する旨の指示を仕様書・運用手順書等に記載したか	5.1節	☐
	③-5）接続することが想定されている全てのクライアントに対して、警告表示が出ないように対策するか、警告表示が出るブラウザはサポート対象外であることを明示したか	5.1節	☐
④暗号スイート設定	☐ ④-i）楕円曲線暗号を利用しない場合は左の□と以下の項目をチェック		
	④-i-1）表1記載の暗号スイート（網掛けを除く）の全部または一部を設定したか	6.1節／6.5.1節	☐
	④-i-2）表1記載のグループαの暗号スイート（網掛けを除く）から少なくとも一つは設定したか	6.1節／6.5.1節	☐
	④-i-3）表1記載の暗号スイートのグループ順番（グループαの暗号スイートの次にグループβの暗号スイートが並ぶ）を守っているか	6.1節／6.5.1節	☐
	④-i-4）表1記載の暗号スイート以外は、すべて利用不可の設定をしたか	6.1節／6.5.1節	☐
	④-i-5）DHEによる鍵交換の鍵長を2048ビット以上に設定したか	6.1節／6.5.1節	☐
	☐ ④-ii）楕円曲線暗号を利用する場合は左の□と以下の項目をチェック		
	④-ii-1）パテントリスクを考慮したうえで楕円曲線暗号を利用すると決めたか	6.1節	☐
	④-ii-2）表1記載の暗号スイート（網掛けを含む）の全部または一部を設定したか	6.1節／6.5.1節	☐
	④-ii-3）表1記載のグループαの暗号スイート（網掛けを含む）から少なくとも一つは設定したか	6.1節／6.5.1節	☐
	④-ii-4）表1記載の暗号スイートのグループ順番（グループαの暗号スイートの次にグループβの暗号スイートが並ぶ）を守っているか	6.1節／6.5.1節	☐
	④-ii-5）表1記載の暗号スイート以外は、すべて利用不可の設定をしたか	6.1節／6.5.1節	☐
	④-ii-6）ECDHEによる鍵交換の鍵長を256ビット以上に設定したか	6.1節／6.5.1節	☐
	☐ ④-ii-7）DHEの暗号スイートを設定する場合は左の□と以下の項目をチェック		
	④-ii-8）DHEによる鍵交換の鍵長を2048ビット以上に設定したか	6.1節／6.5.1節	☐

【表1】

優先順位グループ	暗号スイート名	スイート番号
グループα	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	(0x00,0x9F)
	TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x7D)
	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0,0x2C)
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0,0x30)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x87)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x8B)
グループβ	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	(0x00,0x9E)
	TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x7C)
	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2B)
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2F)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x86)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x8A)

A.3. 推奨セキュリティ型のチェックリスト

【推奨セキュリティ型のチェックリスト（1/2）】

	チェック項目	参照章	済
①要求設定確認	チェック項目なし		
②プロトコルバージョン設定	②-1）TLS1.0を設定有効にしたか	4.1節	□
	②-2）SSL2.0及びSSL3.0を設定無効（利用不可）にしたか	4.1節	□
	□ ②-3）TLS1.2が実装されている場合には左の□と以下の項目をチェック		
	②-4）TLS1.2について設定を有効にしたか	4.1節	□
	□ ②-5）TLS1.1が実装されている場合には左の□と以下の項目をチェック		
	②-6）TLS1.1について設定を有効にしたか	4.1節	□
	□ ②-7）プロトコルバージョン優先順位を設定できる場合には左の□と以下の項目をチェック		
	②-8）設定有効になっているプロトコルバージョンのうち、もっとも新しいバージョンによる接続を最優先にしたか。接続できない場合に、順番に一つずつ前のプロトコルバージョンでの接続するようにしたか	4.1節	□
③サーバ証明書設定	③-1）認証局の署名アルゴリズム（Certificate Signature Algorithm）と鍵長の組合せが以下のいずれかを満たしているか ・ RSA署名とSHA-256の組合せで鍵長2048ビット以上 ・ ECDSAとSHA-256の組合せで鍵長256ビット（NIST P-256）以上	5.1節	□
	③-2）サーバの公開鍵情報（Subject Public Key Info）のSubject Public Key Algorithmと鍵長の組合せが以下のいずれかを満たしているか ・ RSAで鍵長は2048ビット以上 ・ 楕円曲線暗号で鍵長256ビット以上	5.1節	□
	③-3）サーバ証明書の発行・更新をした際に、鍵情報のペアを新たに生成したか	5.1節	□
	③-4）サーバ証明書の発行・更新をする際に、鍵情報のペアを新たに生成する旨の指示を仕様書・運用手順書等に記載したか	5.1節	□
	③-5）接続することが想定されている全てのクライアントに対して、警告表示が出ないように対策するか、警告表示が出るブラウザはサポート対象外であることを明示したか	5.1節	□

（続く）

【推奨セキュリティ型のチェックリスト（2/2）】

チェック項目		参照章	済
④暗号スイート設定	□ ④-i）楕円曲線暗号を利用しない場合は左の□と以下の項目をチェック		
	④-i-1）表2記載の暗号スイート（網掛けを除く）の全部または一部を設定したか	6.1節／6.5.2節	□
	④-i-2）表2記載のグループA及びグループBの暗号スイート（網掛けを除く）から少なくとも一つは設定したか	6.1節／6.5.2節	□
	④-i-3）表2記載の暗号スイートのグループ順番（グループAの暗号スイートの次にグループBの暗号スイートが並ぶ、以下同様）を守っているか	6.1節／6.5.2節	□
	④-i-4）表2記載の暗号スイート以外は、すべて利用不可の設定をしたか	6.1節／6.5.2節	□
	④-i-5）RSAによる鍵交換の鍵長を2048ビット以上に設定したか	6.1節／6.5.2節	□
	□ ④-i-6）DHEを利用する暗号スイートを設定する場合は左の□と以下の項目をチェック		
	④-i-7）DHEによる鍵交換の鍵長を1024ビット以上に設定したか	6.1節／6.5.2節	□
	□ ④-i-8）金融サービスや電子商取引サービスなど、不特定多数に公開されるサービス等において使用されるサーバである場合には左の□と以下の項目をチェック		
	④-i-9）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	□
	□ ④-ii）楕円曲線暗号を利用する場合は左の□と以下の項目をチェック		
	④-ii-1）パテントリスクを考慮したうえで楕円曲線暗号を利用すると決めたか	6.1節	□
	④-ii-2）表2記載の暗号スイート（網掛けを含む）の全部または一部を設定したか	6.1節／6.5.2節	□
	④-ii-3）表2記載のグループA及びグループBの暗号スイート（網掛けを含む）から少なくとも一つは設定したか	6.1節／6.5.2節	□
	④-ii-4）表2記載の暗号スイートのグループ順番（グループAの暗号スイートの次にグループBの暗号スイートが並ぶ、以下同様）を守っているか	6.1節／6.5.2節	□
	④-ii-5）表2記載の暗号スイート以外は、すべて利用不可の設定をしたか	6.1節／6.5.2節	□
	④-ii-6）ECDHE/ECDHによる鍵交換の鍵長を256ビット以上に設定したか	6.1節／6.5.2節	□
	④-ii-7）RSAによる鍵交換の鍵長を2048ビット以上に設定したか	6.1節／6.5.2節	□
	□ ④-ii-8）DHEを利用する暗号スイートを設定する場合は左の□と以下の項目をチェック		
	④-ii-9）DHEによる鍵交換の鍵長を1024ビット以上に設定したか	6.1節／6.5.2節	□
	□ ④-ii-10）金融サービスや電子商取引サービスなど、不特定多数に公開されるサービス等において使用されるサーバである場合には左の□と以下の項目をチェック		
	④-ii-11）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	□

【表2】

優先順位グループ	暗号スイート名	スイート番号
グループA	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	(0x00,0x9E)
	TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x7C)
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	(0x00,0x67)
	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0x00,0xBE)
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	(0x00,0x33)
	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA	(0x00,0x45)
	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2B)
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2F)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x86)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x8A)
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	(0xC0,0x23)
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	(0xC0,0x27)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x72)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x76)
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	(0xC0,0x09)
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	(0xC0,0x13)
グループB	TLS_RSA_WITH_AES_128_GCM_SHA256	(0x00,0x9C)
	TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x7A)
	TLS_RSA_WITH_AES_128_CBC_SHA256	(0x00,0x3C)
	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0x00,0xBA)
	TLS_RSA_WITH_AES_128_CBC_SHA	(0x00,0x2F)
	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	(0x00,0x41)
グループC	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2D)
	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256	(0xC0,0x31)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x88)
	TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x8C)
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	(0xC0,0x25)
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	(0xC0,0x29)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x74)
	TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x78)
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	(0xC0,0x04)
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	(0xC0,0x0E)
グループD	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	(0x00,0x9F)
	TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x7D)
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	(0x00,0x6B)
	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256	(0x00,0xC4)
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	(0x00,0x39)
	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA	(0x00,0x88)
	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0,0x2C)
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0,0x30)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x87)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x8B)
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	(0xC0,0x24)
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	(0xC0,0x28)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x73)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x77)
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	(0xC0,0x0A)
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	(0xC0,0x14)
グループE	TLS_RSA_WITH_AES_256_GCM_SHA384	(0x00,0x9D)
	TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x7B)
	TLS_RSA_WITH_AES_256_CBC_SHA256	(0x00,0x3D)
	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256	(0x00,0xC0)
	TLS_RSA_WITH_AES_256_CBC_SHA	(0x00,0x35)
	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	(0x00,0x84)
グループF	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0,0x2E)
	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384	(0xC0,0x32)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x89)
	TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x8D)
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	(0xC0,0x26)
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	(0xC0,0x2A)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x75)
	TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x79)
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	(0xC0,0x05)
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	(0xC0,0x0F)

A.4. セキュリティ例外型のチェックリスト

【セキュリティ例外型のチェックリスト（1/2）】

チェック項目		参照章	済
①要求設定確認	①-1）セキュリティ例外型の設定基準を満たすことが必要な利用環境であるか	3.1節	□
	①-2）推奨セキュリティ型への早期移行を前提とし、移行計画を策定する、利用終了期限を定める、利用者への注意喚起を行うなど、今後の対処方針を具体的に策定しているか	3.1節	□
②プロトコルバージョン設定	②-1）TLS1.0及びSSL3.0を設定有効にしたか	4.1節	□
	②-2）SSL2.0を設定無効（利用不可）にしたか	4.1節	□
	□ ②-3）TLS1.2が実装されている場合には左の□と以下の項目をチェック		
	②-4）TLS1.2について設定を有効にしたか	4.1節	□
	□ ②-5）TLS1.1が実装されている場合には左の□と以下の項目をチェック		
	②-6）TLS1.1について設定を有効にしたか	4.1節	□
	□ ②-7）プロトコルバージョン優先順位を設定できる場合には左の□と以下の項目をチェック		
	②-8）設定有効になっているプロトコルバージョンのうち、もっとも新しいバージョンによる接続を最優先にしたか。接続できない場合に、順番に一つずつ前のプロトコルバージョンでの接続するようにしたか	4.1節	□
③サーバ証明書設定	③-1）認証局の署名アルゴリズム（Certificate Signature Algorithm）と鍵長の組合せが以下のいずれかを満たしているか ・ RSA署名とSHA-256の組合せで鍵長2048ビット以上 ・ RSA署名とSHA-1の組合せで鍵長2048ビット以上	5.1節	□
	③-2）サーバの公開鍵情報（Subject Public Key Info）のSubject Public Key Algorithmと鍵長の組合せが以下を満たしているか ・ RSAで鍵長は2048ビット以上	5.1節	□
	③-3）サーバ証明書の発行・更新をした際に、鍵情報のペアを新たに生成したか	5.1節	□
	③-4）サーバ証明書の発行・更新をする際に、鍵情報のペアを新たに生成する旨の指示を仕様書・運用手順書等に記載したか	5.1節	□
	③-5）接続することが想定されている全てのクライアントに対して、警告表示が出ないように対策するか、警告表示が出るブラウザはサポート対象外であることを明示したか	5.1節	□

（続く）

【セキュリティ例外型のチェックリスト（2/2）】

	チェック項目	参照章	済
④暗号スイート設定	□ ④-i）楕円曲線暗号を利用しない場合は左の□と以下の項目をチェック		
	④-i-1）表3記載の暗号スイート（網掛けを除く）の全部または一部を設定したか	6.1節／6.5.3節	□
	④-i-2）表3記載のグループA及びグループBの暗号スイート（網掛けを除く）から少なくとも一つは設定したか	6.1節／6.5.3節	□
	④-i-3）表3記載のグループGの暗号スイートを設定したか	6.1節／6.5.3節	□
	④-i-4）表3記載の暗号スイートのグループ順番（グループAの暗号スイートの次にグループBの暗号スイートが並ぶ、以下同様）を守っているか	6.1節／6.5.3節	□
	④-i-5）表3記載の暗号スイート以外は、すべて利用不可の設定をしたか	6.1節／6.5.3節	□
	④-i-6）RSAによる鍵交換の鍵長を2048ビット以上に設定したか	6.1節／6.5.3節	□
	□ ④-i-7）DHEを利用する暗号スイートを設定する場合は左の□と以下の項目をチェック		
	④-i-8）DHEによる鍵交換の鍵長を1024ビット以上に設定したか	6.1節／6.5.3節	□
	□ ④-i-9）金融サービスや電子商取引サービスなど、不特定多数に公開されるサービス等において使用されるサーバである場合には左の□と以下の項目をチェック		
	④-i-10）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.3節	□
	④-i-11）DHE-DSS-DES-CBC3-SHAとDES-CBC3-SHAの少なくとも一方は設定したか	6.1節／6.5.3節	□
	□ ④-ii）楕円曲線暗号を利用する場合は左の□と以下の項目をチェック		
	④-ii-1）パテントリスクを考慮したうえで楕円曲線暗号を利用すると決めたか	6.1節	□
	④-ii-2）表3記載の暗号スイート（網掛けを含む）の全部または一部を設定したか	6.1節／6.5.3節	□
	④-ii-3）表3記載のグループA及びグループBの暗号スイート（網掛けを含む）から少なくとも一つは設定したか	6.1節／6.5.3節	□
	④-ii-4）表3記載のグループGの暗号スイートを設定したか	6.1節／6.5.3節	□
	④-ii-5）表3記載の暗号スイートのグループ順番（グループAの暗号スイートの次にグループBの暗号スイートが並ぶ、以下同様）を守っているか	6.1節／6.5.3節	□
	④-ii-6）表3記載の暗号スイート以外は、すべて利用不可の設定をしたか	6.1節／6.5.3節	□
	④-ii-7）ECDHE/ECDHによる鍵交換の鍵長を256ビット以上に設定したか	6.1節／6.5.2節	□
	④-ii-8）RSAによる鍵交換の鍵長を2048ビット以上に設定したか	6.1節／6.5.3節	□
	□ ④-ii-9）DHEを利用する暗号スイートを設定する場合は左の□と以下の項目をチェック		
	④-ii-10）DHEによる鍵交換の鍵長を1024ビット以上に設定したか	6.1節／6.5.3節	□
	□ ④-ii-11）金融サービスや電子商取引サービスなど、不特定多数に公開されるサービス等において使用されるサーバである場合には左の□と以下の項目をチェック		
	④-ii-12）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.3節	□
	④-ii-13）DES-CBC3-SHAの暗号スイートを設定したか	6.1節／6.5.3節	□

【表3】

優先順位グループ	暗号スイート名	スイート番号
グループA	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	(0x00,0x9E)
	TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x7C)
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	(0x00,0x67)
	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0x00,0xBE)
	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	(0x00,0x33)
	TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA	(0x00,0x45)
	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2B)
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2F)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x86)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x8A)
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	(0xC0,0x23)
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	(0xC0,0x27)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x72)
	TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x76)
	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	(0xC0,0x09)
	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	(0xC0,0x13)
グループB	TLS_RSA_WITH_AES_128_GCM_SHA256	(0x00,0x9C)
	TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x7A)
	TLS_RSA_WITH_AES_128_CBC_SHA256	(0x00,0x3C)
	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0x00,0xBA)
	TLS_RSA_WITH_AES_128_CBC_SHA	(0x00,0x2F)
	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	(0x00,0x41)
グループC	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0,0x2D)
	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256	(0xC0,0x31)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x88)
	TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256	(0xC0,0x8C)
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	(0xC0,0x25)
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	(0xC0,0x29)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x74)
	TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256	(0xC0,0x78)
	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	(0xC0,0x04)
	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	(0xC0,0x0E)
グループD	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	(0x00,0x9F)
	TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x7D)
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	(0x00,0x6B)
	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256	(0x00,0xC4)
	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	(0x00,0x39)
	TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA	(0x00,0x88)
	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0,0x2C)
	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0,0x30)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x87)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x8B)
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	(0xC0,0x24)
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	(0xC0,0x28)
	TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x73)
	TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x77)
	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	(0xC0,0x0A)
	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	(0xC0,0x14)
グループE	TLS_RSA_WITH_AES_256_GCM_SHA384	(0x00,0x9D)
	TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x7B)
	TLS_RSA_WITH_AES_256_CBC_SHA256	(0x00,0x3D)
	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256	(0x00,0xC0)
	TLS_RSA_WITH_AES_256_CBC_SHA	(0x00,0x35)
	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	(0x00,0x84)
グループF	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0,0x2E)
	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384	(0xC0,0x32)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x89)
	TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384	(0xC0,0x8D)
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	(0xC0,0x26)
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	(0xC0,0x2A)
	TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x75)
	TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384	(0xC0,0x79)
	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	(0xC0,0x05)
	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	(0xC0,0x0F)

【表3（続）】

優先順位グループ	暗号スイート名	スイート番号
グループG	TLS_RSA_WITH_RC4_128_SHA	(0x00,0x05)
グループH	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	(0x00,0x16)
	TLS_RSA_WITH_3DES_EDE_CBC_SHA	(0x00,0x0A)

Appendix B：サーバ設定編

本 Appendix では、サーバ設定を行う上での参考情報として、設定方法例を記載する。

なお、利用するバージョンやディストリビューションの違いにより、設定方法が異なったり、設定ができなかったりする場合があることに留意すること。正式な取扱説明書やマニュアルを参照するとともに、一参考資料として利用されたい。

B.1. サーバ設定方法例のまとめ

B.1.1. Apache の場合

Apache HTTP Server の設定ファイル（デフォルトの場合、httpd-ssl.conf）での設定例を以下に示す。

```
<VirtualHost *:443>
        （中略）
        SSLEngine on

  証明書と鍵の設定[39]
        SSLCertificateFile /etc/ssl/chain.crt
        SSLCertificateKeyFile /etc/ssl/server.key

  暗号スイート設定。Appendix C.2 も参照のこと
        SSLCipherSuite "暗号スイート設定"

  プロトコルバージョン設定。Appendix B.2.1 も参照のこと
        SSLProtocol バージョン設定

  暗号スイート順序サーバ優先設定
        SSLHonorCipherOrder On

  HTTP Strict Transport Security、OCSP Stapling、Public Key Pinning の設定をする場合には、
  ここに追記する。7.2 節及び Appendix B.4 以降も参照のこと

</VirtualHost>
```

[39] 設定する内容は以下のとおり。
/etc/ssl/chain.crt：サーバ証明書および中間証明書、/etc/ssl/server.key：サーバ証明書に対応する秘密鍵

B.1.2. lighttpd の場合

lighttpd の設定ファイル（デフォルトの場合、modules.conf と lighttpd.conf）での設定例を以下に示す。

〔modules.conf での設定〕

```
server.modules = (
        （中略）
       "mod_setenv"
)
```

〔lighttpd.conf での設定〕

```
$SERVER["socket"] == "0.0.0.0:443" {
       ssl.engine = "enable"
        （中略）
```

証明書と鍵の設定

```
       ssl.pemfile = "/etc/ssl/serverkey_cert.pem"
       ssl.ca-file = "/etc/ssl/ca.crt"
```

暗号スイート設定。Appendix C.2 も参照のこと

```
       ssl.cipher-list = "暗号スイート設定"
```

プロトコルバージョン設定。Appendix B.2.2 も参照のこと

```
       ssl.use-プロトコルバージョン  = "利用可否"
```

暗号スイート順序サーバ優先設定

```
       ssl.honor-cipher-order = "enable"
```

HTTP Strict Transport Security、Public Key Pinning の設定をする場合には、ここに追記する。7.2 節及び Appendix B.4 以降を参照のこと。なお、lighttpd では OCSP Stapling の設定はできない

```
}
```

B.1.3. nginx の場合

nginx の設定ファイル（デフォルトの場合、nginx.conf）での設定例を以下に示す。

```
server {
       listen 443 ssl;
```

（中略）

証明書と鍵の設定

```
ssl_certificate /etc/ssl/chain.crt;
ssl_certificate_key /etc/ssl/server.key;
```

暗号スイート設定。Appendix C.2 も参照のこと

```
ssl_ciphers "暗号スイート設定";
```

プロトコルバージョン設定。Appendix B.2.3 も参照のこと

```
ssl_protocols プロトコルバージョン設定;
```

暗号スイート順序サーバ優先設定

```
ssl_prefer_server_ciphers on;
```

HTTP Strict Transport Security、OCSP Stapling、Public Key Pinning の設定をする場合には、ここに追記する。7.2 節及び Appendix B.4 以降を参照のこと

}

B.2. プロトコルバージョンの設定方法例

B.2.1. Apache の場合

Apache での設定例を以下に示す。

- 高セキュリティ型
  SSLProtocol TLSv1.2

- 推奨セキュリティ型
  SSLProtocol All -SSLv2 -SSLv3

- セキュリティ例外型
  SSLProtocol All -SSLv2

B.2.2. lighttpd の場合

lighttpd での設定例を以下に示す。

- 高セキュリティ型
  ssl.use-tlsv1.1 = "disable"
  ssl.use-tlsv1 = "disable"
  ssl.use-sslv3 = "disable"
  ssl.use-sslv2 = "disable"

- 推奨セキュリティ型
  ssl.use-sslv3 = "disable"
  ssl.use-sslv2 = "disable"

- セキュリティ例外型
  ssl.use-sslv2 = "disable"

B.2.3. nginx の場合

nginx での設定例を以下に示す。なお、TLS1.1 及び TLS1.2 は、バージョンが 1.1.13 または 1.0.12 であり、かつ OpenSSL のバージョンが 1.0.1 以上の時に利用できる。

- 高セキュリティ型（Ver. 1.1.13/1.0.12 かつ OpenSSL ver. 1.0.1 以上）
  ssl_protocols TLSv1.2;

- 推奨セキュリティ型
  ssl_protocols TLSv1.2 TLSv1.1 TLSv1;（Ver. 1.1.13/1.0.12 かつ OpenSSL ver. 1.0.1 以上）
  ssl_protocols TLSv1;

- セキュリティ例外型
  ssl_protocols TLSv1.2 TLSv1.1 TLSv1 SSLv3;（Ver. 1.1.13/1.0.12 かつ OpenSSL ver. 1.0.1 以上）
  ssl_protocols TLSv1 SSLv3;

B.2.4. Microsoft IIS の場合

各 OS におけるプロトコルバージョンのサポート状況は以下の通りである。

	TLS1.2	TLS1.1	TLS1.0	SSL3.0	SSL2.0
Windows Server 2008	×	×	○	○	○
Windows Vista	×	×	○	○	○
Windows Server 2008 R2（以降）	○	○	○	○	○
Windows 7 以降の Windows	○	○	○	○	○

凡例：○ サポートあり　　× サポートなし

サポートされているプロトコルバージョンの利用可否については、以下の設定例に従い、レジストリを設定する。

参考情報：
特定の暗号化アルゴリズムおよび Schannel.dll のプロトコルの使用を制限する方法
https://support.microsoft.com/en-us/kb/245030

- 高セキュリティ型
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥SSL 2.0¥Server
  "DisabledByDefault"=dword:00000001
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥SSL 3.0¥Server
  "DisabledByDefault"=dword:00000001
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥TLS 1.0¥Server
  "DisabledByDefault"=dword:00000001
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥TLS 1.1¥Server
  "DisabledByDefault"=dword:00000001

- 推奨セキュリティ型
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥SSL 2.0¥Server
  "DisabledByDefault"=dword:00000001
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥SSL 3.0¥Server
  "DisabledByDefault"=dword:00000001

- セキュリティ例外型
  HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurityProviders¥Schannel¥Protocols¥SSL 2.0¥Server
  "DisabledByDefault"=dword:00000001

B.3. 鍵パラメータファイルの設定方法例

B.3.1. OpenSSL による DHE、ECDH、ECDHE 鍵パラメータファイルの生成

OpenSSL コマンドにより、DHE 鍵パラメータファイル（2048 ビット）を生成するには以下を

実行する。

```
openssl dhparam -out dh2048.pem -outform PEM 2048
```

また、ECDH、ECDHE 鍵パラメータファイル（256 ビット）は以下のようにして生成することができる。

```
openssl ecparam -out prime256v1.pem -name prime256v1
```

B.3.2. Apache における DHE、ECDH、ECDHE 鍵パラメータ設定

SSLCertificateFile は設定ファイル中でいくつも指定できるプロパティであり、通常は PEM 形式の SSL サーバ証明書を指定するためのものである。

Apache 2.4.7 以降では、SSLCertificateFile で設定するファイルの中に、DHE、ECDH、ECDHE の鍵長を示すパラメータファイルを明示的に含めることができる。そのために、Appendix B.1.1 の証明書と鍵の設定の部分で指定するファイル（Appendix B.1.1 の場合、/etc/ssl/chain.crt）に対して、Appendix B.3.1 で生成した鍵パラメータファイルを追記する。

例えば、linux 等であれば以下の処理を行う。

- DHE 鍵パラメータファイル（2048 ビット）の指定例
  ```
  cat dh2048.pem >> /etc/ssl/chain.crt
  ```

- ECDH、ECDHE 鍵パラメータファイル（256 ビット）の指定例
  ```
  cat prime256v1.pem >> /etc/ssl/chain.crt
  ```

B.3.3. lighttpd における DHE、ECDH、ECDHE 鍵パラメータ設定

lighttpd では、Appendix B.3.1 で生成した鍵パラメータファイルについて、Appendix B.1.2 の証明書と鍵の設定の部分に、以下のように追加する。

- DHE の鍵パラメータファイル（2048 ビット）の指定例
  ```
  ssl.dh-file = "/etc/ssl/dh2048.pem"
  ```

- ECDH、ECDHE の楕円曲線パラメータ（256 ビット）の指定例
  ```
  ssl.ec-curve = "prime256v1"
  ```

B.3.4. nginx における DHE、ECDH、ECDHE 鍵パラメータ設定

nginx では、Appendix B.3.1 で生成した鍵パラメータファイルについて、Appendix B.1.3 の証明書と鍵の設定の部分に、以下のように追加する。

- DHE の鍵パラメータファイル（2048 ビット）の指定例
  ssl_dhparam /etc/ssl/dh2048.pem;

- ECDH、ECDHE の楕円曲線パラメータ（256 ビット）の指定例
  ssl_ecdh_curve prime256v1;

B.4. HTTP Strict Transport Security（HSTS）の設定方法例

B.4.1. Apache の場合

HTTP ヘッダに HSTS の情報を追加するために、設定ファイルに以下の記述を追加する。なお、max-age は有効期間を表し、この例では 365 日（31,536,000 秒）の有効期間を設定することを意味している。また、includeSubDomains がある場合、サブドメインにも適用される。

```
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

なお、HTTP の場合に強制的に HTTPS にリダイレクトするためには、<VirtualHost *:80>中の RewriteRule、RewriteEngine の設定を以下のように追記する。

```
<VirtualHost *:80>
        （中略）
        ServerAlias *
        RewriteEngine On
        RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>
```

B.4.2. lighttpd の場合

HTTP ヘッダに HSTS の情報を追加するために、設定ファイル（Appendix B.1.2 の場合、lighttpd.conf）に以下の記述を追加する。なお、max-age は有効期間を表し、この例では 365 日（31,536,000 秒）の有効期間を設定することを意味している。また、includeSubDomains がある場合、サブドメインにも適用される。

```
setenv.add-response-header = (
      "Strict-Transport-Security" => "max-age=31536000; includeSubDomains"
)
```

なお、HTTP の場合に強制的に HTTPS にリダイレクトするためには、設定ファイル（Appendix B.1.2 の場合、modules.conf と lighttpd.conf）に以下のように追記する。

〔modules.conf での設定〕

```
server.modules = (
        （中略）
       "mod_redirect"
)
```

〔lighttpd.conf での設定〕

```
$HTTP["scheme"] == "http" {
        （中略）
       $HTTP["host"] =~ ".*" {
            url.redirect = (".*" => "https://%0$0")
       }
}
```

B.4.3. nginx の場合

HTTP ヘッダに HSTS の情報を追加するために、設定ファイルに以下の記述を追加する。なお、max-age は有効期間を表し、この例では 365 日（31,536,000 秒）の有効期間を設定することを意味している。また、includeSubDomains がある場合、サブドメインにも適用される。

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains";
```

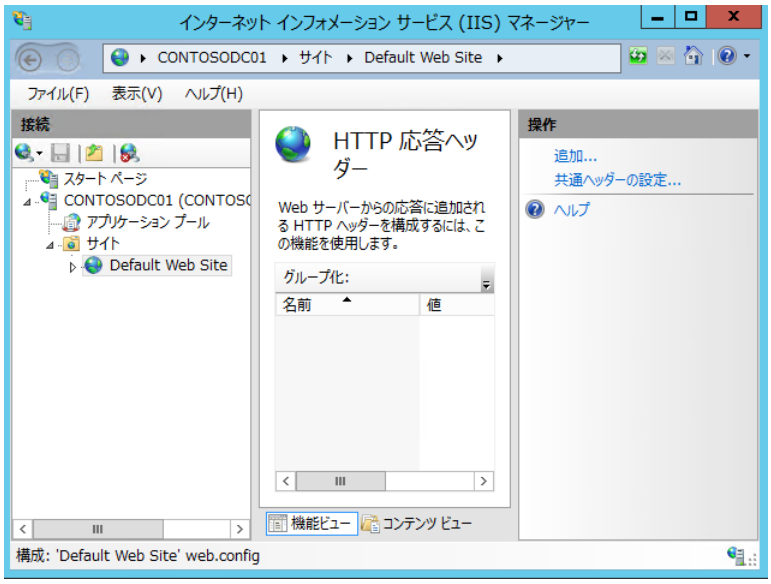
なお、HTTP の場合に強制的に HTTPS にリダイレクトするためには、"listen 80;"中に、以下のように追記する。

```
server {
        listen 80;
        (中略)
        return 301 https://$hostname$request_uri;
}
```

B.4.4. Microsoft IIS の場合

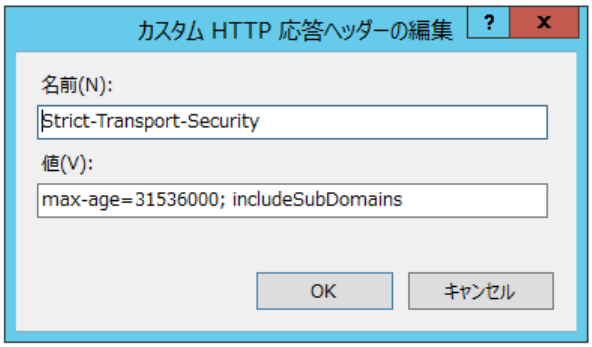
IIS では、HTTP ヘッダに HSTS の情報を追加するために、以下の手順により設定する。

1） 「IIS マネージャー」を開く
2） 「機能ビュー」を開く
3） 「HTTP 応答ヘッダ」をダブルクリックする
4） 「操作」のペインで「追加」をクリックする



5） 「名前」「値」の箇所を以下のように設定する。なお、max-age は有効期間を表し、この例では 365 日（31,536,000 秒）の有効期間を設定することを意味している。また、includeSubDomains がある場合、サブドメインにも適用される

名前：Strict-Transport-Security
値　：max-age=31536000; includeSubDomains



6） 「OK」をクリックする。

B.5. OCSP Stapling の設定方法例

B.5.1. Apache の場合

OCSP stapling を有効にするために、設定ファイルに以下の記述を追加する。

なお、SSLStaplingCache の stapling_cache はキャッシュサイズを表し、この例では 128,000 バイトを設定することを意味している。また、<VirtualHost *:443>の前に記載すること。

```
SSLStaplingCache shmcb:/tmp/stapling_cache(128000)
```

```
<VirtualHost *:443>
        （中略）
        SSLCACertificateFile /etc/ssl/ca-certs.pem
        SSLUseStapling on
</VirtualHost>
```

B.5.2. nginx の場合

OCSP stapling を有効にするために、設定ファイルに以下の記述を追加する。

```
server {
        （中略）
        ssl_stapling on;
        ssl_stapling_verify on;
        ssl_trusted_certificate /etc/ssl/ca-certs.pem;
}
```

B.5.3. Microsoft IIS の場合

Windows Server 2008 以降の Windows では、デフォルトで OCSP Stapling が設定されている。

B.6. Public Key Pinning の設定方法例

Public Key Pinning で使用される HTTP ヘッダの属性名は"Public-Key-Pins"であり、ヘッダの例は以下のようになる。

Public-Key-Pins 'pin-sha256="サーバ証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値"; pin-sha256="中間証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値"; max-age=有効期間; includeSubDomains'
Public-Key-Pins 'pin-sha1="サーバ証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値"; pin-sha1="中間証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値"; max-age=有効期間; includeSubDomains'

- エンドエンティティ（SSL サーバ証明書）から、最上位の中間証明書まで全てを列挙する
- 公開鍵情報のハッシュ値を計算するハッシュ関数は複数指定することができ、それぞれヘッダを追加すればよい。上記の例では、SHA-256 と SHA-1 の両方を指定することになる。
- max-age により有効期間（秒）を指定する。
- includeSubDomains がある場合、サブドメインにも適用される

〔具体的な表記例〕

Public-Key-Pins 'pin-sha256="QtXc8+scL7K6HiPksQ8mqIyY08Xdc4Z5raHT+xSh9/s="; pin-sha256="kb6xLprt35abNnSn74my4Dkfya9arbk5zN5a60YzuqE="; max-age=3000; includeSubDomains'

Public-Key-Pins 'pin-sha1="FhxvMPhD7Q+byiiwygLO0mL7L70="; pin-sha1="KqqJgAYLy9ogXOWETcR36ioKf20="; max-age=3000; includeSubDomains'

この例では、あるサーバ証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値が"QtXc8+"から始まる値であり、中間証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値が"kb6xLp"から始まる値であることを意味する。同様に、"FhxvM"、"KqqJgA"から始まる値はそれぞれの SHA-1 ハッシュ値の Base64 値である。

また、この場合の max-age は 50 分（3,000 秒）の有効期間を意味している。

なお、ハッシュ値の Base64 値を簡単に計算する方法はいくつかある。

例えば、OpenSSL を利用する方法や、PEM 形式のサーバ証明書を入力して Public-Key-Pins ヘッダを自動作成するサイト [40]がある。

〔OpenSSL を利用する方法〕

PEM 形式のあるサーバ証明書（certificate.pem）の SHA-256 ハッシュ値の Base64 値を求める場合は、以下のように計算する

```
openssl x509 -noout -in certificate.pem -pubkey | openssl asn1parse -noout -inform pem -out public.key;
openssl dgst -sha256 -binary public.key | openssl enc -base64
```

B.6.1. Apache の場合

B.6 の表記に従い、mod_headers モジュールを有効にし、以下の設定を追加する。この例では SHA-256 と SHA-1 の両方のヘッダを指定することを意味している。

```
Header add Public-Key-Pins 'pin-sha256="サーバ証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値"; pin-sha256="中間証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値"; max-age=有効期間'
Header add Public-Key-Pins 'pin-sha1="サーバ証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値"; pin-sha1="中間証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値"; max-age=有効期間'
```

ちなみに、mod_headers モジュールを有効にするためには、httpd.conf において

[40] https://projects.dm.id.lv/s/pkp-online/calculator.html

LoadModule headers_module modules/mod_headers.so

を設定する。

B.6.2. lighttpd での設定例 [41]

B.6 の表記に従い、設定ファイルにおいて、以下の設定を追加する。この例では SHA-256 と SHA-1 の両方のヘッダを指定することを意味している。

setenv.add-response-header = (
"Public-Key-Pins" => "pin-sha256=¥"サーバ証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値¥"; pin-sha256=¥"中間証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値¥"; max-age=有効期間",
"Public-Key-Pins" => "pin-sha1=¥"サーバ証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値¥"; pin-sha1=¥"中間証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値¥"; max-age=有効期間"
)

B.6.3. nginx の場合

B.6 の表記に従い、設定ファイルにおいて、以下の設定を追加する。この例では SHA-256 と SHA-1 の両方のヘッダを指定することを意味している。

add_header Public-Key-Pins 'pin-sha256="サーバ証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値"; pin-sha256="中間証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値"; max-age=有効期間';
add_header Public-Key-Pins 'pin-sha1="サーバ証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値"; pin-sha1="中間証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値"; max-age=有効期間';

B.6.4. Microsoft IIS の場合

IIS では、B.6 の表記に従い、以下の手順により設定する。

1) 「IIS マネージャー」を開く
2) 「機能ビュー」を開く
3) 「HTTP 応答ヘッダ」をダブルクリックする
4) 「操作」のペインで「追加」をクリックする

[41] HSTS と Public Key Pinning を同時に設定する場合には、一つの setenv.add-response-header 内に両方の設定を追加すること

5） B.6 の表記に従い、「名前」「値」の箇所に以下のように設定する。この例では SHA-256 と SHA-1 の両方のヘッダを指定することを意味している。

名前：Public-Key-Pinning

値　：pin-sha256="サーバ証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値", pin-sha256="中間証明書の公開鍵情報の SHA-256 ハッシュ値の Base64 値",pin-sha1="サーバ証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値",pin-sha1="中間証明書の公開鍵情報の SHA-1 ハッシュ値の Base64 値",max-age=有効期間

6） 「OK」をクリックする。

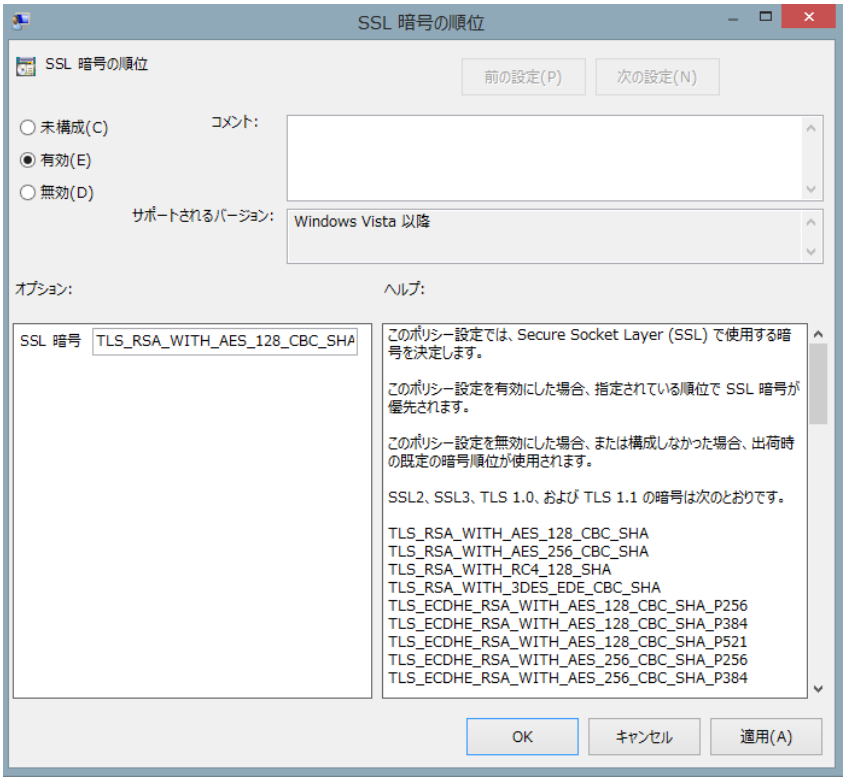
Appendix C：暗号スイートの設定例

本 Appendix では、暗号スイートの設定を行う上での参考情報として、設定方法例を記載する。

なお、利用するバージョンやディストリビューションの違いにより、実装されている暗号スイートの種類や設定方法が異なる場合があることに留意すること。正式な取扱説明書やマニュアルを参照するとともに、一参考資料として利用されたい。

C.1. Windows での設定例 [42]

1. コマンドプロンプトで gpedit.msc と入力し、Enter を押してグループポリシーオブジェクトエディタを起動する。
2. ［コンピューターの構成］＞［管理用テンプレート］＞［ネットワーク］＞［SSL 構成設定］の順に展開する。
3. ［SSL 構成設定］で［SSL 暗号（「SSL 暗号化スイート」と表記される場合もある）の順序］をダブルクリックする。
4. ［SSL 暗号の順序］ウィンドウで、［有効］をクリックする。
5. ウィンドウで、[SSL 暗号]フィールドの内容を、設定したい暗号リストの内容と置き換える。



[42] Windows Server 2008, 2008 R2, 2012, 2012 R2 については、GUI で暗号スイートやプロトコルバージョンを設定できるフリーウェアを NARTAC IIS Crypto が公開している
https://www.nartac.com/Products/IISCrypto/

なお、暗号リストは「,」で暗号スイートを連結して 1 行で記述し、空白や改行を含めない。優先順位は記述した順番で設定される。

- 高セキュリティ型の設定例（楕円曲線暗号あり）
  TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256

- 推奨セキュリティ型の設定例（楕円曲線暗号あり）
  TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384_P384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA

- セキュリティ例外型の設定例（楕円曲線暗号あり）
  TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256_P256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256_P256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256_P256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA_P256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA_P256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384_P384,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384_P384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA_P256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA_P256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_RC4_128_SHA,TLS_RSA_WITH_3DES_EDE_CBC_SHA

6. ［適用（A)］＞［OK］をクリックする。
7. グループポリシーオブジェクトエディタを閉じ、システムを再起動する。

C.2. OpenSSL 系での設定例

C.2.1. Apache, lighttpd, nginx の場合

Apache、lighttpd、nginx での暗号スイートの設定においては、C.2.2 の OpenSSL での暗号スイート設定例に従った設定を行う。

- Apache の場合の記述
  C.2.2 に従い、VirtualHost 中の SSLCipherSuite の設定を以下のように追記する。
  SSLCipherSuite "暗号スイート設定例"

- lighttpd の場合の記述
  C.2.2 に従い、$SERVER 中の ssl.cipher-list の設定を以下のように追記する。
  ssl.cipher-list = "暗号スイート設定例"

- nginx
  C.2.2 に従い、server 中の ssl_ciphers の設定を以下のように追記する。
  ssl_ciphers "暗号スイート設定例";

C.2.2. OpenSSL 系での暗号スイートの設定例

OpenSSL 系では、6.5 節に記載する暗号スイート名に対応する独自の表記を利用する（表 17 参照)。

〔SSLCipherSuite "暗号スイート設定例"の表記方法〕
例えば、高セキュリティ型の設定例（基本）なら
SSLCipherSuite "DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256"
と表記する。

なお、OpenSSL では暗号スイートの設定をパターンによる表記[43]で簡略化して記載することができる。ただし、パターンによる設定は、6.5 節に記載する詳細要求設定に従った設定を行うことが難しいため、本ガイドラインでは取り上げない。

- 高セキュリティ型の設定例（基本[44]）
  DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256

- 高セキュリティ型の設定例（楕円曲線暗号あり[45]）
  ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256

[43] 「ECDHE+AESGCM:DHE+CAMELLIA:DHE+AES:!DSS:!DH:!PSK:!SRP」のような表記をパターンによる表記という
[44] 「DHE+AESGCM:!DSS:!PSK:!SRP」での設定パターンによる暗号スイートを 6.5.1 節の優先順位に合わせたもの
[45] 「ECDHE+AESGCM:EDH+AESGCM:!DSS:!PSK:!SRP」での設定パターンによる暗号スイートを 6.5.1 節の優先順位に合わせたもの

- 推奨セキュリティ型の設定例（基本 [46]）
  DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-CAMELLIA128-SHA:DHE-RSA-AES128-SHA:AES128-GCM-SHA256:AES128-SHA256:CAMELLIA128-SHA:AES128-SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-CAMELLIA256-SHA:DHE-RSA-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:CAMELLIA256-SHA:AES256-SHA

- 推奨セキュリティ型の設定例（楕円曲線暗号あり [47]）
  ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-CAMELLIA128-SHA:DHE-RSA-AES128-SHA:AES128-GCM-SHA256:AES128-SHA256:CAMELLIA128-SHA:AES128-SHA:ECDH-ECDSA-AES128-GCM-SHA256:ECDH-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-CAMELLIA256-SHA:DHE-RSA-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:CAMELLIA256-SHA:AES256-SHA:ECDH-ECDSA-AES256-GCM-SHA384:ECDH-RSA-AES256-GCM-SHA384

- セキュリティ例外型の設定例（基本 [48]）
  DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-CAMELLIA128-SHA:DHE-RSA-AES128-SHA:AES128-GCM-SHA256:AES128-SHA256:CAMELLIA128-SHA:AES128-SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-CAMELLIA256-SHA:DHE-RSA-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:CAMELLIA256-SHA:AES256-SHA:RC4-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA

---

[46] 「DHE+AESGCM:RSA+AESGCM:DHE+CAMELLIA:DHE+AES:RSA+CAMELLIA:RSA+AES:!DSS:!PSK:!SRP」での設定パターンによる暗号スイートを 6.5.2 節の優先順位に合わせたもの

[47] 「ECDHE+AESGCM:DHE+AESGCM:RSA+AESGCM:DHE+CAMELLIA:DHE+AES:RSA+CAMELLIA:RSA+AES:ECDH+AESGCM:!DSS:!PSK:!SRP」での設定パターンによる暗号スイートを 6.5.2 節の優先順位に合わせたもの

[48] 「DHE+AESGCM:RSA+AESGCM:DHE+CAMELLIA:DHE+AES:RSA+CAMELLIA:RSA+AES:RC4-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA:!DSS:!PSK:!SRP」での設定パターンによる暗号スイートを 6.5.3 節の優先順位に合わせたもの

表 17　代表的な暗号スイートの対比表

6.5 節に記載する暗号スイート名	OpenSSL での暗号スイート名表記
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDHE-ECDSA-AES256-GCM-SHA384
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE-RSA-AES256-GCM-SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	ECDHE-ECDSA-AES128-GCM-SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDHE-RSA-AES128-GCM-SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	DHE-RSA-AES256-GCM-SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	DHE-RSA-AES128-GCM-SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	DHE-RSA-AES256-SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	DHE-RSA-AES128-SHA256
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA	DHE-RSA-CAMELLIA256-SHA
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA	DHE-RSA-CAMELLIA128-SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE-RSA-AES256-SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE-RSA-AES128-SHA
TLS_RSA_WITH_AES_256_GCM_SHA384	AES256-GCM-SHA384
TLS_RSA_WITH_AES_128_GCM_SHA256	AES128-GCM-SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256	AES256-SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256	AES128-SHA256
TLS_RSA_WITH_CAMELLIA_256_SHA	CAMELLIA256-SHA
TLS_RSA_WITH_CAMELLIA_128_SHA	CAMELLIA128-SHA
TLS_RSA_WITH_AES_256_CBC_SHA	AES256-SHA
TLS_RSA_WITH_AES_128_CBC_SHA	AES128-SHA
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384	ECDH-RSA-AES256-GCM-SHA384
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	ECDH-ECDSA-AES256-GCM-SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	ECDH-ECDSA-AES128-GCM-SHA256
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256	ECDH-RSA-AES128-GCM-SHA256
TLS_RSA_WITH_RC4_128_SHA	RC4-SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	EDH-RSA-DES-CBC3-SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	DES-CBC3-SHA

Appendix D：ルート CA 証明書の取り扱い

D.1. ルート CA 証明書の暗号アルゴリズムおよび鍵長の確認方法

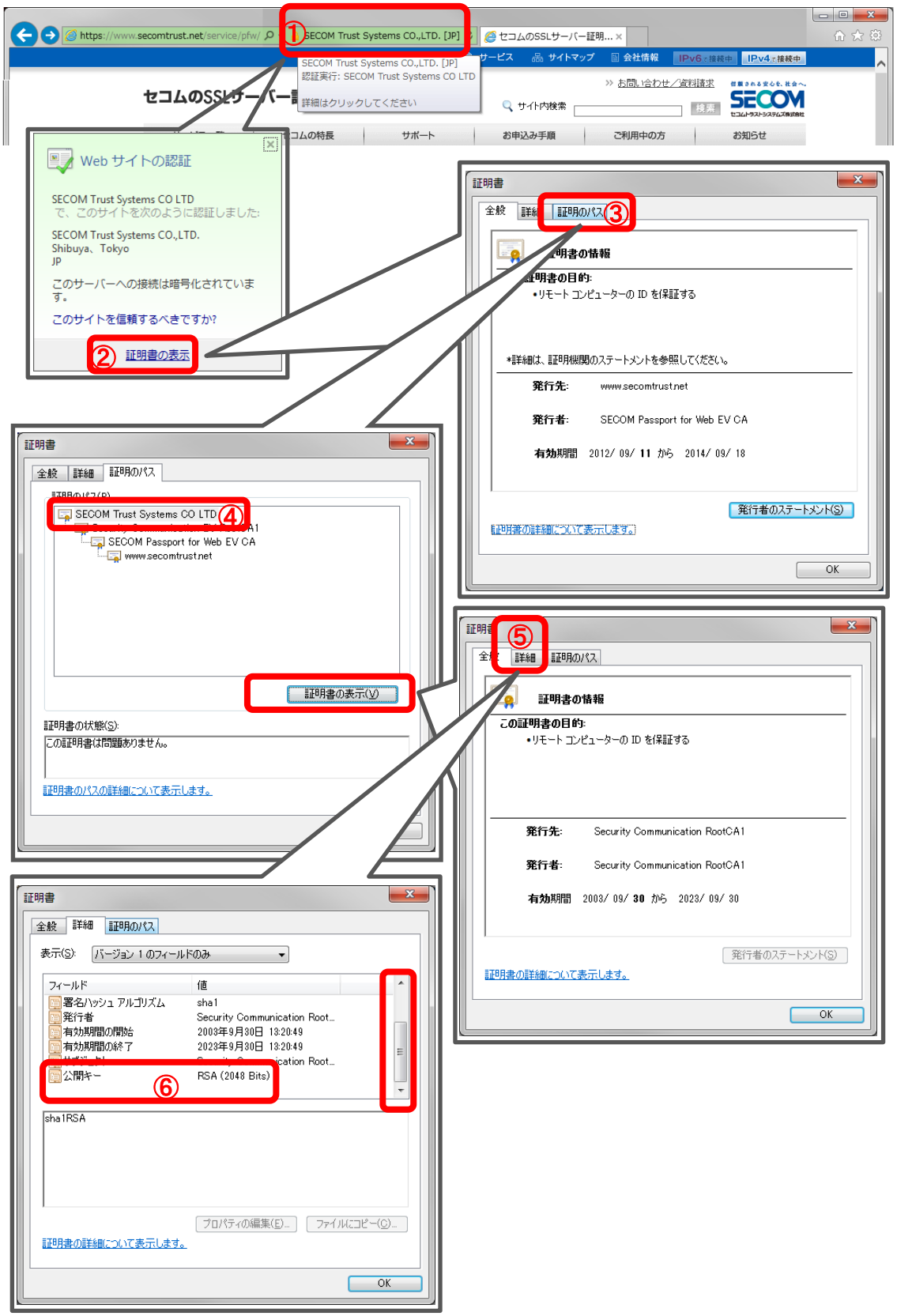
主要な認証事業者のルート CA 証明書の暗号アルゴリズムおよび鍵長を別表に掲載する。

ただし、事業者によってはサーバ証明書発行サービスを複数展開しているケースがあり、サービスによってルート CA が異なる場合があるので、どのサービスがどのルート CA の下で提供されているのかは、各事業者に確認する必要がある。

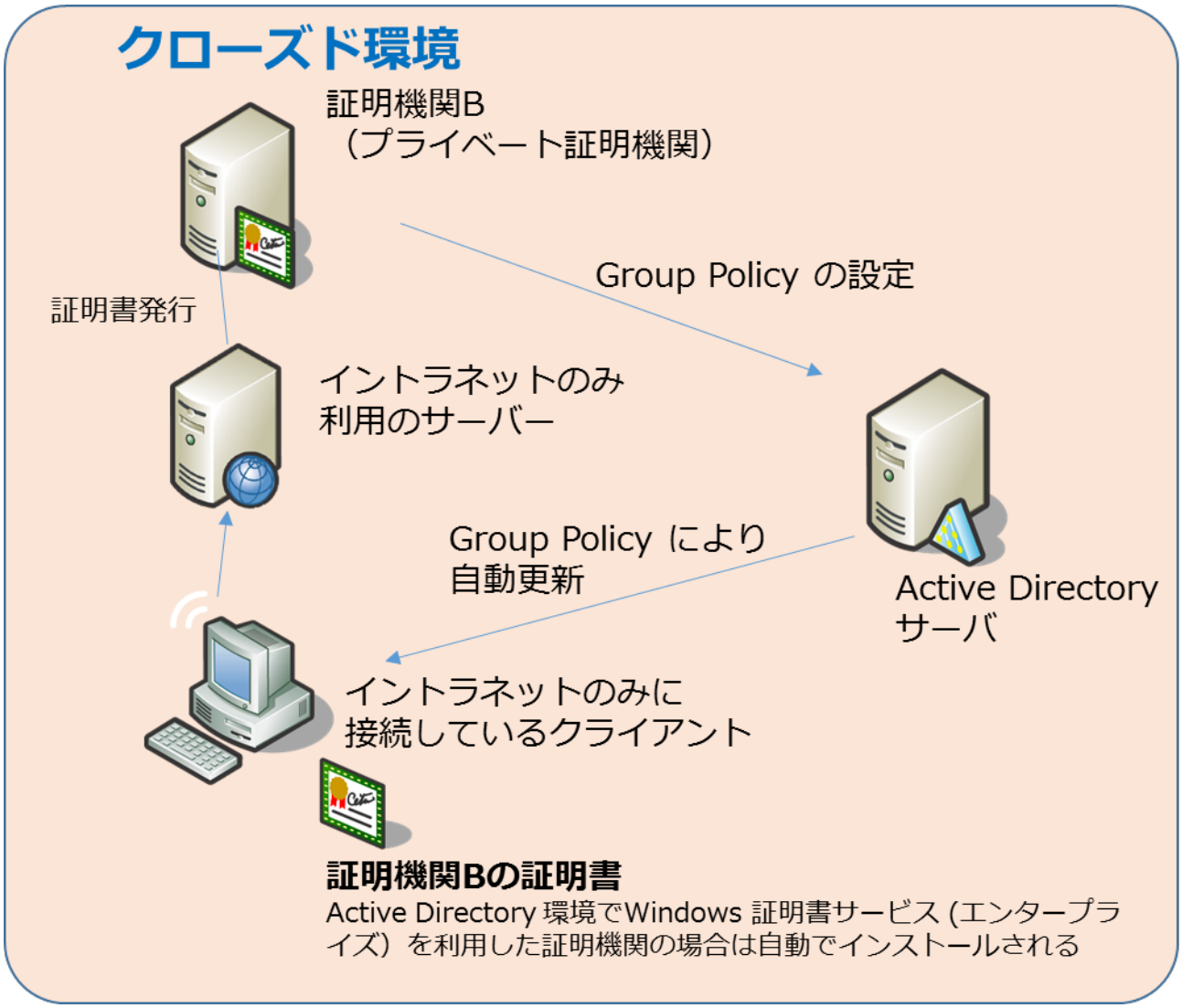
なお､サーバ証明書を発行するサービスから発行された既存のサーバ証明書を利用したサイト、あるいはテストサイトなどの URL がわかっている場合には、当該 URL にアクセスして、以下のような手順を経ることで、ルート CA の公開鍵暗号アルゴリズムおよび鍵長を確認することが可能である。

【Internet Explorer 11 で EV 証明書のサイトにアクセスする場合】

① 南京錠マーク横のサイト運営組織の表示をクリックする
② 「証明書の表示」をクリックする
③ 「証明のパス」タブをクリックする
④ 一番上に表示されている証明書（これがルート CA 証明書に当たる）を選択し､「証明書の表示」をクリックする
⑤ 「詳細」タブをクリックする
⑥ スクロールバーを一番下までスクロールさせ､「公開キー」フィールドに表示されている値（RSA　(2048 Bits)）を確認する
この例では、暗号アルゴリズムが RSA、鍵長が 2048 ビットであることがわかる


①
SECOM Trust Systems CO.,LTD. [JP]
認証実行: SECOM Trust Systems CO LTD
詳細はクリックしてください
Web サイトの認証
SECOM Trust Systems CO LTD
で、このサイトを次のように認証しました:
SECOM Trust Systems CO.,LTD.
Shibuya、Tokyo
JP
このサーバーへの接続は暗号化されています。
このサイトを信頼するべきですか?
② 証明書の表示
証明書
全般 詳細 証明のパス ③
証明書の情報
証明書の目的:
•リモート コンピューターの ID を保証する
*詳細は、証明機関のステートメントを参照してください。
発行先: www.secomtrust.net
発行者: SECOM Passport for Web EV CA
有効期間 2012/ 09/ 11 から 2014/ 09/ 18
発行者のステートメント(S)
証明書の詳細について表示します。
OK
④
SECOM Trust Systems CO LTD
SECOM Passport for Web EV CA
www.secomtrust.net
証明書の表示(V)
証明書の状態(S):
この証明書は問題ありません。
証明書のパスの詳細について表示します。
⑤
詳細
証明書の情報
この証明書の目的:
•リモート コンピューターの ID を保証する
発行先: Security Communication RootCA1
発行者: Security Communication RootCA1
有効期間 2003/ 09/ 30 から 2023/ 09/ 30
発行者のステートメント(S)
証明書の詳細について表示します。
OK
表示(S): バージョン 1 のフィールドのみ
フィールド
値
署名ハッシュ アルゴリズム sha1
発行者 Security Communication Root...
有効期間の開始 2003年9月30日 13:20:49
有効期間の終了 2023年9月30日 13:20:49
公開キー RSA (2048 Bits)
⑥
sha1RSA
プロパティの編集(E)...
ファイルにコピー(C)...
証明書の詳細について表示します。
OK

D.2. Active Directory を利用したプライベートルート CA 証明書の自動更新



Appendix B.1
暗号技術参照関係の俯瞰図

暗号技術参照関係の俯瞰図

1.　目的：

規格の参照関係について、「暗号技術参照関係の俯瞰図」（以下、俯瞰図）を作成することにより、暗号技術がどの規格で仕様として規定され、プロトコルとして利用される技術がどの規格にて決定され、応用先としてどの規格に参照されているかについての現状（2015年2月現在）を整理した。

俯瞰図を利用することによって、今後、暗号技術を標準化、及びその後の普及促進につなげていくために、どのような団体のどのような規格にアプローチしていけばよいかを提案者が判断するための、判断材料として参考となることを目的としている。

※ただし、本俯瞰図はあくまで規格の参照関係のみを整理していることに注意されたい。

俯瞰図における情報は関連分野を網羅的にカバーするものではないことおよび本資料の利用時においては最新情報でない可能性もあるため、自己の責任において最新の情報を入手して利用されたい。

なお、本WGがまとめた情報の利用に起因して生じた不利益や問題について本WG及び事務局をはじめ、CRYPTRECは一切責任を持っていない。

2.　俯瞰図の想定読者：

主に暗号技術の提案者を想定している。

3.　規格の対象範囲：

原則委員が関与している標準化団体が扱っている規格である。一部暗号についての標準化に影響力の強いNIST、ANSI、ITU等委員が直接関与しない団体の規格を含む。なお、俯瞰図中に記載された規格や標準化団体以外にも、本俯瞰図の目的に関係する規格や標準化団体が存在する可能性がある。なるべく多くの規格を載せるよう努力しているが、本俯瞰図は、網羅性を持ったものではないことに注意されたい。

表1.対象としている団体・委員会

標準化団体
ISO/IEC（JTC1/　SC27, SC17, SC31 ,SC6）
ISO（TC215, TC154）
IEC（TC57, TC65）
IEEE（IEEE802, IEEE1888）
ISA（ISA-99, ISA-100）

TCG
ARIB
ETSI
IETF
ANSI
NIST
ITU（ITU-T）

4. 俯瞰図の利用にあたって

① オレンジ色の四角は、「暗号技術の仕様」を規定している規格であることを示している。

② 赤色の四角には、「利用する暗号技術名」を規定している規格であることを示している。

③ オレンジ色もしくは赤色の四角においては、各々の四角内に、黄色の四角が存在する場合がある。黄色の四角内には、規定されている暗号技術のカテゴリを示している。暗号技術のカテゴリとは、例えば、「共通鍵ブロック暗号はB」、「公開鍵暗号はP」等のように表現している。

④ 青色の四角には、暗号技術の仕様や利用する暗号技術名自体は規定しないが、暗号技術を利用する可能性のある応用先の規格であることを示している。

⑤ 規格は複数パートに分かれているものもあるが、パートの表記は省略している。

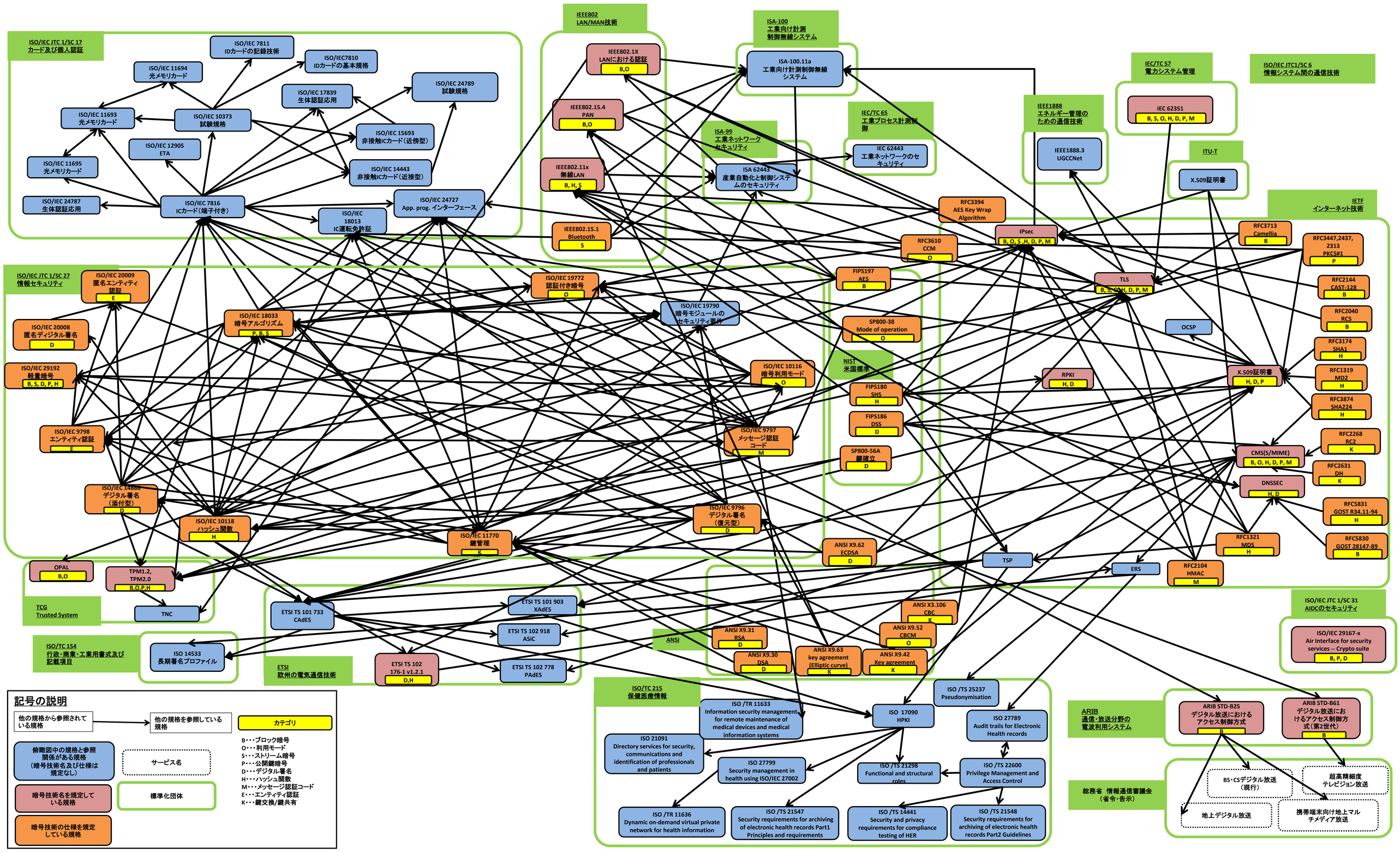
⑥ 点線の四角は、参照した規格を利用したサービス名を示している。

⑦ 俯瞰図は規格の参照関係のみを整理したものであるため、暗号技術の提案に関する情報は、「標準化提案におけるノウハウ・課題及び基本的な情報の整理」を参照のこと。

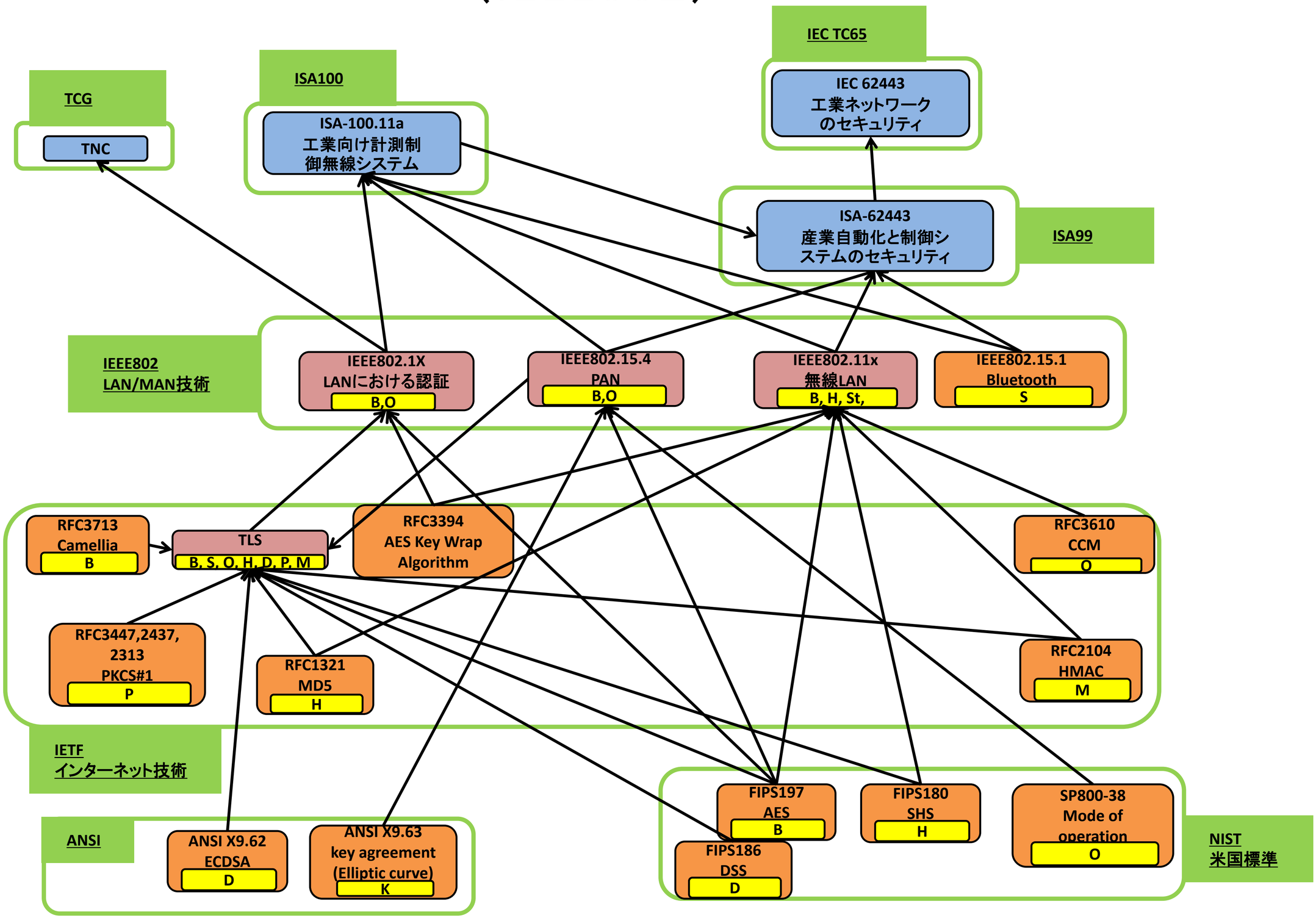
⑧ 俯瞰図中における規格の参照関係は、矢印を用いて示されている。矢印の矢のない方（根元）にある規格を矢印の矢の先端が指している規格が参照している。

以上

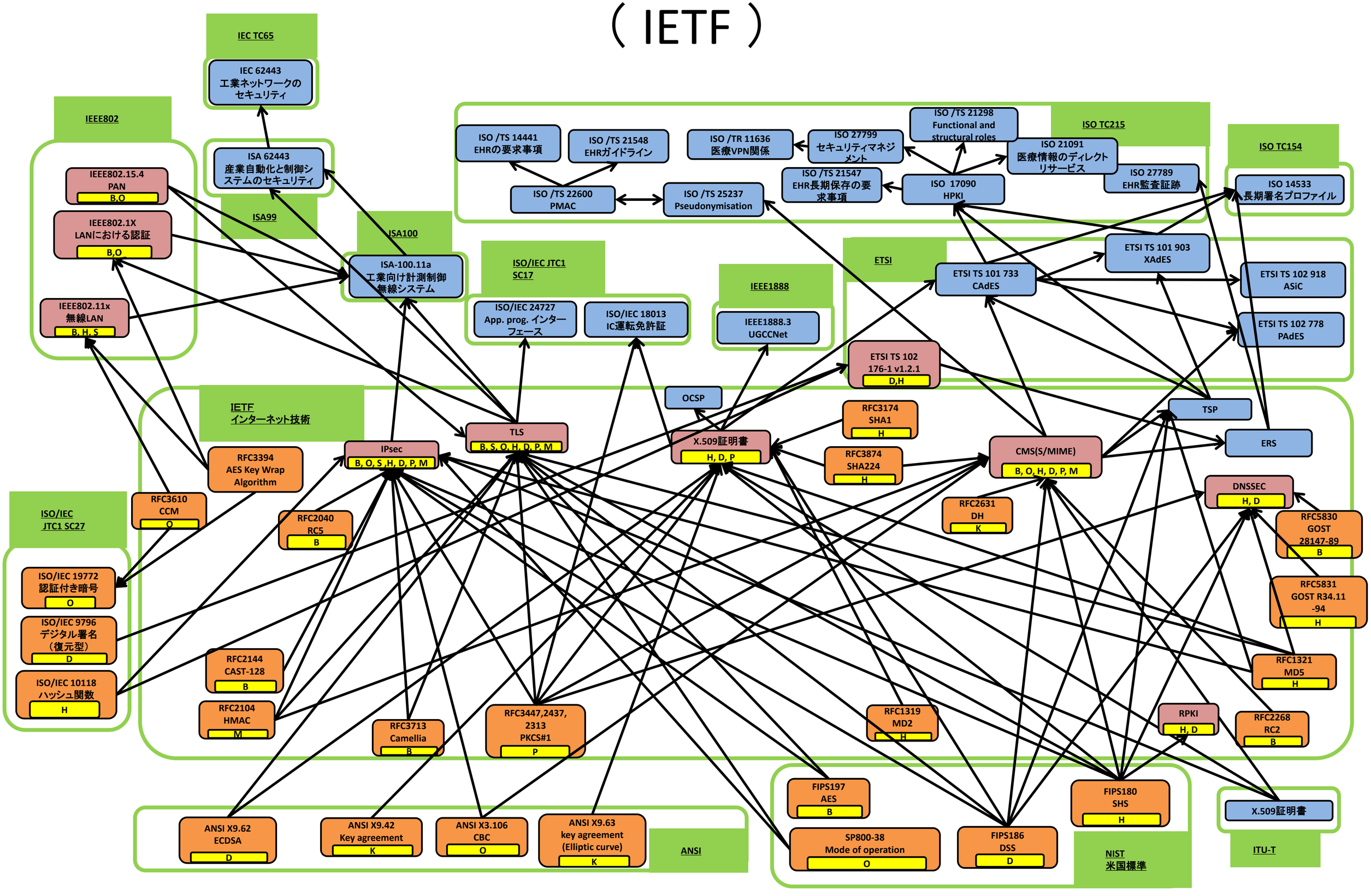
暗号技術参照関係の俯瞰図（全体像）



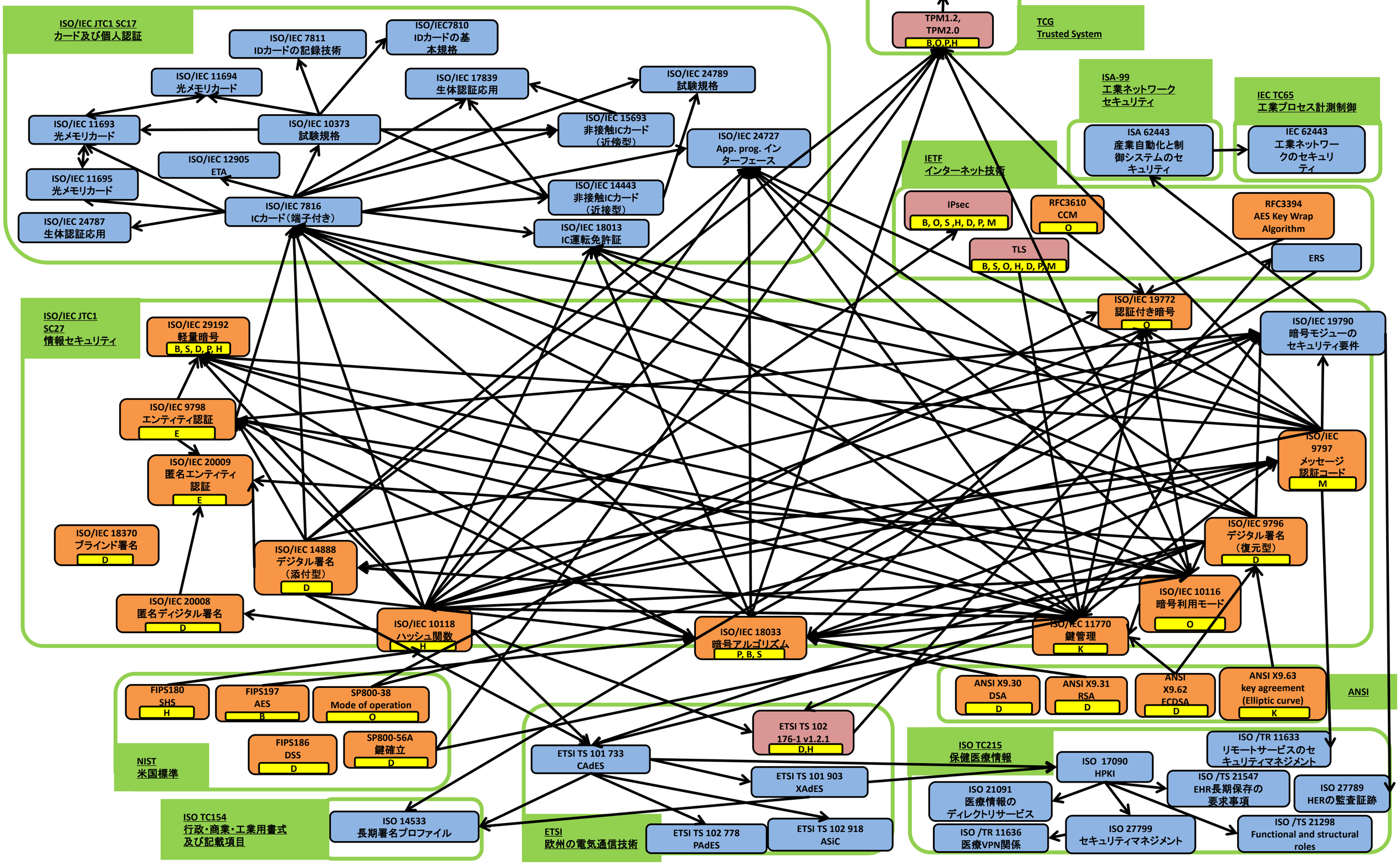
暗号技術参照関係の俯瞰図（IEEE802）



暗号技術参照関係の俯瞰図
（IETF）



暗号技術参照関係の俯瞰図（SC27）



Appendix B.2
標準化提案における交渉ノウハウ・課題及び参考情報

標準化提案におけるノウハウ・課題及び参考情報

目的：

様々な標準化機関に対する日本提案の暗号アルゴリズム標準化を横断的に支援するため、標準化提案の際に知っていると、より効率的に提案が行えるようなノウハウや、標準化団体における参考情報、標準化活動における課題等を整理する。

対象範囲：

対象とした団体は次のとおりである（順不同）。

章	団体
1.1	ISO/IEC JTC 1/SC 27
1.2	ISO/IEC JTC 1/SC 17
1.3	ISO/IEC JTC 1/SC 31
1.4	ISO/IEC JTC 1/SC 6
1.5	ISO/TC 215
2	IEC/TC 65/WG 10 及び ISA99
3.1	IEEE802
3.2	IEEE1888
4	TCG
5	ETSI
6	ARIB
7	ISA 100
8	IETF

利用にあたって：

「標準化提案におけるノウハウ・課題及び参考情報」内の情報は 2015 年 2 月に CRYPTREC 標準化推進 WG（以下、本 WG という）にてまとめられた情報である。記載された情報は関連分野を網羅的にカバーするものではなく、本資料の利用時においては最新情報でない可能性もあるため、自己の責任において最新の情報を入手して利用されたい。また、委員個人の知見に基づく情報であるため、必ずしも客観的ではない可能性があることに注意されたい。

なお、本 WG がまとめた情報の利用に起因して生じた不利益や問題について本 WG

及び事務局をはじめ、CRYPTREC は一切責任を持っていない。

「標準化提案におけるノウハウ・課題及び参考情報の整理」は、本 WG の委員からの情報をもとに、「**A)標準化活動において団体間で共通するノウハウや課題**」と「**B) 団体毎の参考情報、ノウハウ及び課題**」に分けて整理を行った。

「A)標準化活動において団体間で共通するノウハウや課題」では、上記の各団体に共通する項目をまとめていることから、ある程度一般性を持ったノウハウや課題になっていると考えられる。そのため、上記以外の標準化団体において、標準化提案を行う際にも参考にされたい。

「B)団体毎の参考情報、ノウハウ及び課題」では、上記の各団体に固有の情報をまとめた。まず、団体毎の参考情報として、技術提案先を見定めるために参考となる情報である規格の改定タイミングや技術を提案できるタイミングについての情報を盛り込んだ。また、標準化活動の稼動を、ある程度想定しておくために会合の頻度や標準化に必要となる作業についての情報も本ドキュメントに含めた。

その他に団体毎に固有のノウハウや課題についても整理を行った。上記団体やそれに類する団体に対して、技術を提案する際に参考となれば幸いである。

A）団体間で共通するノウハウや課題

＜対象団体間共通＞

今回対象とした標準化団体間で共通する<ノウハウ>及び<課題>は下記のとおりである。

<ノウハウ>

ノウハウ＜共通①＞
標準化提案を受け入れてもらうためには、標準化に関わる他の「人」との関わりが必要不可欠である。 **例えば、一方的に提案するのではなく、周りの人たちとの協力関係を築き、ネゴシエーションしながら標準化を行うと提案がスムーズに受け入れられやすい。** **また、過去の審議経緯や利害関係等を十分に把握していると審議を進めるのに有利である。継続的に規格策定の場に関わっている人とコミュニケーションをとり、審議の経緯等を知っておくとよい。**

ノウハウ＜共通②＞
標準化したい規格において関連する技術分野とのリエゾン関係等がない場合、他の関連する規格を調査したうえで、他の組織と横断的に連携して標準化を行うことが重要である。

<課題>

課題＜共通①＞
日本の場合、企業が標準化団体に参加するため、それまでの担当者が異動してしまうと担当委員が変わるが、欧米の場合、個人が標準化の仕事として参加しているケースが多いので、企業を移っても所属企業名が変わるだけでその人物は引き続き参加する。このように、欧米では、日本に比べて、継続して標準化活動に携わる人が多い。そのため、日本の委員は海外委員との関係構築にエネルギーを要している。

B）団体毎の参考情報、ノウハウ及び課題

1. ISO 及び ISO/IEC 共通

ISO 及び ISO/IEC JTC 1 に共通する<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 投票権を取る条件
国毎に単一の代表組織（National Body、日本の場合は JISC）が 1 票の投票権を持つ。そのため、国として P-member に求められる活動を継続することが重要である。（ISO/IEC Directives、JTC 1 Supplement、JTC 1 Standing Documents 等による。）

● 投票権は誰に帰属しているか
投票権は国の代表組織（National Body）に帰属する。

● 平均的に標準ができるまでどのくらいの期間が必要か
2 年から 4 年程度である。（36 ヶ月が標準的な規格開発期間であるが、24 ヶ月、48 ヶ月という規格開発期間も選択可）

● 規格化のプロセス
標準的なプロセスは 6 段階であり、NP (New Work Item Proposal), WD (Working Draft), CD (Committee Draft), DIS (Draft International Standard), FDIS (Final Draft International Standard)を経て、IS (International Standard)となる。この他にも、NP に先行して PWI (Preliminary Work Item)から開始する場合や、Fast-Track の場合等がある。

● 規格策定のために利用するツール
ISO テンプレート（Word 形式のマクロで提供されている）を用いた文書作成が必要となる。国内委員会に登録してもらい、Web システムを用いて文書共有（ダウンロード、必要があればアップロードも）を行う。

<ノウハウ>

ノウハウ＜ISO 及び ISO/IEC 共通①＞
標準化提案において、各国で一定の実績のある規格が TC/SC メンバ又は ISO と提携関係にある国際的標準化機関から ISO 事務総長に国際規格提案された場合、ある条件を満たせば、Fast-Track を利用して標準化プロセスを短縮できる場合がある。

ノウハウ＜ISO 及び ISO/IEC 共通②＞
国の代表が集まる標準化団体で提案を行う際には、国際的な場で提案を行う前に、国内の議論の場で調整を行うべきである。

ノウハウ＜ISO 及び ISO/IEC 共通③＞
PWI に登録されたタイミングで技術を募集する可能性もあるので、国内委員会に参加し PWI の登録状況を確認することも、標準化提案先を探すうえで有効である。

<課題>

ノウハウ＜ISO 及び ISO/IEC 共通①＞
他国から提案された Fast-Track には、対応のための十分な検討期間が無い場合が生じる。

1.1. ISO/IEC JTC 1/SC27

ISO/IEC JTC 1/SC 27 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
5 年毎に規格が見直され、改廃について議論される。

● どのようなタイミングで技術を提案できるか
定期的な見直しの時期に、理由を付けて改訂を提案することができる。その他任意の時期に追補(amendment)として提案することも可能であるが、理由は定期見直しの場合より強いもの（緊急性）が求められるように思われる。

● リエゾン関係がある団体
次の委員会等とリエゾン関係にある。 ISO 関係：ISO/IEC JTC 1/SC 17, ISO/TC 215 他 IEC 関係：IEC/TC 57, IEC/TC 65 他 他の組織：IEEE, ITU 他 参考： http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=45306

● 技術提案の手続き等
Call for Contribution に対して暗号技術を提案する場合、NB (National Body)提案として提案する場合、Expert 提案として提案する場合が考えられる。日本において暗号技術の提案を行う場合は、どちらの場合であっても、日本の国内委員会である情報処理学会情報規格調査会の SC27/WG2 小委員会に参加し、委員会で合意を得る必要がある。

● 会合が年何回あるか
WG 会合が年に 2 回（4～5 月と 10～11 月）行われる。総会は年 1 回（4～5 月の WG 会合の次の週）行われる。

● 電話会議の時間帯と時間帯の決め方
時間帯は、13:00 UTC 等である。電話会議の時間帯は欧州に合わせる形で決まっている。

● 会議の開催場所
各国からの提案ベースで決定されている。これまで最近開催された順に、メキシコ、香港、韓国、フランス、イタリア、スウェーデン…で開催されてきた。今後はマレーシア、インドでの開催が予定されている。

● 提案者に最低限必要となる作業
提案時には提案理由説明（文書および会合での発表資料、会合での発表と質疑応答対応）が必要となる。 提案が受け入れられた場合は、プロジェクトエディタとなった場合は WD から IS までの文書作成および改訂作業、各国、エキスパートからのコメントへの対応が必要となる。 その他、日本からのエキスパート寄書、NB 寄書の作成、NB 投票案の作成（これらは参考情報④に書かれているように、日本の国内委員会での合意が必要となるため、そこでの説明も求められる）が必要となる。さらに国際の場での合意形成のため、他国との折衝が必要になることもある。

● 国内委員会おけるコンタクト先
一般社団法人情報処理学会情報規格調査会 所在地 〒105-0011 東京都港区芝公園 3-5-8 機械振興会館 308-3 電話番号 03-3431-2808 FAX 03-3431-6493

<ノウハウ>

ノウハウ①
技術の安全性が示されていることが標準となるための必須条件であるため、学術論文や国家的なプロジェクトでの安全性評価があると受け入れられやすい。日本の CRYPTREC での評価結果も信頼されている。

<課題>

課題①
ある規格について、日本から複数の技術を提案する場合、あるいは同規格内に既に国内技術が掲載されているときに追加提案を行う場合は、その提案が互いの普及の妨げとなる可能性があるため、国内での議論が必要となることがある。

1.2. ISO/IEC JTC 1/SC 17

ISO/IEC JTC 1/SC 17 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
5 年毎に規格が見直され、改廃について議論される。

● どのようなタイミングで技術を提案できるか
PWI（予備段階）または NP（提案段階）で新規提案を行う場合、定期的な見直しの際に提案する場合、Call for Contribution（寄書募集）に対応して提案する場合、等で新たに技術を提案できる。

● リエゾン関係がある団体
次の委員会等とリエゾン関係にある。 ISO 関係：JTC 1/SC 6, JTC 1/SC 27, JTC 1/SC 31, JTC 1/SC 37, ISO/TC 68 他 参考：ISO/IEC JTC 1/SC 17 Cards and personal identification ホームページ http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=45144

● 技術提案の手続き等
暗号アルゴリズム自体は ISO/IEC JTC 1/SC 27 等が定めたものを参照する前提であるため、暗号アルゴリズムの提案自体は行われない。

● 会合が年何回あるか
SC17 の傘下で活動中の 8 WG で計年 25 回程度会合が行われる（WG により年 1～5 回の開催）。総会は 1 年に 1 回行われる。

● 電話会議の時間帯と時間帯の決め方
WG 会議・TF 会議・複数 WG の合同会議等で電話会議・WebEx を実施する場合がある。また、対面の会議に電話会議・WebEx の形で部分参加者が加わる場合がある。 1 回 2 時間程度を目途としており、主要参加国の稼働時間帯を中心に参加国の時差を考慮して時間帯を定めるが、複数回の場合は主要参加国の稼働時間帯を考慮した持回りを検討する。

● 会議の開催場所
主要参加国の持回りで開催されている。 SC17 総会の日本開催はほぼ 10 年に 1 回開催している（これまでに 4 回開催)。 SC17 傘下 WG の日本開催は毎年何れかひとつ以上の WG を日本で開催している (近年)。

● 提案者に最低限必要となる作業
まずは、NB 提案とするための国内委員会における合意形成及び提案内容のリファインが必要である（国際標準とすべき内容の特定)。 その後、PWI または NP としての提案資料及び関連寄書の作成、WD 案または CD 案の作成も行う必要がある。 関連国際会議への参加と提案に対する各国意見への対応（国内検討・国際応答)。 可能な場合は Project editor または Project co-editor を引受ける。

● 国内委員会のコンタクト先
SC 17 国内委員会事務局:　JBMIA 一般社団法人 ビジネス機械・情報システム産業協会 〒108-0073　東京都港区三田 3-4-10　リーラヒジリザカ　7 階 Tel: 03-6809-5149　　Fax: 03-3451-1770 URL: http://www.jbmia.or.jp

● その他
国際標準化は国を単位とする活動であるが、国際会議に対応している委員個人に負う部分が大きいのも現実である。単なる情報収集や一方的な自己主張でないことが理解されれば海外からの評価や信頼を得られる。その意味で合意形成のために何をどのように主張したかの積み重ねが大切である。そのためには継続的に国際会議に対応してプレゼンスを示すとともに、外国の委員と個人的にも話ができるようになることが必要である。委員の所属組織は派遣している委員がそのように活動できる環境を用意することがビジネスにプラスになるとの認識を強く持つ必要があり、その集積が技術力強化にもつながると考えられる。

<ノウハウ>

ノウハウ①
ISO/IEC JTC 1/SC 17 はカード及び個人識別を対象とし、各種カードの要素技術からその利用システムまでを含む国際標準化を担当している。

ノウハウ②
カード利用の進展に伴い、JTC 1 の関係 SC, ISO の関係 TC 等と連携して対応を行う場面が増加している。 （例）セキュリティ・生体認証関係： JTC 1/SC 27、JTC 1/SC 37 無線インタフェース関係： JTC 1/SC 6、JTC 1/SC 31 金融サービス関係： ISO/TC 68、ISO/TC 68/SC 7

<課題>

課題①
国際標準化対応のエキスパート養成を継続して推進する必要がある。

1.3. ISO/IEC JTC 1/SC 31

ISO/IEC JTC 1/SC 31 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
5 年毎に規格が見直され、改廃について議論される。

● どのようなタイミングで技術を提案できるか
PWI（予備段階）または NP（提案段階）で新規提案を行う場合、定期的な見直しの際に提案する場合、Call for Contribution（寄書募集）に対応して提案する場合、等で新たに技術を提案できる。

● リエゾン関係がある団体
次の委員会等とリエゾン関係にある。 JTC 1/SC 17 他 参考： http://www.iso.org/iso/iso_technical_committee.html?commid=45332

● 提案者に最低限必要となる作業
まずは、NB 提案とするための国内委員会における合意形成及び提案内容のリファインが必要である（国際標準とすべき内容の特定）。 その後、PWI または NP としての提案資料及び関連寄書の作成、WD 案または CD 案の作成も行う必要がある。 関連国際会議への参加と提案に対する各国意見への対応（国内検討・国際応答）。 可能な場合は Project editor または Project co-editor を引受ける。

● 国内のコンタクト先
一般社団法人情報処理学会情報規格調査会 所在地 〒105-0011 東京都港区芝公園 3-5-8 機械振興会館 308-3 電話番号 03-3431-2808 FAX 03-3431-6493

<ノウハウ>

ノウハウ①
SC31 では、規格策定の際に実機を使ったデモンストレーションが必要となるため、ソフトウェアベンダやハードウェアベンダと協力して規格策定に臨む必要がある。

1.4. ISO/IEC JTC 1/SC 6

ISO/IEC JTC 1/SC 6 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 会議の開催場所
開催地に特に偏りはない。

● 国内委員会におけるコンタクト先
一般社団法人情報処理学会情報規格調査会 **所在地 〒105-0011 東京都港区芝公園 3-5-8 機械振興会館 308-3** **電話番号 03-3431-2808** **FAX 03-3431-6493**

<ノウハウ>

ノウハウ①
国もしくはリエゾンしている組織単位で提案が行われているため、事前に国内で調整が必要になる。

<課題>

課題①
投票の時には組織間の駆け引きが行われているため、純粋に良い技術が採択されているとは思えない。

1.5. ISO/TC 215

ISO/TC 215 について、<参考情報>、<ノウハウ>、<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
3 年毎に規格が見直され、改廃について議論される。

● どのようなタイミングで技術を提案できるか
年 2 回の TC215 会議において、PWI（予備段階）での提案が可能である。提案後、NP 投票に進むにはドラフト（アウトラインでも可）を作成し、WG 内での議論を経た上で総会での承認が必要である。

● リエゾン関係がある団体
次の委員会とリエゾン関係にある。 ・DICOM、IHE、IEC/SC62A 他 参考： http://www.iso.org/iso/home/standards_development/list_of_iso_technical_committees/iso_technical_committee.htm?commid=54960

● 技術提案の手続き等
基本的に P メンバの国のエキスパートであれば提案可能である。

● 会合が年何回あるか
通常年 2 回の会合が行われる。WG2、JWG7 については年三回の場合もある。

● 電話会議の時間帯と時間帯の決め方
ISO/TC215 会議の中で調整して決定する。

● 会議の開催場所
P メンバ各国の持ち回りで開催されることとなっている。

● 提案者に最低限必要となる作業
各種ドキュメントの作成、タスクメンバーの取りまとめと意見調整、TC215 会議における担当時間の議事進行、コメント処理の実施、WG セクレタリとの進捗の調整等が必要である。

● 国内におけるコンタクト先
ISO/TC215 国内対策委員会事務局： 一般財団法人 医療情報システム開発センター 〒162-0825　東京都新宿区神楽坂一丁目１番地　三幸ビル２階 標準化推進部 03-3267-1924

● その他
WG セクレタリが進捗のキーパーソンのため、セクレタリの仕事ぶりが全体の効率を左右する。

<ノウハウ>

ノウハウ①
米国が TC の Chair、セクレタリを担当しているので、米国との良好な関係が円滑な標準化作業の推進につながる。

<課題>

課題①
IT セキュリティの TC215 と医療機器の安全管理の IEC/SC62A との間で医療情報ソフトウェアの扱いについて、勢力争いが発生している。JWG7 という Joint WG で議論しているが JWG7 と WG4 の間での駆け引きもあり、様々な配慮が必要。

2. IEC/TC 65/WG 10 及び ISA 99

IEC/TC 65/WG10 及び ISA99 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
IEC 62443 の定期見直しは不明。

● どのようなタイミングで技術を提案できるか
ISA 99 で、NP（提案段階）もしくは WD（原案作業段階）にてコメントを提出する段階で技術の提案を行うことができる。もしくは、新規プロジェクトを立ち上げた際に技術の提案を行うことができる。

● リエゾン関係がある団体
次の委員会等とリエゾン関係にある。 **・ISO/IEC JTC 1 SC 27/WG 1, ISA 99 他**

● 技術提案の手続き等
暗号アルゴリズム自体は、情報セキュリティ業界、プロセス制御業界でエキスパートとされるグループとして、ISO/IEC JTC 1 SC 27/WG 2 で定めたものを参照する。

● 会合が年何回あるか
IEC/TC 65/WG 10 では、年 2 回会合が行われる。 **ISA 99 では、年 2 回程度の会合が行われる。**

● 投票権を取る条件
ISA 99 の投票権は、ISA 99 の当該 WG に申請し、認められることにより、エキスパート（会社ごとに一名）に与えられる。 **IEC/TC 65/WG 10 では、国の代表組織（National Body）が 1 票の投票権を持つ。そのために、国として P-member に求められる活動を継続することが重要である。**

● 投票権は誰に帰属しているか
ISA99 の投票権はエキスパート個人に帰属する。 **IEC/TC65 の投票権は、国の代表組織に帰属する。**

● 平均的に標準ができるまでどのくらいの期間が必要か
3～6 年程度である。

● 電話会議の時間帯と時間帯の決め方
ISA 99 の電話会議は、毎週もしくは 2 週間に 1 回程度行われる。時間帯は WG リーダもしくは、WG/TG リーダが議論をするために必要なメンバの都合により決めるため、米国東海岸の昼の時間（日本の夜中）となることが多い。

● 規格化のプロセス
WG,CD,CDV,FDIS を経て、国際規格の発行となる。

● 会議の開催場所
IEC/TC 65/WG 10 は、参加国が場所を用意し、ホスト申請して、参加メンバの合意を基に決められる。

● 提案者に最低限必要となる作業
ドラフトを作成し、提案して、コメントを受け、修正を加えて、WD→CD→CDV→FDIS→IS とすることであるが、結局、投票でスムーズに進めるためには、投票権を持っている人達と議論を行い、問題点を洗い出してその解消を行い、支持票数が確保され、反対票が問題化しないよう対策するなど、根回し、フォローが必要となる。

● 国内のコンタクト先
一般社団法人 日本電気計測器工業会（JEMIMA） 所在地： 〒103-0014 東京都中央区日本橋蛎殻町 2-15-12(計測会館) TEL: 03-3662-8181 FAX: 03-3662-8180

<ノウハウ>

ノウハウ①
IEC/TC 65/WG 10 がセキュリティの規格に携わっており、IEC 62443 シリーズを担当している。 IEC 62443 シリーズは、4 つのパートに分かれており、 ① Part 1、IEC/TS 62443-1-1、IEC 62443-1-2、IEC 62443-1-3、IEC 62443-1-4 ② Part 2、IEC 62443-2-1、IEC 62443-2-2、IEC 62443-2-3、IEC 62443-2-4 ③ Part 3、IEC/TR 62443-3-1、IEC 62443-3-2、IEC 62443-3-3 ④ Part 4、IEC 62443-4-1、IEC 62443-4-2 となっている。 IEC/TC 65/WG 10 が直接議論をおこなっているのは、IEC 62443-2-4 のみであり、残りの標準は、ISA S99 が議論を行い策定し、IEC に提出し、投票にかけられる。従って、IEC 62443-2-4 は、IEC/TC 65/WG 10 で対応できるが、残りは、ISA 99 で議論を行うことが必要である。 ISA 99 の議論は、毎週（または 2 週に一回）開催される Web/電話会議に参加して、ドラフト作成、コメント対応に協力する。さらに、Face-to-face Meeting（通常米国）に参加して“仲間”として認識されることが重要である。 暗号がキーワードとして出てくるドキュメントは、IEC 62443-3-3、IEC 62443-2-4、IEC 62443-4-2 である。

ノウハウ②
ISA には、ISCI(ISA Security Compliance Institute)という組織があり、集まったメンバが参加会費を払う。この資金を使って、フルタイムで規格を作成するエキスパートを雇い、制御システムの組み込み機器の評価・認証基準、フレームワークを立ち上げた。これは、短期間で効率が良く高品質な規格を作る手法として考慮されている。その後に、この評価基準が IEC 62443-4-1 にも導入された。 このように、ISCI で規格として提案されることで、ISA 99 で ISA 62443 として取り込まれ、最終的に IEC 標準となるケースもある。

<課題>

課題①
ISA の電話会議の時間であるが、米国東部時間で以下の時間であり、日本時間では夜遅くから深夜となり、フォローするには、夜勤のような体制を所属組織に認めてもらう必要がある。 月: WG3:　ISA-62443-1-1, Terminology, Concepts and Models, 9:00 AM ET. WG2:　ISA-62443-2-1, CSMS, 10:00 AM ET. WG4 TG4: ISA-62443-4-2, Derived Requirements, 11:30 AM ET. 火: WG4 TG5: ISA-62443-1-3, Security Metrics 01:00 PM ET 水: WG4 TG2: ISA-62443-3-3, Foundational Requirements 11:00 AM ET 木: WG4 TG6: ISA-62443-4-1, Product Development Requirements, 11:00 AM ET. 金: WG4 TG3: ISA-62443-3-2, Zones and Conduits, 10:00 AM ET.

3. IEEE 共通

IEEE に共通する<参考情報>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
逐次規格は改訂されている。

● どのようなタイミングで技術を提案できるか
年間 6 回行われる（対面での）会議の際に、いつでも提案を行うことができる。

● 投票権を取る条件
投票権は 4 回の連続する Plenary のうち 3 回目の出席で取得できる。ただし、加えて以下の条件がある。 ・3 回のうち 1 回は Interim で代替可 ・投票権付与は Plenary のみ –会期中のセッションの 75%以上出席しないと出席とは認められない（Base Slot が 18 コマだと 14 コマ以上出席が必要） –投票権維持には、直近 4 回の Plenary 中 2 回(1 回は、Interim でも可能)に出席が必要。

● 規格化のプロセス
Study Group を組成し、Project Authorization Request のフェーズに入る。そこで規格化が決まれば、Task Group を組成し、「Requirement Down Section Procedure」,「Call for Proposal」,「Proposal Presentation」,「Down Selection Merge」,「 TG Draft」,「Internal Comment Resolution」,「WG Letter Ballot」,「Comment Resolution」,「TG Draft」,「Sponsor Ballot」,「Comment Resolution」,「TG Draft」,「Sponsor Ballot」を経て、Standard となる。

3.1. IEEE802

IEEE802 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● リエゾン関係がある団体
次の委員会とリエゾン関係にある。 **・ISO/IEC JTC1 SC6**

● 技術提案の手続き等
委員会のメンバであれば、暗号アルゴリズムを提案できる人に制限は特にない。

● 会合が年何回あるか
年 6 回（奇数月）に会合が行われる。なお、一回毎に Plenary と Interim が交互に開催されることとなっている。

● 投票権は誰に帰属しているか
投票権は個人に帰属する。

● 平均的に標準ができるまでどのくらいの期間が必要か
5 年程度である。ただし、ある程度安定した仕様が出来た段階で、業界団体によるデファクト化が進行する。

● 電話会議の時間帯と時間帯の決め方
電話会議の時間帯等は会合における動議により決定される。

● 会議の開催場所
開催地は北米が中心となっている。近年では、アジア、欧州での開催を年に一回ずつ入れるようになってきた。

● 提案者に最低限必要となる作業
寄与文章の提出とプレゼンを行ったうえで、以後のリーダーシップを取る必要がある。

● 規格策定のために利用するツール
IEEE-SA が用意するドキュメントサーバー等

● IEEE802.11 の技術を普及させるために
IEEE802.11 では、民間コンソーシアム（Wi-Fi Alliance）での標準化も併せて行わないと、普及が厳しくなる。

● IEEE802.15 の技術を普及させるために
IEEE802.15 では、民間コンソーシアム（ZigBee、Bluetooth、Wi-SUN）での標準化も併せて行わないと、普及が厳しくなる。

● 国内におけるコンタクト先
IEEE ジャパン・オフィス **〒107-0062** **東京都港区南青山 1-1-1 新青山ビル東館 19 階** **Tel: 03-3408-3118** **Fax: 03-3408-3553** **E-mail : ieee-japan@ieee.org**

<ノウハウ>

ノウハウ①
IEEE802.11 では、事前に開発された規格を持ち込むことに反発があるので、事前に開発していた規格であったとしても、最初から全てを提案せずに、小出しに提案するとよい。

ノウハウ②
IEEE802 では、Robert's Rule of Order という議事運営規則によって、会議が進行する。Robert's Rule of Order の理解が重要なリーダーシップのためのスキルとなっている。

<課題>

課題①
フォーラム型標準化であり、投票権が個人に帰属することが、大きな特徴である。また、技術の採択には、75%の賛同を得る必要があり、他社とのロビーイングが重要な戦略となる。

3.2. IEEE1888

IEEE1888 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● リエゾン関係がある団体
特になし。

● 技術提案の手続き等
WG メンバになる必要がある。

● 会合が年何回あるか
会合の年間回数は特に決まっておらず、不定期に行われる。

● 投票権は誰に帰属しているか
投票権は個人に帰属する。

● 平均的に標準ができるまでどのくらいの期間が必要か
2 年程度である。

● 電話会議の時間帯と時間帯の決め方
電話会議の時間帯は、メンバ間で決定する。

● 会議の開催場所
主に中国にて開催される。

● 提案者に最低限必要となる作業
積極的な提案活動が必要である。

● 規格策定に利用するツール
IEEE-SA が用意するドキュメントサーバー等

<ノウハウ>

ノウハウ①
日本側では、グリーン東大プロジェクトにて技術の検討が行われる事が多いので、まずはそこにコンタクトすると進めやすい。

<課題>

課題①
中国、日本が中心になっているため、EU や US における知名度が高いとは言えない。

4. TCG

TCG について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
定期的な見直しは特に定められておらず、必要に応じて規格が制定・改訂される。

● どのようなタイミングで技術を提案できるか
WG 毎に提案できるタイミング・フェイズが異なっている。提案の通りやすさとしては、基本的に、新たなデマンド・ユースケースなどを議論している際に、提案が最も採用される可能性が高い。

● リエゾン関係がある団体
TCG は国の組織、学会・大学からのからのリエゾンを受け入れる仕組みがあり、日本からは、IPA(METI)/NICT の方がリエゾンとして参加している。 ※リエゾンメンバは、希望する WG への参加(投票権なし)、および、メンバ会議へ招待。毎回 NSA/BSI/CESG/DCSSI などのメンバが参加している。 (http://www.trustedcomputinggroup.org/about_tcg/industry_participation)

● 技術提案の手続き等
TCG 会員であれば、誰でも提案することが可能。リエゾンメンバからの提案は、従来はできなかったが、去年の理事会で可能となった。

● 会合が年何回あるか
メンバーミーティングと呼ばれる会合が年 3 回 (2 月、6 月、10 月)行われる。その他に、WG 毎に会議(電話会議・会合)が、計画実施されている。

● 投票権を取る条件
一企業につき一票が原則であり、各 WG での投票権はその WG の会議への参加状況によって決まる。現在、規則見直しが行われている。最新の正確な規則は TCG のサイトを参照のこと。

● 投票権は誰に帰属しているか
投票権は参加企業に帰属する。

● 平均的に標準ができるまでどのくらいの期間が必要か
数年程度である。

● 電話会議の時間帯と時間帯の決め方
TCG における電話会議については、各 WG の議長が、メンバの意見を聞きながら決定する。 JRF(TCG 日本支部) の電話会議は、隔週 水曜日 9:00-10:00(JST) と決まっている。

● 規格化のプロセス
一般的な規格・仕様のプロセス 1. WG で規格・仕様の策定、および、承認(投票) 2. WG の上位グループ TC (Technical Committee) で、議論・承認(投票) 3. BoD (理事会)で、TCG メンバのインターナルレビューおよび、IP レビュープロセス(90 日間)の承認(投票) 4. BoD(理事会) で、公開レビューの承認(投票) 5. BoD(理事会) での、公開の承認(投票) TPM 暗号アルゴリズム 採用のプロセス 1. TPM WG に、TCG メンバが採用要求 2. TPM WG で技術的に評価、および、BoD へのレポート(投票) 3. BoD での承認(投票) 4. Registry 仕様書の更新 5. TC にて評価、承認(投票) 6. BoD にて承認(投票): インターナル・IP レビュー 7. BoD にて承認(投票): 公開レビュー、 8. BoD にて承認(投票): 公開

● 会議の開催場所
年間 3 回のメンバ会議は、アメリカ(2 月)→ヨーロッパ(6 月)→アジア(10 月)、から適当に選ばれていたが、アジアからの参加者が少なくなり、現在は、アメリカ(2 月)→ヨーロッパ(6 月)→アメリカ(10 月)から選ばれるようになった。 WG の会議は、WG で決定される。

● 提案者に最低限必要となる作業
WG への参加、提案活動、および、その内容への質問対応が必要である。また、規格書の作成・修正も必要となる。

● 規格策定に利用するツール
Microsoft Office Word – 規格・仕様書 プレゼンなどのためのツールは、特に規定はない。

● 国内におけるコンタクト先
Japan Regional Forum(JRF) http://www.trustedcomputinggroup.org/jp/jrf_in_tcg 連絡先 Anne Price +1-602-840-6495 日本語での連絡先： 重村ゆう子 jp_press@trustedcomputinggroup.org

● その他①
自動車業界のリード、インターネットインフラが整っており IoT(Embedded System)の技術をもっている日本に TCG は興味を持っている。 これらのエリアで、リードできるポジションにいると思われる。（2015 年現在）

● その他②
TPM1.2 は今後改定が行われないため、今後暗号技術を提案することができない。

<ノウハウ>

ノウハウ①
TCG は民間企業による NGO(http://www.trustedcomputinggroup.org/about_tcg) であり、参加企業のビジネスへの貢献がその活動目的の組織である。したがって一般的であるが、下記のことが必須である。 - 企業として参加して、組織活動へ貢献を行う。 - 特に、企業のビジネスのために必要な標準規格・仕様を制定する WG、および、ユースケースを議論する WG をより活発にするために貢献を行う。 - 最終決定を行う理事会との情報を共有できる人脈を作る。 o 他の WG とは異なり JRF(TCG 日本支部)は、理事会直轄の組織なので、そのチャネルを使用することは有効である。例えば、Camellia が TPM2.0 の仕様に採用されたのは、本来ならばそのアルゴリズム採用を推したい民間企業が行うのが通常であるが、例外的に、METI /IPA/JRF の協力のもと採用された。 - 活動している WG の議長など、ステークホルダーとの積極的なコミュニケーション(会議のみならずメールなどのツールを使用した) - メンバ会議でのロビー活動

ノウハウ②
ビジネスケースが成立し、マーケットが広大もしくは、必須であればビジネス追求が目的としている企業の集団である TCG は動きが早い。 例えば、政府が調達要件の必須とし、さらに、その技術・アルゴリズムを採用した製品を提供する企業があれば、規格に速やかに採用される。

<課題>

課題①
利益追求が目的である民間企業の組織のため、ビジネス上の課題・デマンドと、ユースケース、そこに参加企業のビジネスケースが成立するかが重要とある。TPM2 の最終版がほぼ決まり、またインスタンスとしてのプラットフォームの仕様書、例えば、PC 用、サーバー用、モーバイルフォン用、などもほぼ最終版もリリースされる。したがって、暗号アルゴリズムをこれらの規格で採用提案は困難であると思われる。もちろん、ビジネスケースが成立するのであれば可能である。例えば、日本政府の調達必須要件となり、かつ、企業がそれを提供できる、など。 したがって、今後、TCG で策定する規格・仕様書での採用を考えた場合、今後の新たなビジネスとして着目してエリアでの提案が、可能性が高いと思われる。そこでのデマンド、および、ユースケースを提案、そこで使用に最適な暗号アルゴリズムとしての採用を提案する。 TCG が着目しているエリアとして、下記があると思われる（2013JRF ワークショップにて公開された情報より)。 - インダストリー o 自動車 o 電気・ガスなどのインフラ o 銀行・金融 - テクノロジー o IOT o Cloud o Embedded System

5. ETSI

ETSI について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
定期的な見直しは特に定められておらず、必要に応じて規格が制定・改訂される。

● どのようなタイミングで技術を提案できるか
提案自体はミーティングの場で行うことができる。その後、TC に受け入れられるかどうか審議される。

● リエゾン関係がある団体
次の委員会とリエゾン関係にある。 **・ISO/TC154 他**

● 技術提案の手続き等
ETSI/TC ESI では暗号アルゴリズム自体は規格化していない。推奨暗号スイート（署名やハッシュアルゴリズム）の規格は存在するが、このリストに載るアルゴリズムは FIPS や ISO, IETF 等の他の規格を参照している。

● 会合が年何回あるか
通常会合は年 4 回程度行われる。ここ最近は電子署名に関する規制(EU Regulation)に基づく議題が多く、1～2 か月に 1 回程度開催されている（オンラインミーティング含)。

● 投票権を取る条件
ETSI 規格に対してであれば、Associate Member クラス以上であれば投票権は得られる。しかし、欧州規格(EN)の投票権は得られない（国の標準化機関のコンタクトが必要)。

● 投票権は誰に帰属しているか
投票権は、会員組織に帰属する。

● 電話会議の時間帯と時間帯の決め方
ETSI/TC ESI の事務局によって決定される。ほぼ全て欧州のメンバであるため、欧州の時間が基準となる。

● 規格化のプロセス
Technical Specification(TS)や Technical Report (TR)については TC での承認が得られれば発行される。ETSI Guide(EG) や ETSI Standard(ES)では会員による投票を経て発行される。EN では各国による投票を経て発行される。電子署名に関する規格は TS や TR で主であったが、前述のとおり、現在はこれらの規格を EN として再構築する作業が行われている。

● 会議の開催場所
会議の開催場所は毎回変わる。ミーティング後に次回の開催場所が決定される。主に欧州であるが、米国で開催したこともあり、欧州外での開催も実現可能である。

<ノウハウ>

ノウハウ①
現在は直接 ETSI で規格作成する立場ではなく、修正提案を行う立場にある。提案に際しても EU や国際市場でどのような影響を与えるかという視点が必要である。定期的にミーティングに参加し、情報交換を絶やさないことが重要である。

<課題>

課題①
電子署名に関する技術について、日本として、全体をまとめる組織がなく、各専門家が行き来しなければならないため、組織立ってやれていない。

課題②
ETSI を策定しているメンバに直接参加するのは、現状では難しい。日本からの問題提起や細かな修正は受け入れるが、例えば、規格の構造や仕組みそのものを変えるような提案は作業班に直接参加して議論する必要があり、ハードルが高い。一方で、日本から問題点がわかった段階でのみのインプットとなっている現状の関係を、今後常にインプットできるような協力関係に変えていけば、策定メンバに入る余地があると思われる。

6. ARIB

ARIB について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
定期的な見直しは特に定められておらず、必要に応じて規格が制定・改訂される。

● どのようなタイミングで技術を提案できるか
新たな放送方式が審議されるタイミングで標準規格の策定および改定が行われる。

● リエゾン関係がある団体
次の委員会とリエゾン関係にある。 **・ITU、ISO/IEC、IETF 他**

● 技術提案の手続き等
委員会のメンバであれば、暗号アルゴリズムを提案できる人に制限は特にない。

● 会合が年何回あるか
年間回数等は特に決まっておらず、随時会合が行われている。

● 投票権を取る条件
ARIB への入会が条件となる。

● 投票権は誰に帰属しているか
投票権は、開発部会に登録された委員に帰属する。

● 平均的に標準ができるまでどのくらいの期間が必要か
通常は数年程度である。

● 規格化のプロセス
作業班および開発部会での審議の後に規格会議で決議される。通常、作業班下に検討グループ（TG: タスクグループ）が設置され、委員から提案された方式等について審議を行いながら、提案者等を中心にドラフティング作業を進める。その後、作業班、主任会議および開発部会での審議に諮られ、規格会議の決議を経て策定される。TG を設けず、作業班が直接ドラフティング作業を行う場合もある。

● 会議の開催場所
通常、霞が関 日土地ビル内の ARIB 会議室で開催される。

● 提案者に最低限必要となる作業
※「規格化のプロセス」の項目を参照のこと

● 国内におけるコンタクト先
一般社団法人 電波産業会 （ARIB） 研究開発本部 放送グループ 〒100-0013 東京都千代田区霞が関 1-4-1 日土地ビル TEL：(03)5510-8597 FAX：(03)3592-1103 http://www.arib.or.jp/

7. ISA 100

ISA100 について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
定期的な見直しは特に定められておらず、必要に応じて規格が制定・改訂される。

● どのようなタイミングで技術を提案できるか
規格の核となる無線通信技術は既に標準化されているため、新たに標準化するのは難しい。

● リエゾン関係がある団体
次の委員会とリエゾン関係にある。 **・IETF, IEEE 等**

● 技術提案の手続き等
メンバであれば誰でも提案可能である。

● 会合が年何回あるか
年間 1 回の会合に加え、WG によっては追加で 2 回程度の会合が行われる。

● 投票権を取る条件
参加メンバ単位、参加組織単位、ボード単位の投票に分かれている。参加メンバ単位の投票であれば、特に条件はない。

● 投票権は誰に帰属しているか
参加メンバ単位、参加組織単位、ボード単位の投票に分かれているため、それぞれの投票において投票権は参加メンバ、参加組織、ボードのそれぞれに帰属する。

● 平均的に標準ができるまでどのくらいの期間が必要か
3 年以上必要とする場合もある。

● 電話会議の時間帯と時間帯の決め方
WG によって異なるが、概ね EU と米国の時間に合わせる。

● 会議の開催場所
EU、アジア、米国で開催されることとなっているが、米国での開催が多い。

● 規格策定のため利用するツール
Microsoft Office Word

● 国内におけるコンタクト先
特にないため、既にメンバである国内企業にコンタクトするのがよいと思われる。

<ノウハウ>

ノウハウ①
技術の需要側と供給側のメンバの数が大体同じになるように調整されている。そのため、供給側と需要側が比較的結びついている。また、供給側も暗黙のアライアンスを組んでおり二極化している。したがって、技術を導入するためには、そのどちらかとうまく調整を行う必要がある。

ノウハウ②
ISA で承認されると ANSI で承認されやすくなる。そこから IEC での標準化も可能となる。

8. IETF

IETF について、<参考情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<参考情報>

● 何年おきに規格の改定があるのか
定期的な見直しは特に定められておらず、必要に応じて規格が制定・改訂される。

● どのようなタイミングで技術を提案できるか
メーリングリスト（以下、ML という）にて、随時提案することができる。

● リエゾン関係がある団体
下記の委員会等とリエゾン関係がある。 ・ITU-T、ETSI、IEEE、ISO/IEC 他 参照： https://datatracker.ietf.org/liaison/

● 技術提案の手続き等
提案する人に制限等はない。但し IETF への参加、すなわち ML への登録などが必要となる。また RFC5378、RFC3979 といった IETF における活動に関する同意事項がある。

● 会合が年何回あるか
会合は年 3 回開催されている。WG によっては、Interim(中間)ミーティングが IETF ミーティングの前後などに行われることがある。また、Interim ミーティングは電話で行われることがある。

● 投票権を取る条件
IETF では投票権の概念がない。これは voting を避ける理念があるためである。WG では、会場でのハミングの大きさ（を利用した意思表示）や議論の結果（反対意見がないなど）を受けて合意形成される。チェアのハンドリングに依存する部分もあり、例えばハミングが WG 会場で大きくても、重要な意見によってコンセンサスがくつがえることもある。

● 投票権は誰に帰属しているか
投票権という概念はなく、参加者全員がハミングや ML で意見を出すことができる。

● 平均的に標準ができるまでどのくらいの期間が必要か
通常は 2,3 年程度である。ただし、異例的に短く 2,3 ヶ月で RFC 化するものもあれば、広く使われているにも関わらず、RFC になっていないものもある。

● 電話会議の時間帯と時間帯の決め方
電話会議の時間帯はチェアと、活発な参加者や発表者に合わせられる。アナウンス時には既に決まっていることが多く、チェアが活発な参加者と事前に調整していると考えられる。なおチェアや主要な参加者の居住地は米国以外である場合も多く、ケースによると考えられる。

● 規格化のプロセス
PS(Proposed Standard), DS(Draft Standard)を経て、Standard となる。実質 DS で国際標準と認識されることが多い。 RFC(PS,DS,STD)になる前のプロセスについて以下に示す（参考：Tao of IETF)。詳細は RFC2026 参照のこと。 1. ドキュメントをインターネットドラフトとして公開する 2. ドラフトのコメントを受け取る 3. コメントに応じてドラフトを編集する 4. この 1 から 3 のステップを数回繰り返す 5. IESG にドラフトを提出するようにエリアディレクターに依頼(個人に属するものであれば) ドラフトが公式のワーキンググループの製品ならば、WG 議長が AD に IESG へ提出するように依頼する。 6. エリアディレクターが提案を受け付けると、最初の審査を行い、大抵の場合何らかのアップデートを求めてくる。 7. 幅広く IETF メンバから意見を集める。エリアによっては審査チームがあり、IESG へすぐに提出可能かどうかドラフトをチェックします。 特に、Security Directorate(SecDir)と General Area Review Team(Gen-Art)の 2 つの審査チームは活発に審査を行っています。 これらすべての審査によって、最終的な RFC の品質を改良していくことができるのです。 8. IESG メンバと懸念事項を議論する。その懸念事項が簡単な確認で解消されるかもしれませんし、ドキュメントへ追加や変更が必要となるかもしれません。 9. ドキュメントが RFC エディターによって公開されるのを待つ。

● 会議の開催場所
会議の開催場所は、アメリカ、ヨーロッパ、アジアの順に多く開催されてきた。アメリカやヨーロッパからの参加者が多いため、参加者数を確保しやすい場所で行われる傾向があると思われる。詳細がドキュメント化されたことがある(下記)。 参照：IETF Meeting Venue Selection Criteria https://www.ietf.org/archive/id/draft-palet-ietf-meeting-venue-selection-criteria-04.txt

● 提案者に最低限必要となる作業
最低限必要になる作業は、IETF のプロセスにおけるドキュメント作成とコメントを受けた修正、チェアや共著者との調整、ミーティングでのプレゼンテーションなどである。会合への参加も重要である。

● 規格策定に利用するツール
ツールは基本的にテキストエディタである。XML から RFC 形式への変換のほか、Word や PDF からの変換といった Web 上のツールも利用可能になってきている。 参照：IETF Tools http://tools.ietf.org/

● 標準化動向について
SSL/TLS の次のバージョン(バージョン 1.3)が議論されており、採用される暗号アルゴリズムの議論が今後活発化する可能性がある。IPsec はプロトコルの維持や更新の段階と捉えられる状態が続いており、例えば新たな暗号アルゴリズムへの対応といった提案ができる状態にある。X.509v3 の電子証明書や電子署名にも使われる CMS といった基本的な仕様も RFC 化の後に落ち着いた状態となっている。(PKCS#11 等の外部の仕様をプロトコルで扱いやすいように RFC 化する活動があるなど、一見目立たなくても重要な位置づけの活動は存在する。) いくつかの国や大企業による大規模な通信傍受が話題になって以降、ワークショップなどが開催され、「Encryption by default」すなわちプロトコル策定にあたっては暗号化を考慮する旨の考え方がステートメントとして出されている。 SSL/TLS 以外にも、採用される暗号アルゴリズムの議論が行われている。

● 国内におけるコンタクト先
IETF については日本支部のような国内組織はなく、IETF への参加者などが集まって適宜情報交換が行われている。 ISOC 日本支部と JPNIC の主催により、IETF 参加者による IETF 報告会が行われている。 - ISOC-JP http://www.isoc.jp/

● その他
IETF の組織構造の変化や、スノーデン事件を受けた情勢を受けた対応の変化 （様々なプロトコルにおいて暗号機能の実装が推奨されるなど）があり、 変わりつつある。 古い文献ではあるが、IETF そのものに関してまとまった資料を以下に示す。 - Tao of IETF http://www.ietf.org/tao.html - IETF のタオ：初心者のためのインターネット技術タスクフォースガイド (Tao of IETF 和訳) http://www.ietf.org/tao-translated-ja.html - IETF と RFC, JPNIC https://www.nic.ad.jp/ja/tech/rfc-jp.html - JPNIC RFC-JP(Introduction to RFCs) http://rfc-jp.nic.ad.jp/introduction/WhatisRFC.html

<ノウハウ>

ノウハウ①
IETF では、キーパーソンが納得していなければ、提案が進まない場合があるため、キーパーソンを把握し納得してもらうことが大切である。キーパーソンの納得のためには、ML を含めて議論に継続的に参加していること(プレゼンス)と、WG やキーパーソンへのアピール(個別のアプローチ)といった方策が考えられる。

ノウハウ②
IETF では細かく検討された仕様よりも「複数のプログラムにおいて動作する実装が存在している」「広く使われる状況」「オープンソースのように改良しやすい」といった状況が RFC 化までに受け入れやすいと考えられる。ML などで WG へのプレゼンスも重要である。 なお、近年、国などによる大規模な盗聴が問題視されており、例えば国によって策定された暗号アルゴリズムにはバックドアがある可能性が疑われるような見方が広がっている。その場合、個人や私企業の技術提案が有利と見られる。

以上

発行日　2015 年 6 月 30 日　第 1 版
発行者
・〒113-6591
東京都文京区本駒込二丁目 28 番 8 号
独立行政法人 情報処理推進機構
（技術本部　セキュリティセンター　暗号グループ）
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
2-28-8 HONKOMAGOME, BUNKYO-KU
TOKYO, 113-6591 JAPAN

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