2014 年度　第 2 回暗号技術検討会

日時：平成 27 年 3 月 27 日(金) 10:00～12:00
場所：経済産業省別館 1 階 104 各省共用会議室

議　事　次　第

１．開　会

２．議　事
（1） 2014 年度　暗号技術評価委員会活動報告について【承認事項】
（2） 2014 年度　暗号技術活用委員会活動報告について【承認事項】
（3） CRYPTREC 暗号リストの注釈の一部変更について【承認事項】
（4） 2014 年度　暗号技術検討会報告書（案）について【承認事項】
（5） 暗号技術検討会における小グループの設置について【承認事項】
（6） 2015 年度　暗号技術評価委員会活動計画（案）について【承認事項】
（7） 2015 年度　暗号技術活用委員会の活動について【承認事項】

３．閉　会

（資料番号）	（資　料　名）
資料１	2014 年度 暗号技術評価委員会活動報告
資料１別添１	2014 年度 暗号技術調査 WG（暗号解析評価）活動報告
資料１別添２	離散対数問題の困難性に関する調査
資料１別添３	格子問題等の困難性に関する調査
資料１別添４	2014 年度 暗号技術調査 WG（軽量暗号）活動報告
資料１別添５	暗号技術調査 WG（軽量暗号）報告書（案）
資料２	2014 年度 暗号技術活用委員会活動報告
資料２別添１	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
資料２別添２	SSL/TLS 暗号設定ガイドライン
資料２別添３	SSL/TLS 暗号設定ガイドラインチェックリスト
資料２別添４－１	暗号技術参照関係の俯瞰図（全体像）
資料２別添４－２	暗号技術参照関係の俯瞰図
資料２別添５	標準化提案におけるノウハウ・課題・基本的な情報の整理
資料３	CRYPTREC 暗号リストの注釈の一部変更について
資料３別添	CRYPTREC 暗号リストの変更案
資料４	2014 年度 暗号技術検討会報告書（案）
資料５	暗号技術検討会における小グループの設置について（案）
資料６	2015 年度 暗号技術評価委員会活動計画（案）
資料７	2015 年度 暗号技術活用委員会の活動について（案）
参考資料１	2014 年度 第 1 回暗号技術検討会議事概要（案）
参考資料２	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
参考資料３	2014 年度　暗号技術検討会　構成員・オブザーバ名簿

資料 1

2014 年度暗号技術評価委員会 活動報告

1. 活動目的

暗号技術評価委員会では、CRYPTREC 暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。

(1) 暗号技術の安全性及び実装に係る監視及び評価
(2) 新世代暗号に係る調査
(3) 暗号技術の安全な利用方法に関する調査

2. 活動概要

(1) 暗号技術の安全性及び実装に係る監視及び評価

① CRYPTREC 暗号等の監視

- 国際会議等で発表された CRYPTREC 暗号リスト等の安全性及び実装に係る技術（暗号モジュールに対する攻撃とその対策も含む）に関する監視活動を行った。
- CRYPTREC 暗号リストに掲載された暗号技術（ECDSA、ECDH）の仕様書の参照先の変更について検討を行った。 検討対象である SECG SEC1 Ver 2.0 において、「軽微な修正」の範囲を超える部分があることが認められたことから、当該変更については、安全性・実装性のみならず、製品化・利用実績・知財権のほか、実装の適合性評価にも影響が及ぶため、引き続き検討することとなった。

② 電子政府推奨暗号リスト及び推奨候補暗号リストからの運用監視暗号リストへの降格 及び 運用監視暗号リストからの危殆化が進んだ暗号の削除

- CRYPTREC 暗号リストの安全性に係る継続的な監視活動とともに、リストからの降格や削除、注釈の改訂が必要か検討を行った。
- 128-bit key RC4 は、現在、運用監視暗号リストに掲載され、「128-bit RC4 は、SSL(TLS1.0 以上)に限定して利用すること」という注釈が付与されているが、近年の攻撃の進化により、SSL/TLS での利用についても安全性に対する懸念が高まったことから注釈の改定について、昨年度検討を開始し、本年度も引き続き検討を行った。暗号技術活用委員会の協力も得て、至った結論として、早期に RC4 からの移行が進むことが好ましく、「今後は極力利用すべきでない」という注釈変更の意図を明確化するために、以下のように RC4 の注釈を変更することとなった。

「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。」

③ CRYPTREC 注意喚起レポートの発行

監視活動の一環で監視活動報告を毎委員会に行っており、2014 年度は、その活動報告の域を超え注意喚起レポートを必要とする解析結果等に該当する技術分類はなかった。

④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加

SHA-3 の標準化動向などを鑑み、2014 年度は下記の通り、ハッシュ関数 SHA-224、SHA-512/224、SHA-512/256、SHA-3(仕様が確定されていないため NIST Draft FIPS 202 を参照）の安全性について外部評価を実施した。

- 外部評価者：
  - Donghoon Chang 氏、Itai Dinur 氏、Florian Mendel氏
- 評価結果
  - 外部評価では、近年のハッシュ関数解析技術の進展により解析が進んでいることが示され、また新規の解析結果も報告された。外部評価レポートで報告された解析結果からは、対象のハッシュ関数の安全性には十分なマージンがあり、現実的な脅威の観点から大きな問題点は見つかっていない。以上のことから、「ハッシュ関数 SHA-512/256、SHA512/224、SHA-224、SHA-3 の安全性には現時点では現実的な脅威につながる問題はない」という評価結果とした。

⑤ 既存の技術分類の修正を伴わない新技術分類の追加

現時点で該当する技術分類はない。

(2) 新世代暗号に係る調査

① 暗号技術調査ワーキンググループ(暗号解析評価)

- 格子問題等の困難性に関する調査

昨年度に引き続き、量子計算機が実現しても安全性が保たれると期待されている「耐量子計算機暗号」を支える数学的問題の困難性に関する調査を行った。

今年度は、昨年度調査を行った数学的問題の困難性を利用した公開鍵暗号技術とパラメータ選択に関する調査を行った。

・ 離散対数問題の困難性に関する調査

近年、研究の進展している有限体上あるいは楕円曲線上の離散対数問題の困難性に関する調査を行った。

具体的には、大きな標数の素体上構成される DSA 及び DH への安全性に影響はないことの再確認を行い、また、2008・2009 年度に作成した ID ベース暗号に関する調査報告書に関し、利用する有限体に関する注意喚起の方法について検討を行った。

② 暗号技術調査ワーキンググループ(軽量暗号)

・ 2013 年度は、これまでに提案されている軽量暗号の現状調査、アプリケーションに関する調査、実装評価等を行った。2014 年度はこれらをふまえてさらに検討を行い、CRYPTREC における今後の活動方針を検討し、暗号技術評価委員会に提言を行った。

・ 今後の活動方針に対する提言は以下の通り。

- 軽量暗号は、特定の性能指標における優位性が認められ、次世代のネットワークサービスでの活用が期待される一方、電子政府推奨暗号リスト掲載の暗号技術ほど高い安全性を保証していない方式もあり、利用において留意すべき点がある。
- 軽量暗号を選択・利用する際の技術的判断に資することや今後の利用促進をはかることを目的として暗号技術ガイドラインを発行することが有効と考えられる。
- 暗号技術ガイドラインの発行（(A) 暗号技術ガイドライン(軽量暗号の最新動向)、または、(B) 暗号技術ガイドライン(軽量暗号の詳細評価)）について、軽量暗号全体となると膨大であり、また技術分類によって状況が異なる。詳細評価が望ましい分野と、現時点では既存文献のサーベイでよいと思われる分野がある。よって、(A)と(B)のハイブリッド案で軽量暗号に関する暗号技術ガイドラインを作成するのがよいと思われる。
- 詳細評価を行う技術分類は、新規評価の必要性（既存文献で十分な評価結果が得られるかどうか）、当該技術分野における我が国の技術の将来性、当該技術分野の現時点での注目度・重要度、評価結果から期待される学術的貢献等を鑑みて決定するのがよいと考えられる。
- 軽量暗号は、現時点では直ちに電子政府システムで活用される段階ではないと考えるが、今後関連する次世代ネットワークサービスに搭載される可能性があることから、上記の活動は長期的には電子政府システムの安全性向上にも資すると期待される。

(3) 暗号技術の安全な利用方法に関する調査

① 「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)」の更新

2014年10月14日、SSL3.0におけるCBCモードに対して、新たに POODLE攻撃が公表されたため、2013年度に発行した「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）」に POODLE 攻撃に関する記載を加え、ガイドラインの更新を行った。

以上

資料 1-別添 1

2014 年度暗号技術調査 WG（暗号解析評価）活動報告

1　活動目的

公開鍵暗号の安全性は、素因数分解の困難性や離散対数問題の困難性などさまざまな数学的問題に依存している。本ワーキンググループでは、格子問題のほか、NP 困難に係る問題、多変数多項式に係る問題、符号理論に係る問題等、量子計算機が実現しても安全性が保たれると期待されている「耐量子計算機暗号」を支える数学的問題の困難性に関する調査を行う。

2　委員構成

主査：高木 剛(九州大学)
委員：青木 和麻呂(NTT)
委員：太田 和夫(電気通信大学)
委員：草川 恵太 (NTT)
委員：國廣 昇(東京大学)
委員：下山 武司(富士通研究所)
委員：安田 雅哉(富士通研究所)

3　活動方針

3.1 格子問題等の困難性に関する調査

2013 年度は、数学的問題の困難性のうち、

(i)　Shortest Vector Problem (SVP)
(ii)　Learning with Errors (LWE)
(iii)　Learning Parity with Noise (LPN)
(iv)　Approximate Common Divisor (ACD)

の 4 つを選んで調査を行った。2014 年度も引き続き、これらを利用した公開鍵暗号技術とパラメータ選択に関する調査を行う。

3.2 離散対数問題の困難性に関する調査

CRYPTREC において公表した「ID ベース暗号に関する調査報告書」(2008 年度)及び「2009 年度版リストガイト」において、近年の攻撃により脆弱となったパラメータを指摘し、報告書（の一部）を改訂する。

4　活動概要

4.1 スケジュール

第 1 回　2014 年 9 月 22 日　　活動計画案や作業内容についての審議と了承
第 2 回　2015 年 2 月 17 日　　調査内容についての審議と了承

4.2 格子問題等の困難性の調査について

主に、上述の数学的問題の困難性(i)～(iv)を利用した公開鍵暗号技術の例とパラメータ選択に関する記述を補った。

4.3 離散対数問題の困難性の調査について

大きな標数の素体上構成される DSA 及び DH への安全性に影響はないことの再確認を行った。また、過去の ID ベース暗号に関する調査報告書における、小さな標数の離散対数問題の困難性を利用する際の注意喚起の方法について検討を行った。

4.4 予測図の更新

スーパーコンピュータのベンチマーク結果の 1 位から 500 位を 1993 年から半年毎に集計している WebサイトTOP500.Org[1]において、2014 年 6 月・11 月のベンチマーク結果が追加されたので、素因数分解問題及び楕円曲線上の離散対数問題に関する２つの予測図を更新した。

5　成果概要

5.1 格子問題等の困難性の調査について

下記の(a)～(d)の更新部分について検討した。

(a)　第２章　総論
　① 2.1.3 計算機実験
　　最新の実験結果を追加した。

(b)　第 3 章　Learning with error (LWE)問題
　① 3.1.3 節の追加
　　代表的な暗号方式として、Regev による方式および somewhat 準同型暗号方式の概略を記載した。
　② 3.2.2 節の最後に「■近年の攻撃研究の動向」の追加
　　2014 年に ACISP で発表された Binary-LWE 問題に対する攻撃手法の概略について記載した。

[1] http://www.top500.org/

(c)　第4章　Learning parity with noise (LPN)問題

①4.2節に、代表的な暗号方式を追加（旧4.3-4.5節から移動した文書有り）
代表的な暗号方式として、Alekhnovich暗号およびMcEliece暗号の概略を記載した。

②4.3節(旧4.2節)に、いくつかコメントを追加

(d)　第5章　Approximate Common Divisor (ACD)問題

①5.1.3節において、5.1.3.1節及び5.1.3.2節の追加。
ACD問題のアプリケーションとして、van Dijkらおよび Cheonらの somewhat準同型暗号方式の概略を記載した。

②5.1.4節の追加

③5.5節の追加
2014年に、Cheonらが導入したco-ACD問題についての概略を記載した。

表１：2013-2014年度の調査内容と執筆担当

章	執筆担当	内容
1章	事務局	調査の目的、まとめ（非専門家向け）
2章 総論	石黒 司 委員(2013年度)	総論（Generalな攻撃に関する総論）：SVP、LLL、BKZ
3章 LWE	下山 武司 委員、 安田 雅哉 委員	各問題について以下の項目を記述 (1) 公開鍵方式からの帰着、証明の有無、追加の問題・制約など (2) 攻撃や量子アルゴリズム - Generalな攻撃との関係 - 固有の攻撃 - 量子アルゴリズムとの関係
4章 LPN	草川 恵太 委員	
5章 ACD	國廣 昇 委員	

5.2 離散対数問題の困難性の調査について

(a)　過去のIDベース暗号に関する調査報告書(2008年度及び2009年度)に、図1の通り、注意喚起のための文言を挿入する。

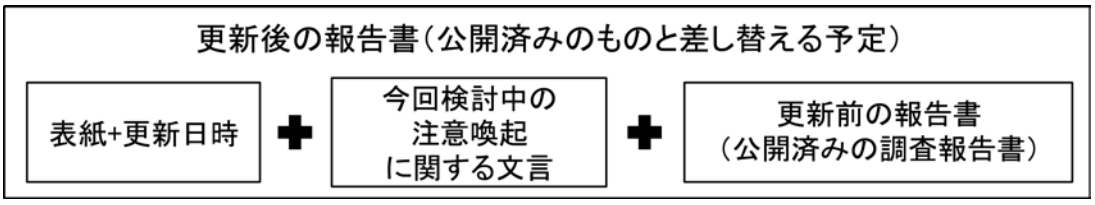


図1: 更新のイメージ

(b)　注意喚起の文案(暗号解析評価 WG 了承)は下記の通りである。

ペアリング暗号の安全性は、楕円曲線上の離散対数問題と有限体上の離散対数問題を解く計算の困難性を基盤としている。有限体上の離散対数問題を効率よく解く手法には数体篩法と関数体篩法があり、前者は標数が大きな有限体に、後者は標数の小さな有限体の場合に利用できる。

標数の大きな有限体上の離散対数問題に適した数体篩法の改良も進んではいるものの、関数体篩法ほどの計算量の改善は現在まで報告されていない。従って、素体上構成されている DSA 及び DH への安全性に影響はない。

近年、関数体篩法において、ペアリング暗号に適した標数の小さいある種のタイプの有限体に対して有効な手法が提案され、計算量が大きく削減された。たとえば、「ID ベース暗号に関する調査報告書(平成 21 年 3 月)」の第 3 章の表内に掲載されているペアリング実装例のうち、表 3.2 の標数 2 において、埋め込み次数 4 で拡大次数 313 以下や、表 3.3 の標数 3 において、埋め込み次数 6 で拡大次数 127 以下は、セキュリティレベルが 80 ビット以下であると見積もられる。

関数体篩法や数体篩法の計算量は、有限体の位数以外に、拡大次数と部分体の位数の比などが関係するため、ペアリング暗号の安全性は利用する有限体ごとに評価する必要がある。詳しくは、「CRYPTREC Report 2014 暗号技術評価報告書 付録 4」を参照のこと。

5.3 予測図の更新

「1 年間でふるい処理を完了するのに要求される処理能力の予測」の更新後の図は、図 2 の通りとなる。

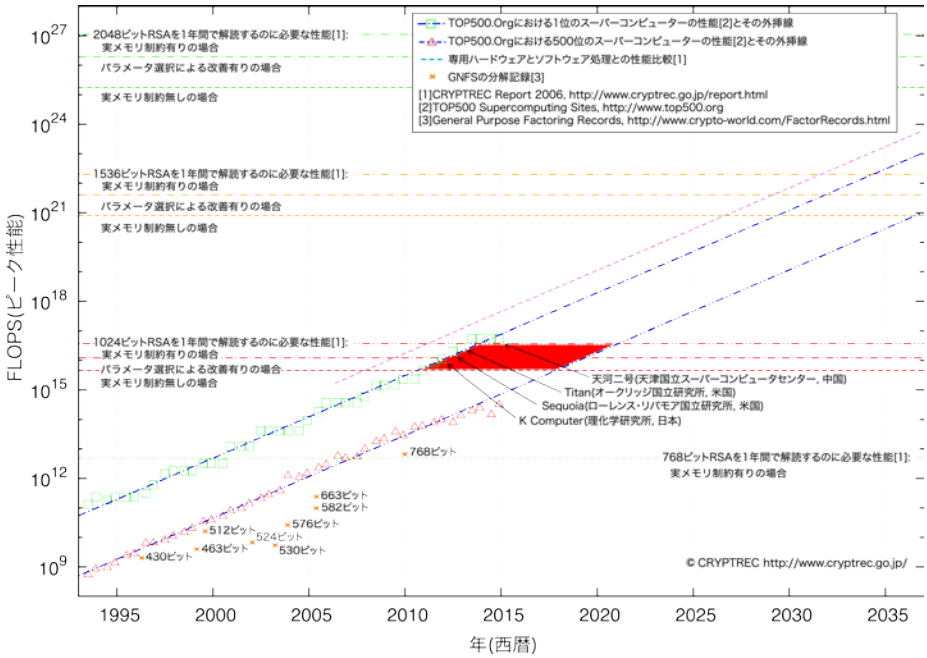


図 2：1 年間でふるい処理を完了するのに要求される処理能力の予測(2015 年 2 月更新)

また、「ρ 法で ECDLP を 1 年で解くのに要求される処理能力の予測」の更新後の図は、図 3 の通りとなる。

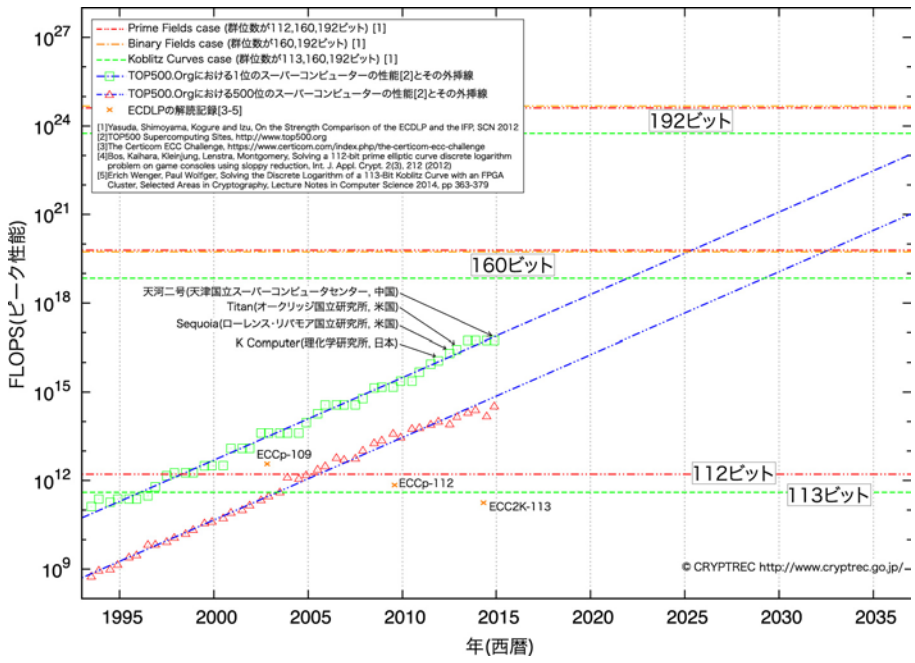


図 3： ρ 法で ECDLP を 1 年で解くのに要求される処理能力の予測(2015 年 2 月)

以上

離散対数問題の困難性に関する調査
関数体篩法の近年の改良とその影響について

2014 年 2 月作成 (2015 年 2 月更新)

概 要

ペアリング暗号の安全性は, 楕円曲線上の離散対数問題と有限体上の離散対数問題を解く計算の困難性を基盤としている. 即ちそれらの離散対数問題の内で一つでも解くことができればペアリング暗号は解読されてしまう. 有限体上の離散対数問題を効率よく解く手法として数体篩法と関数体篩法が挙げられ, 前者は標数が大きい有限体に, 後者は標数の小さい有限体の場合に適している. 近年, 関数体篩法の改良で大きな進展があった. これまで関数体篩法の関係探索段階において, sieving (篩) と呼ばれる手法が採用されてきたが, 近年は pinpointing に代表される新たな手法 (Frobenius representation algorithm など) が提案され, さらに Kummer extension の性質などが利用できる有限体では計算量が大きく削減される. 一方で, 標数の大きい有限体上の離散対数問題に適した数体篩法の改良も進んではいるものの, 関数体篩法ほどの計算量の改善は現在まで報告されていない.

本稿では, ペアリング暗号に適した標数の小さい有限体上の離散対数問題において, 上記の新たな手法を導入した関数体篩法に関する近年の研究報告について簡単に説明する. 特に重要な事実として, Kummer extension などの性質の利用が有効でない, ペアリング暗号で利用される標数の小さな有限体上の離散対数問題に対しても, 新たな手法の有効性を示す研究報告を挙げる.

最後に, 関数体篩法や数体篩法を有効に適用するには, 拡大次数の大きさと部分体の大きさの比などが関係するため, ペアリング暗号の安全性は推奨された有限体ごとに評価される必要があることを注意として挙げる.

1 概説

有限体上の離散対数問題を解く計算の困難性はペアリング暗号の安全性の基盤となっており, 有限体はペアリング暗号の安全性を決定する重要な暗号パラメータとみなされる. さらに, 有限体はペアリング暗号の暗号処理速度にも影響を及ぼすため, 安全性と実用性の双方を考慮して有限体の設定を行う必要がある.

標数が大きい有限体上の離散対数問題を解くことに適したアルゴリズムとして数体篩法が知られており, 同様に標数が小さい場合については関数体篩法が適していることが知られている. 特に標数が小さい場合については下記の三種の有限体に関連付けられるペアリング暗号の研究が盛んに行われている: (i) 標数が 3 で拡大次数が 6ℓ (ℓ は素数, 以下同様) の有限体 $GF(3^{6\ell})$, (ii) 標数が 2 で拡大次数が 4ℓ の有限体 $GF(2^{4\ell})$, (iii) 標数が 2 で拡大次数が 12ℓ の有限体 $GF(2^{12\ell})$. これらの有限体を使用するペアリング暗号の安全性を評価するために, 各々の有限体に適した関数体篩法の研究が様々な組織によって行われている.

関数体篩法では関係探索段階において, sieving (篩) によって relation と呼ばれる, モニックで既約な次数の小さい多項式 (因子基底) の積で表される多項式を生成し収集する. この relation から各因子基底の離散対数を解とする線型方程式が得られ, この後の線型代数段階でその線型方程式を解く. 後述の新しい手法である pinpointing の戦略に沿った手法が登場するまではこの二つの段階の計算量が関数体篩法の計算量を決定していた [1]. Pinpointing は sieving に代わる手法として Joux によって 2012 年に提案された [14].

[1] 関数体篩法の一種で, 漸近的な計算量が quasi-polynomial である Frobenius Representation algorithm [5] の計算量は, 関係探索段階や線型代数段階ではなく, 与えられた離散対数問題を解く段階である個別離散対数計算段階 (Individual Logarithm Phase) の計算量で見積もられる. しかし, Joux と Pierrot らの ASIACRYPT 2014 のスライドに書かれているように, Frobenius Representation algorithm においても, 実際の計算では線型代数段階の計算コストが最も大きい場合が多い.

Sieving では, 篩区間に対応する relation の候補である各多項式に対して, ある因子基底を因子として持つものを, その因子基底による割り算をほとんどすることなく, マーキングのみを行うことで収集していた. 即ち sieving の利点は, 多項式の割り算をマーキングで代用することで, relatoin の候補となる各多項式に対する計算コストを削減することである. しかし, 候補となる多項式の数は膨大である. Pinpointing では, 小さな次数の既約多項式の積で表される多項式を探し, その多項式から複数の同様な異なる多項式を大量に生成する. Pinpointing の狙いは, 一つの relation を得るために必要な候補の多項式の個数を少なくすることである. 有限体 $GF(q^n)$ 上の離散対数問題を解く場合に, $Q := q^n$ と書くことにして, 関数体篩法の計算量を表すために次の関数を用意する:

$$L_Q(\alpha, c) := \exp((c + o(1))(\log Q)^{\alpha}(\log\log Q)^{1-\alpha}),$$

但し $0 < \alpha < 1$ で $c > 0$ とする.

以下で, 関数体篩法の計算量が改善される, 近年までの経緯について簡単に紹介する. (この部分については Adj, Menezes, Oliveira, Henríquez らの原稿に詳しく書かれている [2].) 2006 年に Joux と Lercier によって提案された関数体篩法の計算量は,

$$q = L_Q(1/3, 3^{-2/3}),\ n = 3^{2/3}(\log Q/\log\log Q)^{2/3}$$

の場合に $L_Q(1/3, 1.44)$ であったが, この関数体篩法に pinpointing を適用することにより, その計算量は $L_Q(1/3, 0.96)$ に削減された. その結果 Joux は 1425-bit の有限体 $GF(p^{57})$ ($p = 33341353$ とする) 上の離散対数問題を解くことに成功した.

2013 年, Joux は pinpointing の方針に沿って改良された手法を導入することによって,

$$q \approx n/2$$

の場合に $L_Q(1/4 + o(1), c)$ となるアルゴリズムを提案し, 6168-bit の有限体 $GF(2^{8\cdot 3\cdot 257})$ 上の離散対数問題を解いた [16]. さらに, 同年, Barbulescu, Gaudry, Joux と Thomé は [16] の最後の計算段階を改良することによって

$$q \approx n,\ n \leq q + 2$$

の場合に有限体 $GF(q^{2n}) = GF(Q)$ 上の離散対数問題を解く計算量を quasi-polynomial time

$$(\log Q)^{O(\log\log Q)}$$

に改良することに成功した [5]. 注意すべきはこの計算量が, 任意の $0 < \alpha < 1$ と $c > 0$ に対して, $L_Q(\alpha, c)$ より漸近的に小さいことである. これら [5, 16] の種の手法は Frobenius representation algorithm と呼ばれる [22].

最後に, 上述のように関数体篩法の計算量は適用する有限体の大きさだけではなく, 部分体の大きさと拡大次数の大きさの比などの影響も受ける. 従って, ペアリング暗号の安全性は, 推奨された暗号パラメータごとに評価される必要がある.

2 小さい標数の有限体を使用するペアリング暗号への影響

この節では, 小さい標数の有限体を使用するペアリング暗号への Frobenius representation algorithm が与える影響に関する研究成果について紹介する. 結論としては, 数値実験の報告においても理論値によるその安全性評価の報告においても, 小さい標数の有限体を使用するペアリング暗号の安全性がそれ以前の見積もりより低くなることを意味する結果が報告されている.

表 1: 標数が 2 または 3 である有限体上の離散対数問題に関する記録. 表中の * は Kummer extension または twisted Kummer extension の性質を適用されたことを意味する.

Date	Field	Bitsize	CPU-hours	Algorithm	Authors	Reference
1992	$GF(2^{401})$	401	114000	[6]	Gordon, McCurley	[11]
2001.09	$GF(2^{521})$	521	2000	[19]	Joux, Lercier	[19]
2001	$GF(2^{607})$	607	> 200000	[6]	Thomé	[24]
2005.09	$GF(2^{613})$	613	26000	[19]	Joux, Lercier	[21]
2012.06	$GF(3^{6\cdot 97})$	923	895000	[20]	Hayashi et al.	[13]
2013.02	$GF(2^{2\cdot 7\cdot 127})$	1778*	220	[16]	Joux	[15]
2013.02	$GF(2^{3^3\cdot 73})$	1971*	3132	[7]	Göloğlu et al.	[7]
2013.03	$GF(2^{2^4\cdot 3\cdot 5\cdot 17})$	4080*	14100	[16]	Joux	[17]
2013.04	$GF(2^{809})$	809	19300	[1, 20]	The Caramel Group	[4]
2013.04	$GF(2^{2^3\cdot 3^2\cdot 5\cdot 17})$	6120*	750	[7, 16]	Göloğlu et al.	[8]
2013.05	$GF(2^{2^3\cdot 3\cdot 257})$	6168*	550	[16]	Joux	[18]
2014.01	$GF(3^{6\cdot 137})$	1303	888	[16]	Adj et al.	[3]
2014.01	$GF(2^{2\cdot 3^5\cdot 19})$	9234*	398000	[16]	Granger et al.	[9]
2014.01	$GF(2^{2^2\cdot 3\cdot 367})$	4404	52000	[16]	Granger et al.	[10]
2014.09	$GF(3^{5\cdot 479})$	3796	8600	[16]	Joux, Pierrot	[22]
2014	$GF(3^{6\cdot 163})$	1551	1201	[16]	Adj et al.	[3]
2014.10	$GF(2^{1279})$	1279	35040	[16]	Kleinjung	[23]

まず数値実験に関してであるが, 表 1 は標数が 2 または 3 である有限体上の離散対数問題に関する主な記録をまとめたものである [2]. 表 1 が示すように, Frobenius representation algorithm ([7, 16]) において Kummer extension または twisted Kummer extension の性質などを適用できる場合は, 9234-bit 長の離散対数問題の記録のように, 大きな bit 長の離散対数問題が解かれている. それに比べて素数次拡大の場合の最高記録は 1279-bit 長の離散対数問題となっている. ペアリング暗号で利用される (i) $GF(3^{6\ell})$ (ℓ は素数とする) に分類される有限体については, 素数次拡大の有限体に次いで計算コストの高い有限体に分類でき, $GF(3^{6\cdot 137})$ や $GF(3^{6\cdot 163})$ の場合が解かれている. 従って $\ell \leq 163$ である有限体 $GF(3^{6\ell})$ 上の離散対数問題が現実的な時間内で解かれることが見込まれる. また, (iii) $GF(2^{12\ell})$ (ℓ は素数とする) の場合については, 128-bit 安全性が見込まれていた有限体 $GF(2^{12\cdot 367})$ の場合が解かれている. 従って, その部分体である $GF(2^{4\cdot 367})$ 上の離散対数問題も解くことが可能であるため, $\ell \leq 367$ である有限体 $GF(2^{12\ell})$ と 有限体 $GF(2^{4\ell})$ 上の離散対数問題は現実的な時間内で解かれることが見込まれる.

理論的な安全性評価については, Adj, Menezes, Oliveira, Henríquez らは, Frobenius representation algorithm ([5]) を用いた場合, 特に 128-bit 安全性が見込まれていた有限体 $GF(3^{6\cdot 509})$ の場合は 73.7-bit 安全性と見積もっている [2]. また, Granger, Kleinjung, Zumbrägel らは体の表現を工夫することにより, 同じく 128-bit 安全性が見込まれていた有限体 $GF(2^{4\cdot 1223})$ を使用した場合は 59-bit 安全性と見積もっている [10].

[2] 表 1 は, Joux らがまとめた離散対数問題に関するサーベイ集 “The Past, evolving Present and Future of Discrete Logarithm” [21] の Table 1 を編集し 2014 年 1 月以降の結果などを追記したものである.

3 Pinpointing を用いた関数体篩法の概要

表 1 が示すように, Frobenius representation algorithm は, 標数が小さい有限体上の離散対数問題を現時点で最も効率よく解く手法である. Frobenius representation algorithm の新たな方針は, 関係探索段階において sieving とは異なる手法で relation を効率よく生成することである. この方針が最初に採用されたのは関数体篩法 JL06-FFS [20] の 関係探索段階において pinpointing を導入した手法である [14]. この節では pinpointing 用いた関数体篩法について簡単に説明する. Frobenius representation algorithm [16, 5] については Hayashi が参考文献 [12] で簡明に説明している.

3.1 標数が小さい場合の関数体篩法の例

まず, 関数体篩法 JL06-FFS [20] について簡単に説明する. 有限体 $\mathbb{F}_{q^n}$ 上の DLP を JL06-FFS で解く場合, 二つの多項式 $f_1(x,y) = x - g_1(y), f_2(x,y) = -g_2(x) + y \in \mathbb{F}_q[x,y]$ を用意する. 但し g_1 と g_2 の次数をそれぞれ d_1, d_2 とし, $-g_2(g_1(y)) + y$ は $\mathbb{F}_q$ 上で既約な n 次多項式 $f(y)$ を因子として持つとする. さらに次数 d_1, d_2 と因子基底の最大次数 D は, $d_1 \approx \sqrt{Dn}$ と $d_2 \approx \sqrt{n/D}$ が成り立つように設定される.

この関数体篩法の関係探索段階では,

$$\mathcal{A}(y)g_1(y) + \mathcal{B}(y) = \mathcal{A}(g_2(x))x + \mathcal{B}(g_2(x))$$

の両辺が D-smooth となる一変数の $\mathbb{F}_q$ 係数多項式の組 $(\mathcal{A}(z), \mathcal{B}(z))$ を集める. 但し, $\mathcal{A}(z), \mathcal{B}(z)$ の次数は D 以下とし, さらに $\mathcal{A}(z)$ はモニックとする.

JL06-FFS の計算量は, $q = L_{q^n}(1/3, \alpha D)$ のとき, 関係探索段階の計算量は $L_{q^n}(1/3, c_1)$, 線型代数段階のそれは $L_{q^n}(1/3, c_2)$ となる. ただし

$$c_1 = \frac{2}{3\sqrt{\alpha D}} + \alpha D,\ c_2 = 2\alpha D$$

で, 次の条件が成り立つとする：

$$(D+1)\alpha \geq \frac{2}{3\sqrt{\alpha D}}.$$

3.2 Pinpointing

簡単な例として, 関数体篩法 JL06-FFS において $D = 1$ とした場合で, pinpointing について説明する. まず, $g_1(y) = y^{d_1}$ と設定し, $D = 1$ より $\mathcal{A}(z) = z + a$, $\mathcal{B}(z) = bz + c$ であることから, 次の形の relation の候補について考える:

$$y^{d_1+1} + ay^{d_1} + by + c = xg_2(x) + ax + bg_2(x) + c. \tag{1}$$

この両辺が 1 次多項式の積に分解できる (1-smooth である) 場合に relation が得られる.

3.2.1 One-sided pinpointing

式 (1) の左辺が 1-smooth であることと, $y = au$ とした場合に, 多項式 $u^{d_1+1} + u^{d_1} + ba^{-d_1}u + ca^{-d_1-1}$ が 1-smooth であることは同値である. 従って, $u^{d_1+1} + u^{d_1} + Bu + C \in \mathbb{F}_q$ の形の多項式に注目して, これが 1-smooth となる (B, C) が得られれば, その一つの (B, C) から $q-1$ 個の 1-smooth な多項式 $y^{d_1+1} + ay^{d_1} + by + c$ が得られる. ($a \in \mathbb{F}_q^*$ に対して $b = Ba^{d_1}, c = Ca^{d_1+1}$ とする.)

一つの 1-smooth な $u^{d_1+1}+u^{d_1}+Bu+C$ を得るために, 漸近的に $(d_1+1)!$ 個の候補が必要である. 従って (1) の左辺については $(d_1+1)!+(q-1)$ 個の候補が存在する. またそのときの $q-1$ 個の $a\in\mathbb{F}_q^*$ に対して, (1) の右辺が 1-smooth になる個数の期待値は $(q-1)/(d_2+1)!$ であることから, 一つの relation を得るために必要な候補の期待値は

$$\frac{(d_1+1)!+(q-1)}{(q-1)/(d_2+1)!}=\frac{(d_1+1)!(d_2+1)!}{q-1}+(d_2+1)!$$

となり, sieving の場合の $(d_1+1)!(d_2+1)!$ 個に比べてずっと小さい.

3.2.2 Kummer extensions, Frobenius and advanced pinpointing

拡大次数 n が d_1d_2-1 である Kummer extension の場合に, 式 (1) の両辺に pinpointing を行うことができる. さらに線型方程式の変数を実質的に $1/n$ 倍に減らすことができる.

有限体 $\mathbb{F}_q$ は 1 の原始 n 乗根 μ を含むとする. このとき $\mathbb{F}_q$ 上の n 次の Kummer extension は $P(x)=x^n-K$ で定義される. (K の設定に注意.) K の n 乗根 κ で $\kappa^q=\mu\kappa$ となるものが存在し,

$$P(x)=\prod_{i=0}^{n-1}(x-\mu^i\kappa)$$

とかける. そのような Kummer extension において, $g_1(y),g_2(x)$ を次のように定義する:

$$g_1(y)=y^{d_1}/K,\ g_2(x)=x^{d_2}. \tag{2}$$

このとき $x=g_1(y)$, $y=g_2(x)$ であることから, $x^{d_1d_2}-Kx=0$ となり両辺を x で割ることで $P(x)$ を得る.

$D=1$ で考えていることから因子基底は, $w\in\mathbb{F}_q$ に対して $x+w$ や $y+w$ の形をしている. これらの多項式は Frobenius map によって,

$$\begin{aligned}(x+w)^q &= x^q+w=\mu x+w=\mu(x+w/\mu),\\ (y+w)^q &= y^q+w=\mu y+w=\mu(y+w/\mu)\end{aligned}$$

となる. 従って, $\mathbb{F}_{q^n}^*/\mathbb{F}_q^*$ において,

$$\log(x+w/\mu)=q\log(x+q),\ \log(y+w/\mu)=q\log(y+q)$$

が成り立ち, 線型方程式の変数を減らすことができる.

One-side pinpointing のとき, 即ち式 (1) の場合と同様にして,

$$x^{d_2+1}+bx^{d_2}+ax+c=y^{d_1+1}/K+ay^{d_1}/K+by+c \tag{3}$$

について考える. 式 (3) の右辺が 1-smooth であることと, $u^{d_2+1}+u^{d_2}+ab^{-d_2}u+cb^{-d_2-1}$ が 1-smooth であることは同値であり, 同様に左辺については $v^{d_1+1}/K+v^{d_1}/K+ab^{-d_1}v+cb^{-d_1-1}$ が対応する. さらに $\lambda=c/(ab)$ とすることで, u,v を変数とするこれらの多項式はそれぞれ次のように書くことができる:

$$u^{d_2+1}+u^{d_2}+ab^{-d_2}(u+\lambda),\ (v^{d_1+1}+v^{d_1})/K+ab^{-d_1}(v+\lambda).$$

逆に (A,B,λ) を, $A\neq0$, $B\neq0$, AB^{d_2} が $\mathbb{F}_q$ において n 冪となり (Kummer extension を使用している), さらに

$$u^{d_2+1}+u^{d_2}+A(u+\lambda),\ (v^{d_1+1}+v^{d_1})/K+B(v+\lambda)$$

がそれぞれ 1-smooth となるように選ぶ. このとき, $A = ab^{-d_2}$, $B = ba^{-d_1}$ とすることで, $AB^{d_2} = a^{1-d_1 d_2} = a^{-n}$ から a を定めることができ, さらにその選び方は n とおりである. 各 a に対して $b = Ba^{d_1}$, $c = \lambda ab$ と定める.

最終的に relation 一つ当たりのコストは

$$O\left(\frac{n(d_1+1)!(d_2+1)!}{q-1}\right)+1$$

となるが, Frobenius map の効果で n を相殺できる.

3.3 計算量

$\mathbb{F}_{q^n}$ 上の離散対数問題を, pinpointing を導入した JL06-FFS で解くことを考える. ここで $Q = q^n$ とし, α は次を満たすとする:

$$\alpha = \frac{1}{n}\left(\frac{\log Q}{\log\log Q}\right)^{2/3}.$$

$D = 1$ とした場合に linear algebra step の計算量は $L_Q(1/3, 2\alpha)$ となる. $\alpha \geq 3^{-2/3}$ に対して, このコストは (双方の) pinpointing のコストより大きいため, 総計算量は $L_Q(1/3, 2\alpha)$ となる. $\alpha \in [3^{-2/3}, 2^{2/3})$ に対しては JL06-FFS よりも総計算量は小さくなり, とくに $\alpha = 3^{-2/3}$ のとき, 総計算量は $L_Q(1/3, 1.44)$ から $L_Q(1/3, 0.96)$ に減少する.

3.4 数値実験

まず, $p_1 = 33553771$, $p_2 = 33341353$ とする. このとき有限体 $\mathbb{F}_{p_1^{47}}$ と $\mathbb{F}_{p_2^{57}}$ の大きさはそれぞれ 1175-bit と 1425-bit となる. これらの有限体上の離散対数問題を Advanced pinpointing を使用して解く数値実験を行った場合, 双方とも 32000 CPU-hours を必要としたとの報告がある.

表 2: 文献 [14] の実験結果

Bitsize	Total time (CPU-hours)	Relation construction (CPU-hours)	Linear algebra (CPU-hours)	Indiv. Log. (CPU-hours)
1175	約 32000	3	32000	4
1425	約 32000	6	32000	< 12

4 更新履歴

更新日時	主な更新内容
2015 年 2 月	•概要を追加. •2 節. 表 1 とその解説を加筆.

参考文献

[1] L. M. Adleman, M-D. A. Huang, "Function field sieve method for discrete logarithms over finite fields," Inf. Comput., 151 (1999), 5-16.

[2] G. Adj, A. Menezes, T. Oliveira, F. R. Henríquez, "Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for Discrete Logarithm Cryptography," Proc. of Pairing 2013, LNCS 8365 (2013), 20-44.

[3] G. Adj, A. Menezes, T. Oliveira, F. R. Henríquez, "Computing Discrete Logarithms in $\mathbb{F}_{3^{6\cdot 137}}$ and $\mathbb{F}_{3^{6\cdot 163}}$ using Magma," Proc. of WAIFI 2014, LNCS 9061 (2015), 3-22.

[4] R. Barbulescu, C. Bouvier, J. Detrey, P. Gaudry, H. Jeljeli, E. Thomé, M. Videau, P. Zimmermann, "Discrete Logarithm in $GF(2^{809})$ with FFS, Proc. of Public Key Cryptography 2014, LNCS 8383 (2014), 221-238.

[5] R. Barbulescu, P. Gaudry, A. Joux, E. Thomé, "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic," Proc. of EUROCRYPT 2014, LNCS 8441 (2014), 1-16.

[6] D. Coppersmith, "Fast evaluation of logarithms in fields of characteristic two," IEEE Transactions on Information Theory, 30/4 (1984), 587-593.

[7] F. Göloğlu, R. Granger, G. McGuire, J. Zumbrägel, "On the function field sieve and the impact of higher splitting probabilities - application to discrete logarithms in $\mathbb{F}_{2^{1971}}$ and $\mathbb{F}_{2^{3164}}$," Proc. of CRYPTO 2013, LNCS 8043 (2013), 109-128.

[8] F. Göloğlu, R. Granger, G. McGuire, J. Zumbrägel, "Solving a 6120 -bit DLP on a Desktop Computer," Proc. of SAC 2013, LNCS 8282 (2013), 136-152.

[9] R. Granger, T. Kleinjung, J. Zumbrägel, "Discrete Logarithms in `GF(2^9234)`," `https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1401&L=NMBRTHRY&F= &S=&P=8736`.

[10] R. Granger, T. Kleinjung, J. Zumbrägel, "Breaking '128-bit secure' supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$)," Proc. of CRYPTO 2014, LNCS 8617 (2014), 126-145.

[11] D. M. Gordon, K. S. McCurley, "Massively Parallel Computation of Discrete Logarithms," Proc. of CRYPTO 1992, LNCS 740 (1992), 312-323.

[12] T. Hayashi, "Cryptanalysis of Pairing-based Cryptosystems Over Small Characteristic Fields," Proc. of the Forum of Mathematics for Industry 2013, 1 (2013), 167-176.

[13] T. Hayashi, T. Shimoyama, N. Shinohara, T. Takagi, "Breaking Pairing-Based Cryptosystems Using η_T Pairing over $GF(3^{97})$," Proc. of ASIACRYPT 2012, LNCS 7658 (2012), 43-60.

[14] A. Joux, "Faster index calculus for the medium prime case. Application to 1175-bit and 1425-bit finite fields," Proc. of EUROCRYPT 2013, LNCS 7881 (2013), 177-193.

[15] A. Joux, "Discrete Logarithms in `GF(2^1778)`," `https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1302&L=NMBRTHRY&F=&S=&P=2317`.

[16] A. Joux, "A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic," Proc. of SAC 2013, LNCS 8282 (2013), 355-379.

[17] A. Joux, "Discrete Logarithms in GF(2^4080)," https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1303&L=NMBRTHRY&F=&S=&P=13682.

[18] A. Joux, "Discrete Logarithms in GF(2^6168) [=GF((2^257)^24)]," https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1305&L=NMBRTHRY&F=&S=&P=3034.

[19] A. Joux and R. Lercier, "The function field sieve is quite special," Proc. of ANTS 2002, LNCS 2369 (2002), 431-445.

[20] A. Joux and R. Lercier, "The function field sieve in the medium prime case," Proc. of EUROCRYPT 2006, LNCS 4004 (2006), 254-270.

[21] A. Joux, A. Odlyzko, C. Pierrot, "The Past, evolving Present and Future of Discrete Logarithm," Open Problems in Mathematical and Computational Science Book, (2014), to appear.

[22] A. Joux, C. Pierrot, "Improving the Polynomial time Precomputation of Frobenius Representation Discrete Logarithm Algorithms - Simplified Setting for Small Characteristic Finite Fields," Proc. of ASIACRYPT 2014, LNCS 8873 (2014), 378-397.

[23] Kleinjung, "Discrete Logarithms in GF(2^1279)," https://listserv.nodak.edu/cgi-bin/wa.exe?A2=ind1410&L=NMBRTHRY&F=&S=&P=1170.

[24] E. Thomé, "Computation of Discrete Logarithms in $\mathbb{F}_{2^{607}}$," Proc. of ASIACRYPT 2001, LNCS 2248 (2001), 107-124.

資料1
別添3

格子問題等の困難性に関する調査

2015 年 3 月 26 日版

目次

第 1 章

調査の目的

公開鍵暗号の安全性は, 素因数分解の困難性や離散対数問題の困難性などさまざまな数学的問題に依存している. 本ワーキンググループではこれまで, 素因数分解の困難性及び離散対数問題等の困難性に関する調査を行ってきたが, 量子計算機が実現しても安全性が保たれると期待されている「耐量子計算機暗号」を支える数学的問題の困難性の中でも, 特に近年活発に研究されてきている, 格子に係る数学的問題等に注目して調査を行った.

1.1 調査の概要

各章の執筆担当者及び調査内容は下表の通りである.

章	執筆委員名	内容
第 1 章	事務局	調査の目的, 調査の概要など
第 2 章	石黒 司 委員[*1]	General な攻撃に関する総論
第 3 章	下山 武司 委員 安田 雅哉 委員	各問題について以下の項目を記述 (1) 公開鍵方式からの帰着, 証明の有無, 追加の問題・制約など
第 4 章	草川 恵太 委員	(2) 攻撃や量子アルゴリズム - General な攻撃との関係
第 5 章	國廣 昇 委員	- 固有の攻撃 - 量子アルゴリズムとの関係

第 2 章から第 5 章までの調査内容をまとめると, 下記の通りとなる.

第 2 章 格子の SVP (Shortest Vector Problem) は, ランダム帰着の元で NP 困難問題であることが示されている問題である. SVP (近似版を含む) のうち, 近似因子が次元の多項式で表される場合に適用される, 4 つの解読アルゴリズム (LLL, BKZ, 篩, ボロノイセル) の計算量等に関する概説を行った. 計算機実験 (SVP Challenge, Lattice Challenge, Ideal Lattice Challenge) に関しては, 日本の研究者らの実験結果もいくつかなされている.

第 3 章 LWE (Learning with Errors) 問題は, Machine Learning (機械学習理論) から派生した問題で, GapSVP (the decision version of the shortest vector problem) 及び SIVP (the shortest independent vectors problem) の困難性に関する仮定のもとで解くことが難しいことが知られており, 本問題を効率的に解くことは困難であると予想されている. 現在までに完全準同型暗号スキームをはじめとした, 様々な公開鍵暗号スキームのベースがこの LWE 問題をベースとして提案されており, 今後も安全な暗号を構成する上で重要な要素となると考えられる.

現在までに知られている LWE 問題を解く最良アルゴリズムは指数時間の計算量を持っている．ただし, 実際の LWE 問題をベースとした暗号スキームの構成の際には, BKZ アルゴリズムなどの格子縮約アルゴリズムに対し耐性を持つようにパラメータ設定を行う必要があり, 安全でかつ演算機能等の要件を満足するような LWE パラメータを選択するための, 統一的な方法は知られておらず, 今後の課題となっている．また, LWE 問題に対する攻撃実験評価に関する結果もあまり知られていないため, 今後は計算機実験に関する研究も非常に重要になると思われることから, 安全性理論評価はもちろん攻撃実験評価の視点からも, 今後の動向に注意する必要がある．

第 4 章　LPN 問題は学習理論や符号理論から派生した問題である．誤り確率が十分大きい場合の LPN 問題を多項式時間で効率的に解くことは困難であると予想されている．共通鍵や公開鍵の分野で多くの方式が LPN 問題に基づいて提案されている．LWE 問題と比較した場合, 利点としては, ハードウェア構成との相性が良い点や誤差のサンプリングが容易である点が挙げられる．一方、欠点として, 鍵や暗号文のサイズが大きくなりやすい点や発展的な応用が少ない点が挙げられる．暗号方式のパラメータ設定の際には, 4.3 節で挙げたさまざまなアルゴリズムを考慮する必要がある．アルゴリズムの高速化について盛んに研究されており、動向を注視する必要がある．また、攻撃に用いられるアルゴリズムの研究は理論的なものが多く, 攻撃実験報告は小さいパラメータに対して行ったものが多い．そのため, 攻撃実験に関する研究もこれから非常に重要である．

第 5 章　ACD 問題は, 2001 年に Howgrave-Graham により導入された問題であり, パラメタを適切に選ぶことにより, 効率的に解くことが困難であると予想されている．ACD 問題は, 複数 ACD 問題や GACD 問題など, いくつかの拡張問題をもつ．ACD 問題を, 素因数分解を直接的に経由しないで解くアルゴリズムには, 大別すると, 組み合わせ論に基づく方法と格子理論に基づく方法がある．組み合わせ論に基づくアルゴリズムを用いた場合では, 指数関数時間の計算量が必要であるが, 全数探索アルゴリズムの平方根の計算量で解を求めることができる．最近提案された Chen-Nguyen のアルゴリズムは, 暗号の提案時には考慮されていなかった攻撃であり, 提案論文で書れた推奨パラメタのいくつかが実際に解読されることが示されている．格子理論に基づくアルゴリズムを用いた場合では, 法に対して, 解がある制限よりも小さいときには, 多項式時間で解くことができるものの, 解が十分大きいときには, 解を求めることができない．また, ACD 問題に関連した問題, co − ACD 問題は, 当初の想定よりも弱いことが明らかになっている．これらの結果は, ごく最近に示されたものであり, 今後の研究の動向に注視する必要がある．ACD 問題を安全性の根拠としてもつ, 完全準同型暗号方式が提案されている．適切にパラメータが設定された状況では, 攻撃に成功するのに指数関数時間が必要であるが, 理論上の解析であるため数値実験により安全性の検証をする必要がある．

1.2 2013-2014 年度 暗号技術調査ワーキンググループ (暗号解析評価) の委員構成

主査	高木 剛	国立大学法人九州大学 マス・フォア・インダストリ研究所 教授
委員	青木 和麻呂	日本電信電話株式会社 NTT セキュアプラットフォーム研究所 主任研究員
委員	石黒 司[*2]	株式会社 KDDI 研究所 情報セキュリティ G 研究員
委員	太田 和夫	国立大学法人電気通信大学 大学院 情報理工学研究科 総合情報学専攻 (セキュリティ情報学コース) 教授
委員	草川 恵太	日本電信電話株式会社 NTT セキュアプラットフォーム研究所 研究員
委員	國廣 昇	国立大学法人東京大学大学院 新領域創成科学研究科複雑理工学専攻 准教授
委員	下山 武司	株式会社富士通研究所 ソーシャルイノベーション研究所 セキュアコンピューティング研究部 主任研究員
委員	安田 雅哉	株式会社富士通研究所 ソーシャルイノベーション研究所 セキュアコンピューティング研究部

1.3 更新履歴

更新日時	主な更新内容
2014 年度	●2.1.3 節. 計算機実験に関する記録の更新. ●3.1.3 節の追加. ●3.2.2 節の最後.「■近年の攻撃研究の動向」の追加. ●4.2 節. 代表的な暗号方式を追加 (旧 4.3-4.5 節から移動した文書有り). ●4.3 節 (旧 4.2 節). いくつかコメントを追加. ●5.1.3 節. 5.1.3.1 節及び 5.1.3.2 節の追加. ●5.1.4 節の追加. ●5.5 節の追加.

[*2] 2013 年度まで

第 2 章

総論

2.1 一般的な攻撃に関する総論

本章では, 一般的な攻撃に関する攻撃についてまとめる. 格子に関する困難性問題の中でベースとなる問題は格子の最短ベクトル問題 (SVP) である. 本章では, この格子の最短ベクトル問題の定義と, それに関連するアルゴリズムについてまとめる. 更に, 実際の計算機環境における解析の現状についてまとめる. 最短ベクトル問題は, 格子暗号における重要な困難性問題の一つであり, この問題が解けると, 次章以降で説明する LWE 問題などの格子問題も解けるため計算量解析がとりわけ重要である.

本章で使用する記号・用語を以下にまとめる. $\boldsymbol{b}_i = (b_1, b_2, \ldots, b_n) \in \mathbb{R}^n$ を n 個の一次独立なベクトルとする $(1 \leq i \leq n)$. $\boldsymbol{b}_i$ を列ベクトルとする行列を $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{n \times n}$ とする. この時,

$$\mathcal{L}(\mathrm{B}) = \mathcal{L}(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n) = \left\{ \sum_{1 \leq i \leq n} x_i \boldsymbol{b}_i, x_i \in \mathbb{Z} \right\}$$

を格子とする. また, B を格子基底と呼ぶ. 本章では格子の次元を n とする. ベクトル $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ のノルム (長さ) を $||\boldsymbol{v}|| = (\sum_{1 \leq i \leq n} v_i^2)^{1/2}$ とする. また, 基底 B の最短ベクトルかつ非零ベクトルのノルムを $\lambda_1(\mathrm{B})$ あるいは単に λ_1 と表す. 格子 B のグラムシュミット直交化基底を $\mathrm{B}^* = (\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \ldots, \boldsymbol{b}_n^*)$ とする. $\boldsymbol{b}_i^*$ は, $\boldsymbol{b}_1^* = \boldsymbol{b}_1$ として, $2 \leq i \leq n$ について以下のように帰納的に定義される.

$$\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{1 \leq j \leq i-1} \mu_{i,j} \boldsymbol{b}_j^*, \quad \mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{||\boldsymbol{b}_j^*||}$$

$\mu_{i,j}$ をグラムシュミット係数とよぶ. 基底 $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n), i \in \{1, 2, \ldots, n\}$ における直交射影 $\pi_i : \mathbb{R}^n \to \mathbb{R}^n$ を $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{i-1})$ が生成する部分空間の直交補空間への射影写像とし, $\pi_i(\boldsymbol{v}) = \sum_{1 \leq i \leq n} a_i \boldsymbol{b}_i^*$ と表す. $i \leq j$ となる基底ベクトル $\boldsymbol{b}_j$ に対して $\pi_i(\boldsymbol{b}_j) = \boldsymbol{b}_j^{(i)}$ と表す. また, 格子の射影部分格子を $\mathcal{L}_{[j,k]} = \mathcal{L}((\boldsymbol{b}_i^{(j)})_{j \leq i \leq \min(j+k-1, n)})$ とする.

2.1.1 最短ベクトル問題 (SVP)

格子の最短ベクトル問題を SVP(Shortest Vector Probrem) とよぶ. これはある格子の基底が与えられた時に, その格子上のベクトルの中で長さが最小となる非零ベクトルを探索する問題である. 一般に, 最短ベクトルは必ずしも一つではないため, 最短ベクトルの中の一つのベクトルを見出せば SVP の解となる. また, 長さが最短ベクトルの α 倍以下となるベクトルのうちの一つを探索する問題を近似版最短ベクトル問題 (α-SVP) とよぶ. 以下にそれぞれ詳細な定義を示す.

定義 2.1 (最短ベクトル問題 (SVP)) 格子 $\mathcal{L}(\mathrm{B})$ が与えられて, 格子に含まれるベクトル $\boldsymbol{v} \in \mathcal{L}(\mathrm{B})$ のうちでノルムが最小の非零ベクトル (つまり, $|\boldsymbol{v}| = \lambda_1$) の一つを求める問題を最短ベクトル問題 (SVP) と呼ぶ.

最短ベクトルのノルムについて以下の定理が知られている.

定理 2.2 (ミンコフスキーの第 1 定理) 格子 $\mathcal{L}(\mathrm{B})$ に対して最短ベクトルのノルムは, $\sqrt{n}(\mathrm{vol}(\mathcal{L}(\mathrm{B})))^{\frac{1}{n}}$ 未満となる.

また, より精緻な見積りとしてガウスヒューリスティックスが知られている. ガウスヒューリスティックスによって格子 $\mathcal{L}(\mathrm{B})$ の最短ベクトルのノルムは $GH(\mathcal{L}(\mathrm{B})) = (1/\sqrt{\pi})\Gamma(\frac{n}{2}+1)^{\frac{1}{n}} \cdot |\det(\mathcal{L}(\mathrm{B}))^{\frac{1}{n}}|$ 程度と見積もられる. ここで, $\Gamma(x)$ はガンマ関数を表す. 最短ベクトル問題は, 上記の通り厳密解を求める問題として定義されている. 一方, 暗号アルゴリズムでは最短ベクトルの近似解を求める問題の困難性をベースとして構成される場合もある. 以下に近似版最短ベクトル問題 (α-SVP) を定義する.

定義 2.3 (近似版最短ベクトル問題 (α-SVP)) 格子 $\mathcal{L}(\mathrm{B})$ が与えられて, 格子に含まれるベクトル $\boldsymbol{v} \in \mathcal{L}(\mathrm{B})$ のうちでノルムが $||\boldsymbol{v}|| < \alpha\lambda_1$ となるベクトルの一つを求める問題を近似版最短ベクトル問題 (α-SVP) と呼ぶ. また, α を近似因子と呼ぶ.

2.1.2 求解アルゴリズムと計算量

SVP は Ajtai によって, ランダム帰着の元で NP 困難問題であることが示されている [1]. α-SVP については, 近似因子 $1 < \alpha < \sqrt{2}$ となる範囲ではランダム帰着の元で NP-困難であることが Micciancio[18] によって示され, 任意の定数 α の元での NP 困難性が Khot によって証明されている [16, 17]. 一方, 近似因子が格子の次元 n の多項式となる場合, すなわち $\alpha = poly(n)$ の場合の NP 困難性については証明されておらず, 重要な研究課題となっている. 本節では, SVP, α-SVP それぞれについて求解アルゴリズムを解説する.

■α-SVP　α-SVP を解くアルゴリズムとして, LLL[12], BKZ[31] アルゴリズムがある. LLL アルゴリズムは, Lenstra, Lenstra, Lovász によって提案されたアルゴリズムである. LLL アルゴリズムは格子の基底を入力とし, LLL 簡約基底とよばれる入力された基底と同じ格子を張る別の基底を求めるアルゴリズムである. この LLL 簡約基底は, 基底ベクトルのノルムに制約がある格子基底となっており, 以下のように定義される.

定義 2.4 (簡約基底) 格子基底を B とする. このとき B^* のグラムシュミット係数 $\mu_{i,j}(1 \leq j < i \leq n)$ が $|\mu_{i,j}| < \frac{1}{2}$ を満足するとき, B は簡約基底という.

定義 2.5 (δ-LLL 簡約基底) 格子基底を $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ とし, $\delta \in (0.25, 1]$ とする. 格子 B が簡約基底であり, かつ

$$\delta||\boldsymbol{b}_{i-1}^*||^2 \leq ||\boldsymbol{b}_i^*||^2 + \mu_{i,i-1}^2||\boldsymbol{b}_{i-1}^*||^2$$

という条件を満足するとき, B は δ-LLL 簡約基底という. また, この条件を Lovász 条件とよぶ.

LLL 簡約アルゴリズムを用いると, LLL 簡約基底を求めることができ, 基底ベクトルがノルムの大きさが小さい方から順番に整列される. このとき, $||\boldsymbol{b}_1|| \leq (\frac{2}{\sqrt{3}})^n\lambda_1$ となることが証明されているため, 近似因子 $\alpha = (\frac{2}{\sqrt{3}})^n$ における α-SVP の解とすることができる.

LLL アルゴリズムの概要を以下に示す. 入力は, 格子基底 $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ とし δ-LLL 簡約基底を出力する. LLL アルゴリズムは $\boldsymbol{b}_1$ から順に $\boldsymbol{b}_n$ に向かって簡約を行う. まず, $\boldsymbol{b}_j$ を簡約基底の条件を満足するために $k < j$ に対

して, $\boldsymbol{b}_j = \boldsymbol{b}_j - \lceil \mu_{j,k} \rfloor \boldsymbol{b}_k$ を計算し, $\boldsymbol{b}_j$ に合わせて $\mu_{j,k}$ を再計算する. 次に, $\boldsymbol{b}_j$ が Lovász 条件を満足しない場合には $\boldsymbol{b}_j$ と $\boldsymbol{b}_{j-1}$ を入れ替え, $j = j - 1$ として上記を繰り返す. この処理によって $j = 1$ から $j = n$ まで $\boldsymbol{b}_j$ を簡約する. LLL アルゴリズムは多項式回のループで停止することが示されており, 計算量は $O(n^4 \log(\max_{1 \leq i \leq n} ||\boldsymbol{b}_i||^2))$ となる. また, 出力される基底の第一ベクトルのノルムは $||\boldsymbol{b}_1|| \leq (\frac{2}{\sqrt{3}})^n \lambda_1$ となることが証明されている [12]. 計算機実験上はこの見積りよりも短いベクトルが出力されることが多く, 特に小さい次元の場合には LLL アルゴリズムを用いて最短ベクトルを求めることができる.

LLL を改良したアルゴリズムとして BKZ アルゴリズムが Schnor 等によって提案されている. BKZ アルゴリズムは BKZ 簡約基底を出力するアルゴリズムである. BKZ 簡約基底は LLL 簡約基底よりも広い定義となっており, 以下のように定義される.

定義 2.6 (β-BKZ 簡約基底) 格子基底を $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ とし, $\beta \in [2, n]$ とする. 格子 B が LLL 簡約基底であり, かつ $1 \leq j \leq n$ について $||\boldsymbol{b}_j^*|| = \lambda_1(\mathcal{L}_{[j,\beta]})$ を満足するとき, B は β-BKZ 簡約基底という.

β-BKZ 簡約基底は LLL 簡約基底を拡張したものであり, $\beta = 2$ の場合には LLL 簡約基底そのものになる. BKZ アルゴリズムの概要を以下に示す. BKZ アルゴリズムの入力は, LLL 簡約基底 $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ とし β-BKZ 簡約基底を出力する. まず, $i = 1, 2, \ldots, n-1$ について $\pi_i(\boldsymbol{b})$ が $\mathcal{L}_{[i,\beta]}$ で最短ベクトルとなるような $\boldsymbol{b} \in \mathcal{L}(\mathrm{B})$ を探索する. このようなベクトルは次節で説明する SVP を解くアルゴリズムを用いて求めることができる. 次にこの $||\pi_i(\boldsymbol{b})|| < ||\boldsymbol{b}_i^*||$ となる場合には基底 B にベクトル $\boldsymbol{b}$ を i 番目に挿入し基底 $\mathrm{B}' = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_i, \boldsymbol{b}, \boldsymbol{b}_{i+1}, \ldots, \boldsymbol{b}_n)$ を構成する. これに LLL 簡約基底を適用し, 新たな基底とする. 新たな基底に対して上記を繰り返し, 基底が更新されなくなるまで繰り返すことによって基底簡約を行う. BKZ アルゴリズムの停止性や計算量は証明されていないが, 計算機実験上は高速に動作し, LLL アルゴリズムよりも大きな次元に対して適用することができる. BKZ アルゴリズムを改良したアルゴリズムとして BKZ2.0 アルゴリズム [7, 4] が提案されており, より大きな次元の α-SVP が解けることが示されている. また, ランダムに短いベクトルを生成して基底に挿入し, そこに BKZ アルゴリズムを適用することによって基底を簡約する RSR アルゴリズムも提案されている [34, 13].

■SVP　SVP を解くアルゴリズムとして以下のいくつかの種類のアルゴリズムが提案されている. 代表的な求解手法として, 格子基底簡約アルゴリズム, 列挙アルゴリズム, ボロノイセルアルゴリズム, 篩アルゴリズムがある.

格子基底簡約アルゴリズムは, 上記で説明した LLL,BKZ アルゴリズムなどの基底簡約アルゴリズムであり, 格子基底に適用することによって SVP を解くことができる. 代表的な格子基底簡約アルゴリズムとして LLL アルゴリズム [12], BKZ アルゴリズム [31], L^2 アルゴリズム [21, 22], BKZ2.0[9, 4] がある.

列挙アルゴリズムは, 所謂全数探索で可能性のある係数の総当り探索を行い, 最短ベクトルを見つけるアルゴリズムである. 格子ベクトル $\boldsymbol{v} \in \mathcal{L}(\mathrm{B})$ は, 基底ベクトル $\boldsymbol{b}$ を用いて, $\boldsymbol{v} = \sum_{1 \leq i \leq n} u_i \boldsymbol{b}_i$ と表せる. したがって, 可能性のある全ての係数 $[u_1, u_2, \ldots, u_n]$ を列挙することによって最短ベクトルを見つける事ができる. 列挙アルゴリズムは, Schnorr によって示され [32], 更に探索範囲を削減する枝刈り列挙 ([33, 9, 20]) アルゴリズムが提案されている. 現時点で最も高速な枝刈り列挙アルゴリズムは Gama, Nguyen, Regev によって提案された Extream Pruning Enumeration アルゴリズムである [9]. このアルゴリズムの時間計算量は $2^{O(n)}$ である. 列挙アルゴリズムは特に比較的小さい次元において高速に SVP を解くことができるため, BKZ アルゴリズムの内部関数としても用いられている. 列挙アルゴリズムの計算量を表 2.1 に示した. 列挙アルゴリズムは並列化が容易であることから, GPU 上での高速実装や, クラウドコンピューティングを用いた大規模並列計算によって大きな次元の SVP の求解報告がなされている [27].

Micciancio によってボロノイセルアルゴリズムが提案されている [19]. ボロノイセルアルゴリズムは決定的アルゴリズムであり, $2^{O(n)}$ の時間計算量, 空間計算量となることが示されている. しかし, 現在のところボロノイセルアルゴリ

表 2.1　列挙アルゴリズムの計算量

アルゴリズム	時間	空間	文献
ENUM	$2^{O(n^2)}$	$O(n)$	文献 [32]
Extream Pruning Enumeration	$2^{O(n)}$	$O(n)$	文献 [9]

表 2.2　篩アルゴリズムの計算量

アルゴリズム	時間計算量	空間計算量	文献
AKS Sieve	$O(2^{5.90n})$	$O(2^{2.95n})$	文献 [3]
AKS Sieve without perturbation	$O(2^{0.41n})$	$O(2^{0.21n})$	文献 [23]
List Sieve	$O(2^{3.199n})$	$O(2^{1.325n})$	文献 [19]
Gauss Sieve	$O(2^{0.52n})$	$O(2^{0.21n})$	文献 [19]
List Sieve Birthday	$O(2^{2.465n})$	$O(2^{1.233n})$	文献 [26]
NV Sieve	$O(2^{0.3836n})$	$O(2^{0.2557n})$	文献 [23, 35]

ズムの実装例は知られていない.

篩アルゴリズムは SVP を解く確率的アルルゴリズムである. 2001 年に Ajtai 等によって AKS Sieve[3] が提案され, それ以降, より計算量を削減したアルゴリズムが提案されている [23, 6, 5, 19, 26, 35]. 一般に篩アルゴリズムの時間・空間計算量は $2^{O(n)}$ である. 現在, 理論上最も高速な篩アルゴリズムは NV Sieve であり, 時間計算量は $O(2^{0.3836n})$, 空間計算量は $O(2^{0.2557n})$ となっている. 篩アルゴリズムの計算量を表 2.2 に示した.

2.1.3　計算機実験

本章では, 計算機実験によって実際に解かれた SVP についてまとめる. 現在, ダルムシュタット工科大学によって SVP に関するコンテストが開催されている. このコンテスによって統一された問題設定においてアルゴリズム・実装性能の評価が可能となっている. しかし実験環境, 計算機環境についての制限はないため, アルゴリズムや実装手法以外にも, 計算機性能や実験規模などが異なることに注意する必要がある.

SVP Challenge[30] はランダムに与えられた格子基底に対して SVP を解き, より大きい次元について, より短いベクトルを求めることによって順位が競われている. コンテストのサイトには, 実際に解かれたベクトルが掲載されている. ただし, 掲載されているベクトルは必ずしも最短のベクトルではないことに注意されたい. Lattice Challenge[29] は与えられた格子基底について α-SVP を解き, SVP チャレンジと同様により大きい次元, より短いベクトルを解くことが競われている. Ideal Lattice Challenge[24] は, イデアル格子と呼ばれる, 暗号で用いられることが多い特殊な格子 [11, 8, 10] に対する SVP, α-SVP の問題が掲載されている. コンテストに掲載されている問題の設定については文献 [25] を参照にされたい.

α-SVP に対する実験結果を表 2.3 に表す. 現在, α-SVP の求解は BKZ2.0 アルゴリズム [9], あるいはその改良方式 [7, 4] が用いられており, 825 次元までの α-SVP が解かれている. 詳細なアルゴリズム, 計算機環境についてはそれぞれの文献を参照されたい. また, SVP に対する実験結果を 2.3 に示す. SVP Challenge の結果として Kashiwabara らの RSR アルゴリズムの改良手法 [15], BKZ2.0[9] が有効であることが示されており, 最も大きな次元に対する求解は

表 2.3　q-ary lattice に対する, Approx-SVP の求解 (Lattice Challenge[29])

	次元	ノルム	アルゴリズム	時期	文献
Chen, Nguyen	825	120.37	BKZ2.0 の改良	2013-3	
Aono, Naganuma	825	122.38	BKZ2.0 の改良	2012-10	文献 [4]
Chen, Nguyen	800	106.60	BKZ2.0	2013-3	文献 [7]
Aono, Naganuma	800	117.69	BKZ2.0 の改良	2012-10	文献 [4]
Chen, Nguyen	775	100.14	BKZ2.0 の改良	2013-3	文献 [7]
Aono, Naganuma	775	106.68	BKZ2.0 の改良	2012-10	文献 [4]
Chen, Nguyen	750	87.76	BKZ2.0	2013-3	文献 [7]
Chen, Nguyen	725	80.65	BKZ2.0	2013-3	文献 [7]
Aono, Naganuma	725	83.61	BKZ2.0 の改良	2012-9	文献 [4]
Chen, Nguyen	700	72.46	BKZ2.0	2013-3	文献 [7]
Aono, Naganuma	700	76.17	BKZ2.0 の改良	2012-9	文献 [4]

表 2.4　Ideal-SVP(< 1.05 Gaussian heuristic) の求解 (Ideal Lattice Challenge[24])

	次元	ノルム	アルゴリズム	時期	文献
Ishiguro, Kiyomoto, Miyake, Takagi	128	2959	Gauss Sieve の改良	2013-4	文献 [14]
Ishiguro, Kiyomoto, Miyake, Takagi	108	2669	Gauss Sieve の改良	2013-4	文献 [14]

表 2.5　Approx-SVP($n \det^{1/n}$) の求解 (Ideal Lattice Challenge[24])

	次元	ノルム	アルゴリズム	時期	文献
Wang, Aono, Hayashi, Takagi	500	507596	Progressuve BKZ	2015-1	SCIS2015

Kashiwabara らによる RSR アルゴリズムの改良方式などである [15]. 彼らの手法は, 短いベクトルの統計的な情報から, 最短ベクトルの分布を予測し高速に短いベクトルを生成するように改良している.

また, Ideal Lattice Challenge においては 128 次元の SVP が解かれている [14]. 彼らの手法は, 篩アルゴリズムの一つである Gauss Sieve アルゴリズムの並列化によって 84 台の計算機を用いて 128 次元の SVP を求めている. Gauss Sieve アルゴリズムはイデアル格子の性質を用いて次元が 2 の冪乗となる場合に高速化できることが示されている. 更に, イデアル格子のいくつかの次元において Gauss Sieve を高速化できる条件も見つかっているが, 一般のイデアル格子の性質を用いた高速化手法は, 他の求解手法も含めて見つかっていないため, 格子暗号の安全性を議論する上で重要な研究課題となっている.

表 2.6　SVP の求解 (SVP Challenge[30])

	次元	ノルム	アルゴリズム	時期	文献
Kashiwabara, Teruya	140	3025	RSR アルゴリズムの改良	2015-1	
Kashiwabara, Teruya	138	3077	RSR アルゴリズムの改良	2014-12	
Kashiwabara, Teruya	134	2976	RSR アルゴリズムの改良	2014-7	耐量子暗号 WS(2014 年 11 月)
Kashiwabara, Fukase	132	3012	RSR アルゴリズムの改良	2014-4	耐量子暗号 WS(2014 年 11 月)
Aono, Nguyen	130	2883	BKZ2.0 + Randomized ENUM	2014-10	
Kashiwabara, Fukase	130	3025	RSR アルゴリズムの改良	2013-11	文献 [15]
Kashiwabara, Fukase	128	2984	RSR アルゴリズムの改良	2013-9	文献 [15]
Aono, Nguyen	126	2855	BKZ2.0 + Extreme pruning	2014-9	
Kashiwabara, Teruya	126	2897	RSR アルゴリズムの改良	2014-8	
Aono	126	2906	BKZ2.0 + Extreme pruning	2014-7	
Kashiwabara, Fukase	126	2944	RSR アルゴリズムの改良	2013-9	文献 [15]
Chen, Nguyen	126	2969	BKZ2.0 + Randomized ENUM	2013-4	文献 [7]
Chen, Nguyen	124	2884	BKZ2.0 + Randomized ENUM	2013-3	文献 [7]
Chen, Nguyen	122	2913	BKZ2.0 + Randomized ENUM	2013-3	文献 [7]
Kashiwabara, Fukase	120	2756	BKZ2.0 の改良	2013-3	文献 [4]
Aono, Naganuma	120	2830	BKZ2.0 の改良	2013-9	文献 [15]

第 2 章の参照文献

[1] M. Ajtai. The Shortest Vector Problem in L^2 is NP-hard for Randomized Reductions (Extended Abstract). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, STOC'98, pages 10–19. ACM, 1998.

[2] M. Ajtai and C. Dwork. A Public-key Cryptosystem with Worst-case/average-case Equivalence. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, STOC'97, pages 284–293. ACM, 1997.

[3] M. Ajtai, R. Kumar, and D. Sivakumar. A Sieve Algorithm for the Shortest Lattice Vector Problem. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, STOC'01, pages 601–610. ACM, 2001.

[4] 青野，長沼BKZ2.0 アルゴリズムの実装と改良. 信学技報, vol. 112, no. 211, ISEC2012-45, pp. 15–22, 2012.

[5] V. Arvind and P. S. Joglekar. Some Sieving Algorithms for Lattice Problems. In *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS'08, volume 2 of *LIPIcs*, pages 25–36. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2008.

[6] J. Blömer and S. Naewe. Sampling Methods for Shortest Vectors, Closest Vectors and Successive Minima. *Journal of Theoretical Computer Science*, volume 410, issue 18, pages 1648–1665, 2009.

[7] Y. Chen, and N. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *Proceedings of the 19th Annual International Conference on Theory and Application of Cryptology and Information Security.*, ASIACRYPT'11, volume 7073 of *LNCS*, pages 1-20. Springer, 2011.

[8] S. Garg, C. Gentry, and S. Halevi. Candidate Multilinear Maps from Ideal Lattices. Cryptology ePrint Archive, Report 2012/610, 2012.

[9] N. Gama, P. Nguyen, and O. Regev. Lattice Enumeration Using Extreme Pruning. In *Proceedings of the 29th Annual International Conference on Theory and Application of Cryptographic Techniques*, Eurocrypt'10, volume 6110 of *LNCS*, pages 257–278. Springer, 2010.

[10] C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In *Proc of the 41st Annual ACM Symposium on Theory of Computing*, STOC'09, pages 169–178. ACM, 2009.

[11] J. Hoffstein, J. Pipher, and J. Silverman. NTRU: A Ring-based Public Key Cryptosystem. In *Algorithmic Number Theory*, volume 1423 of *LNCS*, pages 267–288. Springer, 1998.

[12] A. Lenstra, H. Lenstra, and L. Lovász. Factoring Polynomials with Rational Coefficients. *Journal of Mathematische Annalen*, volume 261, issue 4, pages 515–534, 1982.

[13] J. Buchmann and C. Ludwig. Practical Lattice Basis Sampling Reduction. In *Proceedings of the 7th International Symposium*, ANTS-VII, volume 4076 of *LNCS*, pages 222–237, Springer, 2006.

[14] T. Ishiguro, S. Kiyomoto, Y. Miyake and T. Takagi. Parallel Gauss Sieve Algorithm: Solving the SVP Challenge over a 128-Dimensional Ideal Lattice. Cryptology ePrint Archive, Report 2013/388, 2013.

[15] 柏原 賢二. 格子の最短ベクトル問題の新しいアルゴリズム. 第 5 回暗号及び情報セキュリティと数学の相関ワークショップ, CRISMATH2013, 2013. http://www.risec.aist.go.jp/events/2013/1226-ja.html.

[16] S. Khot. Hardness of Approximating the Shortest Vector Problem in Lattices, *Journal of the ACM*, Vol. 52, No. 5, pages 789–808, Springer, 2005.

[17] S. Khot. Inapproximability Results for Computational Problems on Lattices, *Information Security and Cryptography - The LLL Algorithm*, pages 453–473, Springer, 2010.

[18] D. Micciancio. The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, FOCS'98, pages 92–98. IEEE Computer Society, 1998.

[19] D. Micciancio and P. Voulgaris. A Deterministic Single Exponential Time Algorithm for Most Lattice Problems Based on Voronoi Cell Computations. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC'10, pages 351–358. ACM, 2010.

[20] D. Micciancio and P. Voulgaris. Faster Exponential Time Algorithms for the Shortest Vector Problem. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'10, volume 65, pages 1468–1480. SIAM, 2010.

[21] P. Q. Nguyen and T. Vidick. Floating-point LLL Revisited. In *Proceedings of the 24th Annual Eurocrypt Conference*, volume 3495 of *LNCS*, pages 215–233, Springer, 2005.

[22] P. Q. Nguyen and T. Vidick. LLL on the Average. In *Proceedings of the 7th International Symposium*, ANTS-VII, volume 4076 of *LNCS*, pages 238–256, Springer, 2006.

[23] P. Q. Nguyen and D. Stehlé. Sieve Algorithms for the Shortest Vector Problem Are Practical. *Journal of Mathematical Cryptology*, volume 2, pages 181–207, 2008.

[24] T. Plantard and M. Schneider. Ideal Lattice Challenge. http://www.latticechallenge.org/ideallattice-challenge/.

[25] T. Plantard and M. Schneider. Creating a Challenge for Ideal Lattices. Cryptology ePrint Archive, Report 2013/039, 2013.

[26] X. Pujol and D. Stehlé. Solving the Shortest Lattice Vector Problem in Time $2^{2.465n}$. Cryptology ePrint Archive, Report 2009/605, 2009.

[27] M. Schneider. *Computing Shortest Lattice Vectors on Special Hardware*. PhD thesis, Technische Universität Darmstadt, 2011.

[28] M. Schneider. Sieving for Shortest Vectors in Ideal Lattices. Cryptology ePrint Archive, Report 2011/458, 2011.

[29] R. Lindner, M. Rueckert, P. Baumann and L. Nobach. Lattice Challenge. http://www.latticechallenge.org/.

[30] M. Schneider and N. Gama. The SVP Challenge. http://www.latticechallenge.org/svp-challenge/.

[31] C.-P. Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Journal of Theoretical Computer Science*, volume 53, issue 2-3, pages 201–224, 1987.

[32] C.-P. Schnorr. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. *Journal of Mathematical programming*, pages 181–191. Springer, 1993.

[33] C.-P. Schnorr and H. H. Horner. Attacking the Chor-Rivest Cryptosystem by Improved Lattice Reduction. In *Proceedings of the 14th annual international conference on Theory and application of cryptographic techniques*, Eurocrypt'95, pages 1–12. Springer, 1995.

[34] C.-P. Schnorr. Lattice reduction by random sampling and birthday methods. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, STACS'03, pages 145–156. Springer, 1995.

[35] X. Wang, M. Liu, C. Tian and J. Bi. Improved Nguyen-Vidick Heuristic Sieve Algorithm for Shortest Vector Problem. Cryptology ePrint Archive, Report 2010/647, 2010.

第 3 章

LWE

3.1 LWE の概説

近年, 2005 年に Regev[Reg05] によって紹介された LWE (Learning with Errors) 問題の計算量困難性に依存した暗号技術がこれまで数多く提案されている. ここでは, 主に LWE 問題を用いた様々な暗号技術へのアプリケーションの紹介と, LWE 問題の計算量困難性についてまとめる (本章をまとめるにあたり, 文献 [Reg] を主に参考にした).

3.1.1 LWE とは

LWE 問題とは, Machine Learning (機械学習理論) から派生した, 解くことが難しいとされている問題の一種である. 簡単に説明すると, 秘密情報 $\vec{s} \in \mathbb{F}_q^n$ に関するランダムな線形 “近似値” の列が与えられたときに, その秘密情報 $\vec{s}$ を復元する問題のことをいう. 具体的な数値例として, 変数 $\vec{s} = (s_1, s_2, s_3, s_4)$ に関する線形近似値の列

$$\begin{cases} 14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \pmod{17} \\ 13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \pmod{17} \\ 6s_1 + 10s_2 + 13s_3 + s_4 \approx 12 \pmod{17} \\ 10s_1 + 4s_2 + 12s_3 + 16s_4 \approx 12 \pmod{17} \\ 9s_1 + 5s_2 + 9s_3 + 6s_4 \approx 9 \pmod{17} \\ 3s_1 + 6s_2 + 4s_3 + 5s_4 \approx 16 \pmod{17} \\ \vdots \\ 6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \pmod{17} \end{cases}$$

が与えられたとする (ただし, 各線形方程式の誤差は ± 1 程度とする). このとき, 上記の方程式の列の解 $\vec{s} = (s_1, s_2, s_3, s_4)$ を求めるのが LWE 問題の例である (実際, 上記の数値例では, $\vec{s} = (0, 13, 9, 11) \in \mathbb{F}_{17}^4$ が解となる). ここで注意しておかなくてはいけない事は, 上記の線形方程式で誤差がない場合は, ガウスの消去法 (または掃出し法ともいう) を用いれば多項式時間で簡単に解を求めることができる点である. つまり, 与えられる誤差の度合いが LWE 問題をより難しくしている. ここで, LWE 問題の定義を与えておく.

定義 3.1 (LWE 問題 [Reg05]) サイズパラメータ $n \geq 1$, 剰余パラメータ $q \geq 2$, $\mathbb{F}_q$ 上の誤差に関する確率分布 χ が与えらたとする. このとき, $A_{\vec{s},\chi}$ を

$$A_{\vec{s},\chi} = \left\{ (\vec{a}, \langle \vec{a}, \vec{s} \rangle + e) \in \mathbb{F}_q^n \times \mathbb{F}_q \mid \vec{a} \leftarrow \mathbb{F}_q^n, e \leftarrow \chi \right\}$$

で定義される確率分布とする (ただし, $\vec{a}$ は $\mathbb{F}_q^n$ 上一様ランダムに選ばれた元とし, $\langle \vec{a}, \vec{s} \rangle$ は 2 つのベクトル間の内積値とする). 秘密情報 $\vec{s} \in \mathbb{F}_q^n$ に対し, $A_{\vec{s},\chi}$ からサンプリングされた任意個数の元が与えられた時に, 秘密情報 $\vec{s}$ を求める問題を LWE 問題という.

上記で定義した LWE 問題は, ランダム線形符号の復号問題, または, 格子上ランダムな bounded distance decoding(BDD) 問題として見なすことができる. さらに, $q = 2$ のとき, LWE 問題は learning parity with noise (LPN) 問題に対応する (LPN 問題については, 第 4 章で説明). 上記の定義において, 確率分布 χ としてガウス分布を用いる場合がほとんどであったが, 近年では一様分布を用いた場合の研究も進み始めている [DQ13, MP13].

またこの節で, 上記で定義した LWE の変形問題である ring-LWE 問題も紹介しておく (以下の定義では, 2 べき整数 n の場合しか説明しないが, 近年では一般の整数 n を用いた ring-LWE 問題も紹介され, 色々な暗号方式を構成する際に応用されている. 参考文献として [LPR13] を参照することを勧める).

定義 3.2 (ring-LWE 問題 [LPR10]) n を 2 べき整数とし, q を $q \equiv 1 \bmod 2n$ を満たす素数とする. また, R_q を環 $\mathbb{F}_q[x]/(x^n + 1)$ と定め, R_q 上の誤差に関する確率分布 χ を固定しておく. ただし, 写像 $a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \mapsto (a_0, a_1, \ldots, a_{n-1})$ より環 R_q は n-次元ベクトル空間 $\mathbb{F}_q^n$ と同一視することができ, R_q の元を $\mathbb{F}_q^n$ の元として見なすことができる. ring-LWE 問題では, $\vec{a} \bullet \vec{s}$ を R_q 上の乗算とした時, 秘密情報 $\vec{s} \in R_q \simeq \mathbb{F}_q^n$ に対して, 集合

$$\left\{ (\vec{a}, \vec{b} = \vec{a} \bullet \vec{s} + \vec{e}) \in R_q \times R_q \mid \vec{a} \leftarrow R_q, \vec{e} \leftarrow \chi \right\}$$

からサンプリングされた元たちが与えられた時に, 秘密情報 $\vec{s}$ を求める問題を ring-LWE 問題と呼ぶ.

通常の LWE 問題に比べて, ring-LWE 問題は格子ベースの暗号スキームをより効率的にすることができ, 近年では ring-LWE 問題をベースとした (主に準同型) 暗号スキームが数多く提案されている.

3.1.2 LWE の一般的な利点 (アプリケーション)

一般的に, LWE 問題は暗号技術の様々な分野に応用することが可能で, これまでに様々な研究者によって提案されている. 代表的な応用例として以下のものが知られている.

- 公開鍵暗号スキームの構成
  - 選択平文攻撃に対して安全な方式 [Reg05, KTX07, PVW08]
  - 選択暗号文攻撃に対して安全な方式 [PW08, Pei09]
- 紛失通信プロトコル [PVW08]
- identity-based encryption (IBE) スキームの構成 [GPV08, CHKP10, ABB10]
- leakage-resilient 暗号の構成 [AGV09, ACPS09, DGK10, GKPV10]

さらに, 2009 年の Gentry[Gen09] の完全準同型暗号の構成に関する結果以降では, 特に ring-LWE 問題ベースの (完全 or somewhat) 準同型暗号スキームが数多く提案されており, 代表的な完全準同型暗号スキームに関するものとして, [SV11, BGV12, GHS12a, GHS12b, GHPS12] の結果が知られている.

3.1.3 代表的な LWE ベースの暗号方式

ここでは, LWE 問題をベースとした代表的な暗号方式をいくつか紹介する.

3.1.3.1 [Reg05] による公開鍵暗号方式

LWE 問題をベースとした公開鍵暗号として, [Reg05] で提案された方式が代表的である. [Reg05] の暗号方式の構成のためには, 以下の 4 つのパラメータが必要である:

- n: 安全性パラメータ
- m: LWE サンプルの個数 ($m = 1.1 \cdot n \log q$ となる整数を選ぶ)
- q: 剰余パラメータ (q として $n^2 \leq q \leq 2n^2$ を満たす素数を選ぶ)
- $\alpha > 0$:ノイズパラメータ ($\alpha = 1/\left(\sqrt{n}\log^2 n\right)$)

以下に具体的な暗号方式の構成を示す:

秘密鍵の生成 一様ランダムに $\vec{s} \leftarrow \mathbb{F}_q^n$ を選ぶ.

公開鍵の生成 秘密鍵 $\vec{s}$, 剰余パラメータ q, ノイズパラメータ α を持つ LWE 分布から生成した m 個のサンプル $(\vec{a}_i, b_i)_{i=1}^m \leftarrow A_{\vec{s},\chi}^m$ を公開鍵とする (つまり各 i に対し, $\vec{a} \leftarrow \mathbb{F}_q^n$ で $e_i \leftarrow \chi = D_{\mathbb{Z},\alpha q}$ と選び, $b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i \in \mathbb{F}_q$ と構成する).

暗号化 集合 S を $\{1, 2, \ldots, m\}$ の中から一様ランダムに選んだ部分集合とする (例えば, $S = \{1, m\}$). このとき, 平文ビットが 0 の暗号文を $\left(\sum_{i\in S} \vec{a}_i, \sum_{i\in S} b_i\right)$ とし, 平文ビットが 1 の暗号文を $\left(\sum_{i\in S} \vec{a}_i, \lfloor \frac{q}{2} \rfloor + \sum_{i\in S} b_i\right)$ とする.

復号 暗号文 $(\vec{a}, b)$ に対し, $b - \langle \vec{a}, \vec{s} \rangle \in \mathbb{F}_q$ が $\lfloor \frac{q}{2} \rfloor$ より 0 に近い場合, 復号結果として 0 を出力し, それ以外の場合は 1 を出力する.

復号の正当性について, $(\vec{a}, b) = \left(\sum_{i\in S} \vec{a}_i, \sum_{i\in S} b_i\right)$ の場合 (つまり, 平文 0 に対応する暗号文の場合),

$$b - \langle \vec{a}, \vec{s} \rangle = \sum_{i\in S} (b_i - \langle \vec{a}_i, \vec{s} \rangle) = \sum_{i\in S} e_i$$

なので, $-\frac{q}{4} < \sum_{i\in S} e_i < \frac{q}{4}$ であれば復号に成功する (つまり, 復号として 0 が出力される). 各ノイズ e_i は標準偏差が αq のガウス分布 $\chi = D_{\mathbb{Z},\alpha q}$ から選ばれているので, $\sum_{i\in S} e_i$ の標準偏差は高々 $\sqrt{m}\alpha q$ となる. ここで, 各パラメータの選択方法から $\sqrt{m}\alpha q < q/\log n$ なので, 非常に高い確率で復号に成功することが分かる (平文ビットが 1 の暗号文に対しても同様の議論が成り立つ). また, 上記の暗号方式の安全性については, LWE 仮定の下で CPA 安全であることが証明されている [Reg09, Section 5].

ここで紹介した [Reg05] による暗号方式は, 公開鍵サイズが $(mn \log q) = \widetilde{O}(n^2)$ で, 暗号文サイズも平文サイズの $O(n \log q) = \widetilde{O}(n)$ 倍に増加するため, 決して効率的ではない (より効率的な方式としては [PVW08] などを参照).

■パラメータ設定について 上記で構成した [Reg05] による公開鍵暗号方式の具体的なパラメータ設定例が [MR09] で示されている. パラメータ設定例として, $(n, m, q, \alpha) = (136, 2008, 2003, 0.0065)$, $(192, 1500, 16381, 0.0009959)$, $(233, 1042, 32749, 0.000217)$ などが挙げられており, これらの各パラメータ設定は格子ベース暗号の安全性を測る root Hermite factor δ の値が 1.01 程度になるように設定されている (root Hermite factor δ については後述の 3.2.2 節を参照).

3.1.3.2 [BV11] による somewhat 準同型暗号方式 ([LNV11] で少し改良)

近年, 効率的な LWE ベースの暗号方式を得るために, [LPR10] で紹介されている ring-LWE 問題 (定義 3.5 を参照) の困難性に依存した方式がいくつか提案されている. 以下では, [BV11] で提案されている somewhat 準同型暗号方式を紹介する (somewhat 準同型暗号とは暗号化したまま限定回の加算と乗算が可能な暗号方式). [BV11] の somewhat

準同型暗号方式の構成のために, 以下の 4 つのパラメータが必要である:

- n: 2 べき整数で, 暗号方式を構成する基礎的な環 $R = \mathbb{Z}[x]/(x^n + 1)$ を定義する (n が 2 べき整数の場合のみ, 多項式 $x^n + 1$ は $\mathbb{Z}$ 上既約となることに注意).
- q: $q \equiv 1 \bmod 2n$ を満たす素数で, 暗号文空間の基礎環 $R_q = \mathbb{F}_q[x]/(x^n + 1)$ を定義する.
- t: 条件 $t < q$ を満たす整数で, 暗号方式の平文空間 $R_t = (\mathbb{Z}/t\mathbb{Z})[x]/(x^n + 1)$ を定義する.
- σ: ノイズを与えるためのガウス分布の標準偏差.

そこで, [BV11] の somewhat 準同型暗号方式は以下のように構成される (少しだけ改良された方式として [LNV11] も参照): また, 以下の構成では, 定義 3.5 と同じように $a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \to (a_0, a_1, \ldots, a_{n-1})$ より環 R を $\mathbb{Z}^n$ と同一視する (同様に, $R_q \simeq \mathbb{F}_q^n$ と同一視することが可能).

鍵生成 まず, $R \ni s \leftarrow \chi = D_{\mathbb{Z}^n,\sigma}$ を選び, 一様ランダムに $p_1 \in R_q$ を取り, 小さなエラー $e \leftarrow \chi$ を固定する ([BV11] では $s \leftarrow \chi$ を一様ランダムに選択するのに対し, [LNV11] では一様ランダムには選択しない点だけが異なる). そこで, 公開鍵を $\mathsf{pk} = (p_0, p_1)$ とし (ただし, $p_0 = -(p_1 s + te)$ とする), 秘密鍵を $\mathsf{sk} = s$ とする.

暗号化 平文情報 $m \in R_t$ と公開鍵 $\mathsf{pk} = (p_0, p_1)$ に対し, まず $R \ni u, f, g \leftarrow \chi$ を選び, 暗号文を

$$\mathsf{Enc}(m, \mathsf{pk}) = (c_0, c_1) = (p_0 u + tg + m, p_1 u + tf),$$

と定義する. ただし, 条件 $t < q$ より, 上記の数式では元 $m \in R_t$ を環 R_q の元として見なして計算する. つまり, 上記の暗号文は $(R_q)^2$ の元として表現される.

準同型暗号演算 (暗号加算・暗号乗算) 上記の暗号アルゴリズムでは暗号文として $(R_q)^2$ の元を出力するが, 以下で定義する暗号乗算では暗号文の長さを長くする操作であるため, ここでは任意の長さの暗号文に対する暗号加算・乗算を定義する; 2 つの暗号文 $\mathsf{ct} = (c_0, c_1, \ldots, c_\xi)$ and $\mathsf{ct}' = (c'_0, c'_1, \ldots, c'_\eta)$ が与えられているとする.

- まず, 暗号加算 "$\dot{+}$" は, 以下のように各成分ごとの加算

$$\mathsf{ct} \dot{+} \mathsf{ct}' = (c_0 + c'_0, \ldots, c_{\max(\xi,\eta)} + c'_{\max(\xi,\eta)})$$

  で与えられる. 同様に, 暗号減算も各成分ごとの減算で与えられる.

- 次に, 暗号乗算 "$*$" は以下で与えられる:

$$\mathsf{ct} * \mathsf{ct}' = (\hat{c}_0, \hat{c}_1, \ldots, \hat{c}_{\xi+\eta})$$

  ただし, z を変数としたとき, 各 $\hat{c}_i$ は以下の関係式から計算可能である:

$$\sum_{i=0}^{\xi+\eta} \hat{c}_i z^i = \left(\sum_{i=0}^{\xi} c_i z^i\right) \cdot \left(\sum_{j=0}^{\eta} c'_j z^j\right)$$

復号 任意の長さの暗号文 $\mathsf{ct} = (c_0, c_1, \ldots, c_\xi)$ に対して, 復号は

$$\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\tilde{m}]_q \bmod t \in R_t,$$

で計算できる. ただし, $\tilde{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$ であり, $[\tilde{m}]_q$ は元 $\tilde{m}$ の各係数の $[-q/2, q/2)$ への剰余とする. また, $\vec{s} = (1, s, s^2, \ldots)$ としたとき, この復号処理を $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\langle \mathsf{ct}, \vec{s} \rangle]_q \bmod t$ と書き直すこともできる.

復号の正当性については, 上記の暗号アルゴリズムで得られる暗号文 $\mathsf{ct} = (c_0, c_1)$ に対し, 関係式 $p_0 + p_1 s = -te$ が成り立つので

$$\langle \mathsf{ct}, \vec{s} \rangle = (p_0 u + tg + m) + s \cdot (p_1 u + tf) = m + t \cdot (g + sf - ue)$$

表 3.1　[BV11] による somewhat 準同型暗号方式のパラメータ設定例とその安全性レベル (詳細は [LNV11, Table 1] を参照, δ は各パラメータに対する root Hermite factor で詳細は 3.2.2 節を参照)

パラメータ (n, q, t, σ)	暗号乗算の深さ	distinguishing attack の攻撃計算量
$(2048, 52\text{-bit}, 128, 8)$	1	2^{198} $(\delta = 1.0041)$
$(4096, 86\text{-bit}, 128, 8)$	2	2^{250} $(\delta = 1.0035)$
$(4096, 118\text{-bit}, 128, 8)$	3	2^{149} $(\delta = 1.0048)$
$(4096, 150\text{-bit}, 128, 8)$	4	2^{92} $(\delta = 1.0062)$
$(16384, 338\text{-bit}, 128, 8)$	9	2^{243} $(\delta = 1.0035)$

が環 R_q 上で成り立つ. ここで, 元 $m + t \cdot (g + sf - ue)$ を環 R の元と見なしたとき, その各係数が $[-q/2, q/2)$ 内に収まっている限り, $[\langle \mathsf{ct}, \vec{s} \rangle]_q = m + t \cdot (g + sf - ue)$ が環 R 上で成立する (元 $e, f, g, u \leftarrow \chi$ が十分小さなノイズとして選択されていることに注意). この場合, 剰余 mod$t$ の操作で正しい復号結果 $m \in R_t$ が得られる. また, 暗号加算・暗号乗算された暗号文について, 2 つの暗号文 $\mathsf{ct}_1, \mathsf{ct}_2$ に対し,

$$\begin{cases} \langle \mathsf{ct}_1 \dot{+} \mathsf{ct}_2, \vec{s} \rangle = \langle \mathsf{ct}_1, \vec{s} \rangle + \langle \mathsf{ct}_2, \vec{s} \rangle \\ \langle \mathsf{ct}_1 * \mathsf{ct}_2, \vec{s} \rangle = \langle \mathsf{ct}_1, \vec{s} \rangle \cdot \langle \mathsf{ct}_2, \vec{s} \rangle \end{cases}$$

が成り立つので, 暗号文のノイズが十分小さい限り, 準同型演算が可能な暗号方式となっている. 具体的には, 暗号文 $\mathsf{ct}_1, \mathsf{ct}_2$ が平文情報 $m_1, m_2 \in R_t$ に対応しているとき, 各暗号文のノイズが小さい場合に限り

$$\begin{cases} \mathsf{Dec}(\mathsf{ct}_1 \dot{+} \mathsf{ct}_2, \mathsf{sk}) = m_1 + m_2 \\ \mathsf{Dec}(\mathsf{ct}_1 * \mathsf{ct}_2, \mathsf{sk}) = m_1 \times m_2 \end{cases}$$

が成立する.

また, この暗号方式の安全性については, 定義 3.5 で与えられた ring-LWE 問題を少し変形した以下の問題の計算量困難性に依存する (以下は [LNV11] を引用):

定義 3.3 (polynomial-LWE 問題 [BV11], [LNV11]) パラメータ (n, q, t, σ) が与えられた時, polynomial-LWE 問題 $\mathsf{PLWE}_{n,q,\chi}$ とは, 次の 2 つの分布を識別することである:

1. 一様ランダムに $(R_q)^2$ の元 (a_i, b_i) をサンプリングする.
2. 一様ランダムに $s \leftarrow \chi = D_{\mathbb{Z}^n, \sigma}$ を選び, 一様ランダムに $a_i \leftarrow R_q$ をサンプリングし, $e_i \leftarrow \chi$ を選び $b_i = a_i s + e_i$ とする. このとき, $(a_i, b_i) \in (R_q)^2$ をサンプリングする.

上記で構成した somewhat 準同型暗号方式の安全性については, 具体的には上記の polynomial-LWE 問題の計算量困難性仮定の下で KDM 安全 (key dependent message security) であることが証明されている [BV11].

■パラメータ設定について　上記で構成される somewhat 準同型暗号方式に関して, [LNV11, Table 1] で具体的なパラメータ設定例が挙げられている. 表 3.1 で,　[LNV11, Table 1] の中で代表的なパラメータ設定例を示すと共に、そのパラメータ設定に対する distinguishing attack による攻撃計算量の見積もりも示しておく. また, distinguishing attack による攻撃原理とその攻撃計算量評価については, 後述の 3.2.2 節で説明する (具体的には, 表 3.1 の distinguishing attack の攻撃計算量は式 (3.3) から算出した値である).

3.2 LWE 問題の困難性について

ここでは, LWE 問題の困難性に簡単について説明する. ここでは, 他の格子問題への帰着という理論的な困難性に関するものと, 実際の攻撃実験による困難性評価に関するものの 2 つの面による結果を説明する.

3.2.1 他の格子問題への帰着とその困難性

文献 [Reg] でも説明されているように, 以下に挙げる 3 つの理由から現在 LWE 問題を解くことは難しいと信じられている.

(A) まず, LWE 問題を解く, 知られているものの中で最良のアルゴリズムは指数時間アルゴリズムである (量子アルゴリズムを用いた場合でさえも難しい).

(B) §3.1.1 で説明したように, LWE 問題は LPN 問題の一般化であり, LPN 問題自体が格子理論において解くのが困難な問題と予想されている. さらに, LPN 問題はランダム線形バイナリ符号の復号問題として定式化可能であり, LPN 問題を効率的に解くこと自体符号理論におけるブレークスルーである (LPN 問題については, 第 4 章を参照).

(C) さらに最も重要なこととして, GapSVP (the decision version of the shortest vector problem) や SIVP (the shortest independent vectors problem) のような標準的な格子問題の最悪ケースの困難性に関するある仮定のもとで, LWE 問題は困難であることが知られている [Reg05, Pei09].

ここで, 上記の (A) と (C) の点について具体的に説明した定理を挙げておく.

定理 3.4 ([Reg09] における Theorem 1.1) n, q を 2 つの整数とし, $\alpha \in (0,1)$ は $\alpha q > 2\sqrt{n}$ を満たすとする. もし $\mathsf{LWE}_{n,q,\bar{\Phi}_\alpha}$(§3.2.2 の定義 3.5 を参照) を解く効率的なアルゴリズムが存在するなら, 最悪時の因子 $\gamma = \tilde{O}(n/\alpha)$ を持つ GapSVP_γ と SIVP_γ を効率的に解くことができる量子アルゴリズムが存在する. ただし, Φ_α は平均値が 0 で標準偏差が $\frac{\alpha}{\sqrt{2\pi}}$ を持つ確率分布で, $\bar{\Phi}_\alpha$ は Φ_α を離散化した確率分布とする.

別の言い方をすると, 上記の定理は GapSVP と SIVP を効率的に解く量子アルゴリズムが存在しないなら, LWE 問題を効率的に解くアルゴリズムは存在しないことを示している. また一方で, 任意の多項式因子 γ を持つ GapSVP_γ と SIVP_γ を解く多項式時間を持つ量子アルゴリズム [NC00] は存在しないと予想されており, このことから LWE 問題を解くことは困難であると予想されている.

ちなみに GapSVP_γ 問題とは, n 次元格子 L と与えらえた値 $d > 0$ に対し, $\lambda_1(L)$ を各々 L の最小ベクトルの長さ, $\lambda_n(L)$ を n 個の一次独立なベクトル集合に含まれる最大ベクトル長の最小値, $\gamma = \gamma(n)$ を 1 以上の近似因子として, $\lambda_1(L) \leq d$ なら Yes を, $\lambda_1(L) > \gamma(n)d$ なら No を返す問題であり, SIVP_γ とは, 同じく L に対して, 長さ $\gamma(n) \cdot \lambda_n(L)$ 以下の n 個の一次独立なベクトルを求める問題である.

その他, 安全性証明に関連する結果として, 文献 [LMSV12] では, ring-LWE 問題をベースとした Somewhat Homomorphic Encryption スキーム (演算回数に制約がある準同型暗号スキームで, 完全準同型暗号スキームの構成要素) が IND-CCA1 を満たすことが示されている.

3.2.2 LWE 問題の困難性の実験評価

Lindner と Peikert [LP11] は, LWE 問題の困難性について NTL ライブラリ (具体的には, NTL ライブラリ内の BKZ アルゴリズムを利用) を用いて実際の攻撃実験を行い, その困難性評価指標を定めている. ここでは, 彼らの困難性評価指標について, 簡単にまとめておく. まず, 彼らが評価対象とした decision version の LWE 問題を以下で正確に定義する.

定義 3.5 (decision version, $\mathsf{LWE}_{n,q,\chi}$) 定義 3.1 で与えたように, $n \geq 1$ と $q \geq 2$ と, $\mathbb{F}_q$ 上の確率分布 χ を考える (ただし, 文献 [LP11] では, 確率分布 χ は $\mathbb{Z}$ 上の標準偏差 σ を持つ離散ガウス分布 $D_{\mathbb{Z},\sigma}$ から生成されたものにしている). このとき, 秘密情報 $\vec{s} \in \mathbb{F}_q^n$ に対し, 定義 3.1 で紹介した $A_{\vec{s},\chi}$ からランダムにサンプリングされた元 $(\vec{a}, \langle \vec{a}, \vec{s} \rangle + e)$ と, $\mathbb{F}_q^n \times \mathbb{F}_q$ 上の一様分布で得られる元とを区別する問題を $\mathsf{LWE}_{n,q,\chi}$ と定義する.

上記で定義した $\mathsf{LWE}_{n,q,\chi}$ 問題に対して, 文献 [LP11] で Lindner-Peikert は 2 つの効率的な攻撃手法を紹介している.

- distinguishing attack (Micciancio-Regev[MR07] が提案)
- decoding attack (Lindner-Peikert 自身が文献 [LP11] で提案)

文献 [LP11] によると, decoding attack よりも distinguishing attack の方が常に効率的であるが, 実際の攻撃評価結果 [LP11, Figure 4 in Section 6] を比べてみると, $\varepsilon = 2^{-32}$ または $\varepsilon = 2^{-64}$ 程度の実用的なレベルの advantage を想定した場合には, 上記 2 つの攻撃の効率性は同程度であったという結果を得たとのこと.

■Distinguishing attack による攻撃原理 そこで, 以下では $\mathsf{LWE}_{n,q,\chi}$ 問題に対する distinguishing attack の攻撃原理を少し紹介しておく. 秘密情報 $\vec{s} \in \mathbb{F}_q^n$ に対し, 集合 $A_{\vec{s},\chi}$ からランダムにサンプリングされた元

$$\vec{a}_i \in \mathbb{F}_q^n,\ b_i = \langle \vec{a}_i, \vec{s} \rangle + e_i \in \mathbb{F}_q \tag{3.1}$$

を数多く (ここでは m 個) 集めることで, 以下の情報を得ることができる (ここでは, すべてのベクトルは n-次元の行ベクトルで表記したとする):

$$\mathbf{A} = (\vec{a}_1^T, \vec{a}_2^T, \ldots, \vec{a}_m^T) \in \mathbb{F}_q^{n \times m}, \vec{b} = (b_1, b_2, \ldots, b_m) \in \mathbb{F}_q^m, \vec{e} = (e_1, e_2, \ldots, e_m) \in \mathbb{Z}^m$$

すると, 上記の記法を用いると, 関係式 (3.1) から

$$\vec{b} = \vec{s} \cdot \mathbf{A} + \vec{e} \pmod{q}$$

という関係式を得ることができる. そこで, $\mathbb{F}_q^n \times \mathbb{F}_q$ 上の一様分布で得られる元と区別するために, 攻撃者はまず (scaled な) 双対格子

$$\Lambda^{\perp}(\mathbf{A}) := \left\{ \vec{v} \in \mathbb{Z}^m \mid \vec{v} \cdot \mathbf{A}^T \equiv 0 \pmod{q} \right\}$$

の最短ベクトル $\vec{v} \neq \vec{0} \in \mathbb{Z}^m$ を見つけたとする. ここで, その攻撃者は内積値 $\langle \vec{v}, \vec{b} \rangle \pmod{q}$ が 0 に十分近いかどうかで $\mathbb{F}_q^n \times \mathbb{F}_q$ 上一様分布にサンプリングされた元かどうか判定することができる. その理由は, ベクトル $\vec{v}$ は $\vec{v} \cdot \mathbf{A}^T \equiv 0 \pmod{q}$ を満たすので,

$$\langle \vec{v}, \vec{b} \rangle \equiv \langle \vec{v}, \vec{s} \cdot \mathbf{A} + \vec{e} \rangle \equiv \langle \vec{v}, \vec{s} \cdot \mathbf{A} \rangle + \langle \vec{v}, \vec{e} \rangle \equiv \langle \vec{v}, \vec{e} \rangle \pmod{q}$$

となる. さらに, ベクトル $\vec{e} \in \mathbb{Z}$ の各成分 e_i は $\chi = D_{\mathbb{Z},\sigma}$ からサンプリングされた元なので, そのサイズは σ 程度となり, 上記の内積値 $\langle \vec{v}, \vec{b} \rangle$ のサイズはおよそ $\sigma \cdot ||\vec{v}||$ 程度となることが分かる. よって, 攻撃者は十分小さなベクトル $\vec{v} \in \Lambda^{\perp}(\mathbf{A})$ を見つけることができた場合, 上記の内積値の小ささを測ることで, $\mathbb{F}_q^n \times \mathbb{F}_q$ 上一様にサンプリングされた元か $A_{\vec{s},\chi}$ からサンプリングされた元か区別することができる.

表 3.2　$\log_2(T_{\mathrm{BKZ}})$ と δ_{BKZ} の関係 [DPSZ12, Appendix D]

$\log_2(T_{\mathrm{BKZ}})$	80	100	128	192	256
δ_{BKZ}	1.0066	1.0059	1.0052	1.0041	1.0034

■Distinguishing attack に対する解読計算量評価　さらに, 文献 [MR07] によると, advantage ε を持つ攻撃者は双対格子 $\Lambda^{\perp}(\mathbf{A})$ から長さ $c \cdot q/\sigma$ を持つ格子元を見つけることができた場合, distinguishing attack を成功することができると示している (詳細は, [LP11, Section 6] を参照). ただし, $c \approx \sqrt{\log_2(1/\varepsilon)/\pi}$ とする. また一方, 格子縮約アルゴリズムはある格子の中からかなり短い格子元を出力するアルゴリズムで, その格子縮約アルゴリズムがどのくらい短い格子元を出力することが可能かを図る指標として, *root Hermite factor* という指標がよく用いられる (root Hermite factor の説明については, [GN08] を参照). d-次元の格子 L に対して,

$$\delta := \left(\frac{||\vec{b}_1||}{|\det(L)|^{1/d}} \right)^{1/d}$$

の値を格子縮約アルゴリズムの root Hermite factor と呼ぶ. ただし, 格子縮約アルゴリズムを出力される格子基底を $\{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_d\}$ とし, その長さを $||\vec{b}_i||$ と表す (さらに, $||\vec{b}_1|| \leq ||\vec{b}_2|| \leq \cdots$ と仮定). そこで, distinguishing attack を用いて, $\mathsf{LWE}_{n,q,\chi}$ 問題を解くためには, 攻撃に利用する格子縮約アルゴリズムの root Hermite factor δ は

$$c \cdot q/\sigma = \delta^m \cdot |\det(\Lambda^{\perp}(\mathbf{A}))|^{1/m} = \delta^m \cdot q^{n/m}$$

の条件を満たす必要がある. さらに, distinguishing attack に最適な格子次元 $m = \sqrt{n \log_2(q)/\log_2(\delta)}$ を想定した場合, 上記の関係式から

$$c \cdot q/\sigma = 2^{2\sqrt{n \log_2(q) \log_2(\delta)}} \tag{3.2}$$

という n, q, σ の関係式を新たに得ることができる.

一方, BKZ アルゴリズムは効率的な格子縮約アルゴリズムであることが知られている. そこで, Lindner-Peikert [LP11] は NTL ライブラリで実装済みの BKZ アルゴリズムを利用した場合の distinguishing attack の計算量 T_{BKZ} に対して,

$$\log_2(T_{\mathrm{BKZ}}) = \frac{1.8}{\log_2(\delta_{\mathrm{BKZ}})} - 110 \tag{3.3}$$

という見積もり値を示している. ただし, ここでの δ_{BKZ} は BKZ アルゴリズムの root Hermite factor で, その指標値は BKZ アルゴリズムのブロックサイズに関するパラメータにより定まる (ブロックサイズが大きくなるほど root Hermite factor は小さくなるため, distinguishing attack の計算量 T_{BKZ} は増大する). 表 3.2[*1] に, 文献 [DPSZ12, Appendix D] で示されている T_{BKZ} と δ_{BKZ} の関係式を示した表を紹介しておく. 表 3.2 から分かることは, BKZ アルゴリズムを利用した攻撃に対して $\mathsf{LWE}_{n,q,\chi}$ 問題のセキュリティレベルを 80-bit 程度以上に保つためには, root Hermite factor $\delta = 1.0066$ に対し, 関係式 (3.2) を満たすように $n, q, \chi = D_{\mathbb{Z},\sigma}$ のパラメータを選択する必要があることを示している. しかし, Lindner-Peikert による見積もり攻撃評価 (3.3) は, NTL ライブラリ実装による BKZ アルゴリズムに関するもので, すでに最新の実装結果ではないことに注意. 現在知られている BKZ アルゴリズムは, Chen-Nguyen ら [CN11] が実装した BKZ 2.0 というアルゴリズムが代表的で, 彼ら自身のアルゴリズム評価によると, 80-bit セキュリティを得るためには, BKZ アルゴリズムの root Hermite factor が 1.0050 程度以下を想定する必要があることを示している.

[*1] 表 3.2 における $\log_2(T_{\mathrm{BKZ}})$ の 192 は元論文では 196 と記載されているが, 誤植であろうと考えられる.

■近年の攻撃研究の動向　[BG14] では LWE 問題の特殊な場合に有効な攻撃手法を提案している．具体的には, 定義 3.1 で紹介した LWE 問題において, 秘密情報 $\vec{s} \in \mathbb{F}_q^n$ と取り方として, $\vec{s} \leftarrow \{-1, 0, 1\}^n$ と限定する binary-LWE 問題について考察している．この binary-LWE 問題に対して, [BG14] では節 3.2.2 で少し紹介した decoding attack をベースとした攻撃手法を提案している．より具体的には, binary-LWE 問題を inhomogeneous short integer solution(ISIS) 問題に帰着させて解く手法を示しており (ISIS 問題: $(\mathbf{A}, \vec{v})$ が与えられた時, $\vec{v} \equiv \mathbf{A}\vec{y} \pmod q$ を満たす短い整数ベクトル $\vec{y}$ を見つける問題), 通常の攻撃よりも非常に効率的であることを理論的かつ実験的にも示している．

3.2.3　アプリケーションのためのパラメータ設定について

LWE 問題を用いた暗号技術応用において, LWE 問題の困難性を十分保ちながら暗号プロトコルなどを正しく動作させるためのパラメータ設定は一般的にかなり難しい問題である．ここでは, これまで知られている LWE 問題におけるパラメータ設定の代表例を挙げておく：

- Lindner-Peikert らは, [LP11, Section 3] で Micciancio[Mic10] が概要を示した LWE 問題ベースの公開鍵暗号方式の具体的な構成方法を示し, さらに彼らは [LP11, Section 6] でその暗号方式に対する具体的なパラメータ例を [LP11, Figure 3] に示している．また近年では, 青野らは表 [ABPW13, Table 2] において [LP11] で挙げたパラメータの安全性を再評価する一方で, LWE ベースの proxy re-encryption(PRE) スキームの具体的なパラメータを [ABPW13, Table 1] で示し, その各パラメータの安全性を [ABPW13, Table 3] で評価している．
- LWE 問題をベースとした準同型暗号方式に関しては, AES 回路を暗号化したまま行うために, Gentry-Halevi-Smart ら [GHS12b] が [BGV12] で提案されたレベル付き完全準同型暗号の具体的なパラメータ設定方法を示している．一方, 完全準同型暗号ではなく限定回の加算と乗算が可能な somewhat 準同型暗号の具体的なパラメータとして, Lauter-Naehrig-Vaikuntanathan ら [LNV11] が [BV11] で提案された somewhat 準同型暗号を利用して, 平均・標準偏差・ロジスティック回帰などの統計計算を暗号化したまま行うための具体的なパラメータを表 [LNV11, Table 1] で示している．

3.3　まとめ

LWE (Learning with Errors) 問題は, Machine Learning(機械学習理論) から派生した問題で, GapSVP 及び SIVP の困難性に関する仮定のもとで解くことが難しいことが知られており, 本問題を効率的に解くことは困難であると予想されている．現在までに完全準同型暗号スキームをはじめとした, 様々な公開鍵暗号スキームのベースがこの LWE 問題をベースとして提案されており, 今後も安全な暗号を構成する上で重要な要素となると考えられる．現在までに知られている LWE 問題を解く最良アルゴリズムは指数時間の計算量を持っている．ただし, 実際の LWE 問題をベースとした暗号スキームの構成の際には, BKZ アルゴリズムなどの格子縮約アルゴリズムに対し耐性を持つようにパラメータ設定を行う必要があり, 安全でかつ演算機能等の要件を満足するような LWE パラメータを選択するための, 統一的な方法は知られておらず, 今後の課題となっている．また, LWE 問題に対する攻撃実験評価に関する結果もあまり知られていないため, 今後は計算機実験に関する研究も非常に重要になると思われることから, 安全性理論評価はもちろん攻撃実験評価の視点からも, 今後の動向に注意する必要がある．

第 3 章の参照文献

[ABB10] S. Agrawal, D. Boneh and X. Boyen, "Efficient lattice (H)IBE in the standard model", In *Advances in Cryptology–EUROCRYPT 2010*, Springer LNCS 6110, 553–572, 2010.

[ABPW13] Y. Aono, X. Boyen, L.T. Phong and L. Wang, "Key-private re-encryption under LWE", In *Progress in Cryptology–INDOCRYPT 2013*, Springer LNCS 8250, 1–18, 2013.

[ACPS09] B. Applebaum, D. Cash, C. Peikert and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems", In *Advances in Cryptology–CRYPTO 2009*, Springer LNCS 5677, 595–618, 2009.

[AGV09] A. Akavia, S. Goldwasser and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks", In *Theory of Cryptography–TCC 2009*, Springer LNCS 5444, 474–495, 2009.

[BG14] S. Bai and S.D. Galbraith, "Lattice decoding attacks on binary LWE", In *–Australasian Conference on Information Security and Privacy–ACISP 2014*, Springer LNCS 8544, 322–337, 2014.

[BGV12] Z. Brakerski, C. Gentry and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping", In *Innovations in Theoretical Computer Science–ITCS 2012*, ACM, 309–325, 2012.

[BV11] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages", In *Advances in Cryptology–CRYPTO 2011*, Springer LNCS 6841, 505–524, 2011.

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz and C. Peikert, Bonsai trees, or how to delegate a lattice basis, Journal of Cryptology, **25**(4) (2012), 601–639 (Preliminary version was presented at EUROCRYPT 2010), 2012.

[CN11] Y. Chen and P.Q. Nguyen, "BKZ 2.0: better lattice security estimates", In *Advanecs in Cryptology–ASIACRYPT 2011*, Springer LNCS 7073, 1–20, 2011.

[DGK10] Y. Dodis, S. Goldwasser, Y. Kalai, C. Peikert and V. Vaikuntanathan, "Public-key encryption schemes with auxiliary inputs", In *Theory of Cryptography–TCC 2010*, Springer LNCS 5978, 361–381, 2010.

[DPSZ12] I. Damgård, V. Pastro, N.P. Smart and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption", In *Advances in Cryptology–CRYPTO 2012*, Springer LNCS 7417, 643–662, 2012.

[DQ13] N. Döttling and J. Müller-Quade, "Lossy codes and a new variant of the learning-with-errors problem", In *Advances in Cryptology–EUROCRYPT 2013*, Springer LNCS 7881, 18–34, 2013.

[GN08] N. Gama and P.Q. Nguyen, "Predicting lattice reduction", In *Advances in Cryptolog–EUROCRYPT 2008*, Springer LNCS 4965, 31–51, 2008.

[Gen09] C. Gentry, "Fully homomorphic encryption using ideal lattices", In *Proc. 41st ACM Symp. on Theory of Computing–STOC 2009*, ACM, 169–178, 2009.

[GHS12a] C. Gentry, S. Halevi and N.P. Smart, "Fully homomorphic encryption with polylog overhead", In *Advances in Cryptology–EUROCRYPT 2012*, Springer LNCS 7237, 465–482, 2012.

[GHS12b] C. Gentry, S. Halevi and N.P. Smart, "Homomorphic evaluation of the AES circuit", In *Advances in Cryptology–CRYPTO 2012*, Springer LNCS 7417, 850–867, 2012.

[GHPS12] C. Gentry, S. Halevi, C. Peikert and N.P. Smart, "Ring switching in BGV-style homomorphic encryption", In *Security and Cryptography for Networks–SCN 2012*, Springer LNCS 7485, 19–37, 2012.

[GKPV10] S. Goldwasser, Y. Kalai, C. Peikert and V. Vaikuntanathan, Robustness of the learning with errors assumption, Tsinghua University Press, 2010.

[GPV08] C. Gentry, C. Peikert and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions", In *Proc. 40th ACM Symp. on Theory of Computing–STOC 2008*, ACM, 197–206, 2008.

[KTX07] A. Kawachi, K. Tanaka and K. Xagawa, "Multi-bit cryptosystems based on lattice problems", In *Public Key Cryptography–PKC 2007*, Springer LNCS 4450, 315–329, 2007.

[LMSV12] J. Loftus, A. May, N.P. Smart, F. Vercauteren, "On CCA-Secure Somewhat Homomorphic Encryption", In *Selected Areas in Cryptology–SAC 2011*, LNCS 7118, pp. 55–72. 2012.

[LNV11] K. Lauter, M. Naehrig and V. Vaikuntanathan, "Can homomorphic encryption be practical?", In *ACM workshop on Cloud computing security workshop–CCSW 2011*, ACM, 113–124, 2011.

[LP11] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based cryptography", In *RSA Conference on Topics in Cryptology–CT-RSA 2011*, Springer LNCS 6558, 319–339, 2011.

[LPR10] V. Lyubashevsky, C. Peikert and O. Regev, "On ideal lattices and learning with errors over rings", In *Advances in Cryptology–EUROCRYPT 2010*, Springer LNCS 6110, 1–23, 2010.

[LPR13] V. Lyubashevsky, C. Peikert and O. Regev, "A toolkit for ring-LWE cryptography", In *Advances in Cryptology–EUROCRYPT 2013*, Springer LNCS 7881, 35–54, 2013.

[Mic10] D. Micciancio, "Duality in lattice cryptography", Invited talk at Public Key Cryptography–PKC 2010.

[MP13] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with Small Parameters", In *Advances in Cryptology–CRYPTO 2013*, Part I, Springer LNCS 8042, 21–39, 2013.

[MR07] D. Micciancio and O. Regev, Worst-case to average-case reduction based on gaussian measures, SIAM J. Computing **37**(1) (2007), 267–302, 2007.

[MR09] D. Micciancio and O. Regev, "Lattice-based cryptography", In *Post Quantum Crytography*, Springer, 147–191, 2009.

[NC00] M.A. Nielsen and I.L. Chuang, Quantum computation and quantum information, Cambridge University Press, 2000.

[Pei09] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problems", In *Proc. 41st ACM Symp. on Theory of Computing–STOC 2009*, ACM, 333–342, 2009.

[PVW08] C. Peikert, V. Vaikuntanathan and B. Waters, "A framework for efficient and composable oblivious transfer", In *Advances in Cryptology–CRYPTO 2008*, Springer LNCS 5157, 554–571, 2008.

[PW08] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications", In *Pro. 40th ACM Symp.*

on Theory of Computing–STOC 2008, ACM, 187–196, 2008.

[Reg05] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, J. ACM, **56**(6) (2009), 1–40 (Preliminary version was presented at STOC 2005), 2009.

[Reg] O. Regev, The learning with errors problem, survey paper, available at `http://www.cims.nyu.edu/~regev/papers/lwesurvey.pdf`.

[Reg09] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, (2009), available at `http://www.cims.nyu.edu/~regev/papers/qcrypto.pdf`.

[SV11] N.P. Smart and F. Vercauteren, Fully homomorphic SIMD operations, Designs, Codes and Cryptography, preprint, July 2012, doi:10.1007/s10623-012-9720-4, 2012.

第 4 章

LPN

4.1 Learning Parity with Noise (LPN) 問題の概説

本章では Learning with Parity Noise (LPN) 問題や符号に関連する問題の困難性について調査結果を述べる.

4.1.1 LPN 問題とは

LPN 問題とは誤差付きの線型方程式を解けるかどうかという問題である. 1993 年に, Blum, Furst, Kearns, Lipton [BFKL93] が困難と思われる問題として挙げ, 定式化を行った. 第 3 章において, この問題を一般化した LWE 問題を既に扱っている.

以下では $\mathbb{F}_q$ で位数が q の有限体を表す. Ber_τ でパラメータ τ のベルヌーイ分布を表すことにする. (確率 τ で 1, 確率 $1-\tau$ で 0 となる $\mathbb{F}_2$ 上の分布である.) また, 自然数 $k \geq 1$ について, Ber_τ^k で, Ber_τ から独立に k 個サンプルを取ったときの $\mathbb{F}_2^k$ 上の分布を表す.

■LPN **問題**: $\mathbb{F}_2$ 上の分布 χ および $\vec{s} \in \mathbb{F}_2^n$ について, オラクル $\mathcal{O}_{\vec{s},\chi}$ を以下で定義する. (1) $\vec{a}$ を $\mathbb{F}_2^n$ からランダムに選び, (2) e を分布 χ に従い選び, (3) $b = \vec{s} \cdot \vec{a}^\top + e$ と計算し, (4) $(\vec{a}, b)$ を出力する. 定義より, このオラクルは第 3 章定義 3.1 で定義される分布 $A_{\vec{s},\chi}$ からのサンプル $(\vec{a}, b) \in \mathbb{F}_2^{n+1}$ を返す. また, オラクル $\mathcal{U}$ を $(\vec{a}, b) \leftarrow \mathbb{F}_2^{n+1}$ とランダムな組を出力するオラクルとして定義する.

定義 4.1 (探索版 LPN 問題) 探索版 LPN 問題とは, オラクル $\mathcal{O}_{\vec{s},\chi}$ へのアクセスが可能なときに, $\vec{s}$ を出力する問題である.

特に $\chi = \mathsf{Ber}_\tau$ のとき, $\mathsf{LPN}_{n,\tau}$ 問題と呼ぶ. また $\mathsf{LPN}_{n,\tau}$ 問題でオラクルからのサンプル数が $m = m(n)$ に制限されるものを, $\mathsf{LPN}_{n,m,\tau}$ 問題と呼ぶ.

定義 4.2 (探索版 LPN 仮定) $\mathbb{F}_2$ 上の確率分布 χ について, 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(n) = \Pr_{\vec{s} \leftarrow \mathbb{F}_2^n}[\mathcal{A}^{\mathcal{O}_{\vec{s},\chi}}(1^n) = \vec{s}]$$

で定義する. 任意の多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できるとき, 探索版 LPN 仮定が成立するという.

暗号プリミティブや暗号プロトコルの安全性証明のために, 判定版 LPN 仮定を用いることも多い. 判定版 LPN 問題と判定版 LPN 仮定は以下で定義される.

定義 4.3 (判定版 LPN 問題) 判定版 LPN 問題とは, オラクル $\mathcal{O}_{\vec{s},\chi}$ またはオラクル $\mathcal{U}$ へのアクセスが与えられたときに, どちらのオラクルにアクセスしているかを判定する問題である.

定義 4.4 (判定版 LPN 仮定) $\mathbb{F}_2$ 上の確率分布 χ について, 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(n) = \left| \Pr_{\vec{s}\leftarrow\mathbb{F}_2^n}[\mathcal{A}^{\mathcal{O}_{\vec{s},\chi}}(1^n) = 1] - \Pr[\mathcal{A}^{\mathcal{U}}(1^n) = 1] \right|$$

で定義する. 任意の多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できる関数であるとき, 判定版 LPN 仮定が成立するという.

探索版 LPN 問題にはランダム自己帰着が存在する [BFKL93]. すなわち, ランダムに選ばれた $\vec{s} \in \mathbb{F}_2^n$ について探索版 LPN 問題を解けるならば, 任意の $\vec{s} \in \mathbb{F}_2^n$ について探索版 LPN 問題を解くことが出来る.

Katz, Shin, Smith [KSS10] によれば, [BFKL93, Reg09] と同様に判定版 LPN 仮定を探索版 LPN 仮定に帰着することが出来る.

定理 4.5 ([KSS10]) 判定版 $\mathsf{LPN}_{n,\tau}$ 仮定を破る t ステップ, m 回のクエリ, 優位性 δ の敵が存在すると仮定する. このとき, 探索版 $\mathsf{LPN}_{n,\tau}$ 仮定を破る t' ステップ, m' 回のクエリ, 優位性 δ' の敵が存在する. ここで,

$$t' = O(\delta^{-2} tn \log n),\ m' = O(\delta^{-2} m \log n),\ \delta' \geq \delta/4.$$

■変種: 以上に列挙した LPN 問題・仮定では, 基礎となる体として $\mathbb{F}_2$ を用いていた. 体を $\mathbb{F}_q$ に変更した LPN 問題・仮定が用いられることもある. 特に q を素数とした場合には LWE 問題と非常によく似た問題・仮定となるが, 誤差分布 χ の定義が異なることが多い.

LWE 問題では剰余環 $\mathbb{Z}_q$ を用いている. 応用の観点からは, 誤差分布 χ からのサンプル x の絶対値が高い確率で小さいことが重視される.

一方, LPN 問題では有限体 $\mathbb{F}_q$ を用いている. また, 符号からの要求としてハミング重みを考えることが多いため, 誤差分布 χ は 0 を取る確率が大きいことが求められる. たとえば, ベルヌーイ分布の一般化として, 確率 τ で 0 を確率 $1-\tau$ で $\mathbb{F}_q \setminus \{0\}$ のランダムな値を取る分布が用いられる. これは格子問題と符号問題のアナロジーとして考えることができる.

4.1.2 LPN 問題の拡張

4.1.2.1 復号問題

オラクルからのサンプル数を固定し $m = m(n)$ とする. $\mathsf{LPN}_{n,m,\tau}$ 問題での m 個のサンプル $(\vec{a}_1, b_1), (\vec{a}_2, b_2), \ldots, (\vec{a}_m, b_m)$ を行列・ベクトル表示して,

$$\boldsymbol{A} = [\vec{a}_1^\top \vec{a}_2^\top \ldots \vec{a}_m^\top] \in \mathbb{F}_2^{n\times m}, \vec{b} = \vec{s}\cdot\boldsymbol{A} + \vec{e}$$

とする. 符号理論の観点からは, ランダム行列 $\boldsymbol{A}$ を生成行列とする線形符号の受信語 $\vec{b}$ から元のメッセージ $\vec{s}$ を復元する問題と捉えることができる.

4.1.2.2 シンドローム復号問題

先ほど挙げた復号問題の “双対” として, シンドローム復号問題が挙げられる. シンドローム復号問題 $\mathsf{SD}_{k,m,w}$ とは,

$$\boldsymbol{H} = [\vec{h_1}^\top \vec{h_2}^\top \ldots \vec{h_m}^\top] \in \mathbb{F}_2^{k\times m}, \vec{u} \in \mathbb{F}_2^k$$

および自然数 w が与えられた時に, $\vec{e} \cdot \boldsymbol{H}^\top = \vec{u}$ かつハミング重みが w 以下となる $\vec{e} \in \mathbb{F}_2^m$ を求める問題である.

$\boldsymbol{H}$ として $\boldsymbol{A}$ で生成される符号のパリティ検査行列を取り, $\vec{u}$ として $\vec{b} \cdot \boldsymbol{H}^\top (= \vec{e} \cdot \boldsymbol{H}^\top)$ をとれば, $\mathsf{LPN}_{n,m,\tau}$ 問題や復号問題をシンドローム復号問題 $\mathsf{SD}_{m-n,m,O(\tau m)}$ に変換可能である.

4.1.2.3 Exact-LPN 問題

誤差分布として, $\vec{e} \leftarrow \mathsf{Ber}_\tau^m$ ではなく, ハミング重みが丁度 w のものだけを考える. このように誤差分布を変えた問題を Exact-LPN 問題と呼ぶ.

4.1.2.4 Sparse-LPN 問題

一部の暗号方式では, $\vec{s}$ のハミング重みが小さい, すなわち, 疎 (sparse) であることを要求する. Applebaum ら [ACPS09] は $\vec{s}$ を誤差分布である χ^n から選んだ場合の LPN 問題と $\vec{s}$ を $\mathbb{F}_2^n$ からランダムに選んだ場合の問題とが等価であることを示している.

4.1.2.5 Subspace-LPN 問題

Pietrzak [Pie12a] は, 敵のオラクルへのクエリを強めた問題として, 以下の Subspace-LPN 問題を考察した. LPN 仮定で定義されたオラクルを $\mathcal{O}_{\vec{s},\chi}$ から以下で定義される $\mathcal{O}'_{\vec{s},\chi}$ に変更する. 二つの Affine 関数 $\phi_a(\vec{a}) = \vec{a}\boldsymbol{X}_a + \vec{x}_a$, $\phi_s(\vec{s}) = \vec{s}\boldsymbol{X}_s + \vec{x}_s$, $(\boldsymbol{X}_a, \boldsymbol{X}_s \in \mathbb{F}_2^{n \times n}, \vec{x}_a, \vec{x}_s \in \mathbb{F}_2^n,)$ をクエリとして受け取り, $\mathrm{rank}(\boldsymbol{X}_a^\top \boldsymbol{X}_s) \geq d + \delta$ ならば, $\vec{a} \leftarrow \mathbb{F}_2^n$ および $b = \phi_s(\vec{s}) \cdot \phi_a(\vec{a})^\top + e$ を出力する.

Peitrzak は, $2^{-\delta+1}$ が無視できるならば, 新しいオラクル $\mathcal{O}'_{\vec{s},\chi}$ を用いた Subspace-LPN 問題は, 次元 d の LPN 問題と困難性が等価であることを示した.

4.1.2.6 Toeplitz-LPN 問題

Gilbert, Robshaw, Seurin [GRS08] が認証プロトコルの効率化のために導入した.

行列 $\boldsymbol{A} = \{a_{i,j}\} \in \mathbb{F}_2^{n \times m}$ が Toeplitz 行列であるとは, 任意の i, j について $a_{i-1,j-1} = a_{i,j}$ が成立することである. Toeplitz 行列を表現するには左端の列ベクトルおよび最も上の行ベクトルがあれば良い. そのため $\boldsymbol{A}$ の表現は $n + m - 1$ ビットで可能である.

復号問題の節で, 探索版 LPN 問題でのサンプル $(\vec{a}_1, b_1), (\vec{a}_2, b_2), \ldots, (\vec{a}_m, b_m)$ を行列・ベクトル表示して,

$$\boldsymbol{A} = [\vec{a}_1^\top \vec{a}_2^\top \ldots \vec{a}_m^\top] \in \mathbb{F}_2^{n \times m}, \vec{b} = \vec{s} \cdot \boldsymbol{A} + \vec{e}$$

を考えた. オラクル $\mathcal{O}$ (および $\mathcal{U}$) を変更し, $\boldsymbol{A}$ が必ず Toeplitz 行列になる場合の LPN 問題を考える. これを Toeplitz-LPN 問題と呼ぶ.

4.1.2.7 Ring-LPN 問題

Heyse, Kiltz, Lyubashevsky, Paar, Pietrzak [HKL+12] は, Ring-LPN 問題を定義した. この問題は ring-LWE 問題 (定義 3.2) と同様に定義される.

定義 4.6 (探索版 Ring-LPN 問題) 適当な n 次の $\mathbb{F}_q$ 係数多項式 $f(x)$ を考え, 環 $R_q = \mathbb{F}_q[x]/(f(x))$ を固定する. R_q 上の確率分布 χ を固定する.

R_q 上の誤差分布 χ および $s \in R_q$ について, オラクル $\mathcal{O}_{\vec{s},\chi}$ を以下で定義する. (1) a を R_q からランダムに選び, (2) e を分布 χ に従い選び, (3) $b = sa + e$ と計算し, (4) $(a, b) \in R_q^2$ を出力する.

探索版 Ring-LPN 問題とは, オラクル $\mathcal{O}_{s,\chi}$ へのアクセスが可能なときに, $s \in R_q$ を出力する問題である.

4.2 LPN 問題のアプリケーション

90 年代から様々な応用が提案されている.

疑似乱数生成器 Blum, Furst, Kearns, Lipton [BFKL93] による疑似乱数生成器が有名である. Fischer, Stern [FS96] の構成や Appelbaum, Cash, Peikert, Sahai [ACPS09] による構成も知られている

共通鍵による両側認証 Hopper と Blum によって, 後に HB プロトコルと呼ばれる認証プロトコルが提案された [HB01]. 多くの変種が提案されており, 現在も研究が続けられている.

共通鍵暗号 Gilbert, Robshaw, Seurin [GRS08] による LPN-C と呼ばれる IND-CPA 安全な共通鍵暗号方式がある. Applebaum, Cash, Peikert, Sahai [ACPS09] は, LPN-C の変種が KDM-CPA 安全であることを示した. また Applebaum, Harnik, Ishai [AHI11] は ACPS09 の共通鍵方式が RKA-CPA 安全であることを示し, OT への応用を考察している. Applebaum [App13] は ACPS09 の共通鍵方式が RKA-KDM-CPA 安全であることを示し, このような方式を用いれば, Free-XOR 構成を用いた Yao's GC が標準モデルで安全であることを示した.

署名 大別して二つのタイプがある.

- Fiat-Shamir 変換によるもの: Stern の認証方式や Veron の認証方式に Fiat-Shamir 変換を施すことによって得られる署名である. 署名長の観点から効率が悪く実用には向いていない.
- Full-Domain Hash によるもの: CFS 署名が知られている.

今までのところ標準モデルでの安全性証明は行われていない.

公開鍵暗号 大きく分けて二つの系統がある.

- Alekhnovich 暗号: Alekhnovich [Ale11] 暗号は LPN 仮定のみから IND-CPA 安全性を証明可能な方式である. IND-CCA2 版は Döttling, Müller-Quade, Nasciment [DMQN12] によって構成されている.
- McEliece 暗号または Niederreiter 暗号: McEliece [McE78] および Niederreiter [Nie86] によって提案された暗号である. Li, Deng, Wang [LDW94] が「Niederreiter 暗号の OW-CPA 性は McEliece 暗号の OW-CPA 性と等価である」ことを示している.
  - McEliece 暗号の派生: Kobara, Imai [KI01] は McEliece 暗号用のパディング方法を提案し, その方式がランダムオラクルモデルで ND-CCA2 安全であることを示した. Nojima, Imai, Kobara, Morozov [NIKM08] は McEliece 暗号の変種が標準モデルで IND-CPA 安全でああることを示した. McEliece 暗号を基にした IND-CCA2 暗号については [DDMQN12] や Persichetti [Per13] に構成が見られる.
  - Niederreiter 暗号の派生: 標準モデルで IND-CCA2 安全な Niederreiter ベースの暗号として, Freeman, Goldreich, Kiltz, Rosen, Segev の構成 [FGK+13] や, Mathew, Vasant, Venkatesan, Rangan の構成 [MVVR12] が知られている.

紛失転送 McEliece 暗号を用いた紛失転送プロトコルが提案されている [DvdGMQN12, DNdS12, DNMQ12].

以下では, LPN 仮定に基づく公開鍵暗号方式の例として Alekhnovich 暗号を取りあげる. また, 追加の仮定が必要であるが Alekhnovich 暗号よりも効率が良い公開鍵暗号方式の例として McEliece 暗号を取り上げる.

4.2.1 Alekhnovich 暗号 [Ale11]

Alekhnovich は [Ale11] で 2 つ公開鍵暗号方式を提案している. ここではシンプルな 1 つ目の暗号方式を取り上げる. パラメータを以下とする.

- n: 安全性パラメータ
- m: LPN サンプルの個数 (例: $m = 2n + 1$)
- $\tau > 0$: 誤差パラメータ (例: $\tau = n^{-1/2-\epsilon}$)

このとき Alekhnovich 暗号は以下で構成される:

秘密鍵の生成: ランダムに $\vec{e} \leftarrow \mathsf{Ber}_\tau^m$ を選ぶ.

公開鍵の生成 ランダムに $\boldsymbol{A} \leftarrow \mathbb{F}_2^{n\times m}$ を選ぶ. ランダムに $\vec{s} \leftarrow \mathbb{F}_2^n$ を選ぶ. $\vec{b} = \vec{s}\boldsymbol{A} + \vec{e} \in \mathbb{F}_2^m$ を計算し, $\boldsymbol{B} = \binom{\boldsymbol{A}}{\vec{b}}$ とする. $\boldsymbol{M} \in \mathbb{F}_2^{(m-n-1)\times m}$ を $\ker(\boldsymbol{B}^\top)$ の基底とし, 公開鍵を $\boldsymbol{M}$ とする.

暗号化 平文が 0 の場合, $\vec{t} \leftarrow \mathbb{F}_2^{m-n-1}$ と $\vec{f} \leftarrow \mathsf{Ber}_\tau^m$ をランダムに選び, $\vec{c} = \vec{t}\boldsymbol{M} + \vec{f} \in \mathbb{F}_2^m$ を出力する.
平文が 1 の場合, ランダムに $\vec{c} \leftarrow \mathbb{F}_2^m$ を選び出力する.

復号 暗号文 $\vec{c} \in \mathbb{F}_2^m$ について, $\delta = \langle \vec{c}, \vec{e} \rangle$ を計算する. δ を出力する.

復号の正当性について以下考察する. 平文が 1 の場合, 復号は確率 $1/2$ で成功する.
一方, 平文が 0 の場合, $\vec{e} \in \mathrm{Span}(\boldsymbol{B})$ および $\vec{t}\boldsymbol{M} \in \ker(\boldsymbol{B}^\top)$ より $\vec{t}\boldsymbol{M}\vec{e}^\top = 0$ であることに注意すると,

$$\langle \vec{c}, \vec{e} \rangle = \vec{t}\boldsymbol{M} \cdot \vec{e}^\top + \vec{f}\vec{e}^\top = \langle \vec{f}, \vec{e} \rangle$$

なので, $\langle \vec{f}, \vec{e} \rangle = 0$ であれば復号に成功する. 誤り確率を評価すると, $\Pr[\langle \vec{f}, \vec{e} \rangle = 1] \approx (1-\tau)^{\tau m} = o(1)$ となり, $1 - o(1)$ の確率で復号に成功する. また, 上記の暗号方式の安全性については, 判定版 LPN 仮定の下で CPA 安全であることが証明される.

ここで紹介した暗号方式は, 1 ビット暗号であり, 復号誤りの確率も高いため実用的ではない. 効率的な方式としては 2 つ目の Alekhnovich 暗号や次に挙げる McEliece 暗号を参考にされたい.

4.2.2 McEliece 暗号

以下では, S_m で m 次対称群を表し, $\mathrm{GL}_n(\mathbb{F}_q)$ で n 次の $\mathbb{F}_q$ 要素正則行列全体がなす群を表す.

- n: 安全性パラメータ
- m: サンプルの個数
- τ: 誤差パラメータ (例: $\tau = cn$)
- t: 誤り訂正符号の誤り訂正能力 ($t = \Omega(\tau m)$)

鍵生成: 誤り訂正能力が t である (m, n)-線形符号の生成行列 $\boldsymbol{G}$ を生成する. $\boldsymbol{S} \leftarrow \mathrm{GL}_n(\mathbb{F}_2)$ をランダムに選ぶ. $\boldsymbol{P} \leftarrow S_m$ をランダムに選ぶ. $\boldsymbol{M} = \boldsymbol{SGP}$ とする.
公開鍵を $\boldsymbol{M}$ とし, 秘密鍵を $(\boldsymbol{S}, \boldsymbol{G}, \boldsymbol{P})$ とする.

暗号化: 平文を $\vec{v} \in \mathbb{F}_2^n$ とする. 乱数 $\vec{e} \leftarrow \mathsf{Ber}_\tau^m$ を選び, 暗号文 $\vec{c} = \vec{v}\boldsymbol{M} + \vec{e}$ を計算する.

復号: $\hat{\vec{v}} = \vec{c}\boldsymbol{P}^{-1}$ を計算する. $\hat{\vec{v}}$ を誤り訂正符号で訂正し復号すると $\vec{v}' = \vec{v}\boldsymbol{S}$ を得る. $\vec{v} = \vec{v}'\boldsymbol{S}^{-1}$ を出力する.

復号の正当性は以下で確認される. $\vec{c} = \vec{v}\boldsymbol{M} + \vec{e}$ として, $\hat{\vec{v}} = \vec{c}\boldsymbol{P}^{-1}$ を計算すると,

$$\hat{\vec{v}} = \vec{v}\boldsymbol{M}\boldsymbol{P}^{-1} + \vec{e}\boldsymbol{P}^{-1} = \vec{v}\boldsymbol{S}\boldsymbol{G} + \vec{e}\boldsymbol{P}^{-1}$$

を得る. $\vec{v}\boldsymbol{S}\boldsymbol{G}$ は符号語であり, $\vec{e}\boldsymbol{P}^{-1}$ は誤りであり. $\vec{e}\boldsymbol{P}^{-1}$ の重みが t 以下であれば, 誤り訂正符号の復号により, $\vec{v}' = \vec{v}\boldsymbol{S}$ を得る. よって, 高い確率で復号に成功する.

平文 $\vec{v}$ および $\boldsymbol{M}$ がランダムであれば, 暗号文 $\vec{c}$ は LPN 仮定の下で疑似ランダムである. $\boldsymbol{M}$ が疑似ランダムであることを言うためには, McEliece 仮定と呼ばれる仮定が必要となる.

定義 4.7 (McEliece 仮定) $[m(n), n]_{q(n)}$-符号のクラス $\mathcal{C}$ を固定する. 敵 $\mathcal{A}$ の優位性を

$$\mathsf{Adv}_{\mathcal{A}}(n) = \left| \Pr_{\boldsymbol{S}\leftarrow \mathrm{GL}_n(\mathbb{F}_q), \boldsymbol{G}\leftarrow\mathcal{C}, \boldsymbol{P}\leftarrow S_m}[\mathcal{A}(1^n, \boldsymbol{M} = \boldsymbol{S}\boldsymbol{G}\boldsymbol{P}) = 1] - \Pr_{\boldsymbol{M}\leftarrow\mathbb{F}_q^{n\times m}}[\mathcal{A}(1^n, \boldsymbol{G}) = 1] \right|$$

で定義する. 任意の多項式時間の敵 $\mathcal{A}$ について, その優位性が無視できる関数であるとき, McEliece 仮定が成立するという.

左側の敵は McEliece 暗号の公開鍵 (または Niederreiter 暗号の公開鍵の双対) を受け取っている. そのため, この仮定は, McEliece 暗号の公開鍵はランダムな同サイズの行列と見分けが付かないということを意味する.

Faugère, Gauthier-Umaña, Otmani, Perret, Tillich [FGOPT13] は元となる Goppa 符号 (または Alternant 符号) のレートが高い場合には, McEliece 仮定を破るアルゴリズムを提案している. 暗号として用いる場合には, パラメータ設定によって彼らの攻撃を避けることが可能である.

■McEliece 暗号の変種の安全性について: 二元 Goppa 符号を用いた場合, 鍵サイズが大きくなることが知られている. そのため, 元となる符号を変更し, 鍵サイズや暗号文サイズを小さくする研究が進められてきた. しかし, 符号が特殊な場合には多くの方式が破られている. McEliece 暗号やその変種を用いる場合には, 符号の選定やパラメータの設定おいて, より一層の注意が必要である.

Bernstein, Lange, Peters が [BLP10] および [BLP11a] で q 元-Goppa 符号を用いた q 元-McEliece 暗号についてパラメータの提案を行っている.

■パラメータ設定について: Bernstein, Lange, Peters は [BLP10] および [BLP11a] で q 元-Goppa 符号を用いた q 元-McEliece 暗号についてパラメータの提案を行っている. 具体的なチャレンジ問題も入手可能である. [*1] たとえば 128-bit 安全性を考える際には, $(q, n, m, \tau m) = (2, 2325, 3009, 57)$ といったパラメータを提案している.

LPN 問題をベースとした暗号方式を実際に構成する際には, 4.3 節で挙げる各種のアルゴリズムに耐性を持つよう, パラメータ設定を行う必要がある. たとえば, Damgård と Park [DP12] は Alekhnovich 暗号の変種として公開鍵暗号を提案し, Bernstein と Lange の攻撃 [BL12] を元にしたパラメータ設定 (表 4.1) を行っている.

4.3 LPN 問題に対する評価

サンプル数を固定した場合, $\boldsymbol{A}$ および $\vec{b}$ の最悪時を考えると NP 困難になることが Berlekamp, McEliece, van Tilborg [BMvT78] によって示されている. また, Håstad [Hås01] により近似版 LPN 問題の NP 困難性も示されている.

[*1] `http://pqcrypto.org/wild-challenges.html`.

表 4.1　Damgård と Park によるパラメータ設定の例 ([DP12] より)

セキュリティレベル	n	τ
80-bit	9000	0.0044
112-bit	21000	0.0029
128-bit	29000	0.0024
196-bit	80000	0.0015
256-bit	145000	0.0011

しかし平均時の困難性についてはよく分かっていない．そのため LPN 問題を解くための提案されたアルゴリズムについて調査を行った．

$\mathsf{LPN}_{n,m,\tau}$ 問題を解くための素朴な方法として, 総当たり法がある．閾値 $d \geq 1$ を固定する．$\vec{s} \in \mathbb{F}_2^n$ の候補ごとに, $\vec{e} = \vec{b} - \vec{s}\boldsymbol{A}$ を計算し, $\vec{e}$ のハミング重みが $(1+1/d)\tau m$ 以下であれば $\vec{s}$ を解として出力するというものである．Chernoff の補題から $\vec{e} \leftarrow \mathsf{Ber}_\tau^m$ としたとき, $d \geq 1$ について $\Pr[Hw(\vec{e}) \leq (1+1/d)\tau m] \leq \exp(-\tau m/3d^2)$ である．従ってこの方法を用いると, 時間 $O(2^n)$ で圧倒的な確率で $\mathsf{LPN}_{n,m,\tau}$ 問題を解くことが可能である．

以降では, $O(2^n)$ 以下の時間で解を求めるアルゴリズムについて考察する．現在では, 大別して 3 つのアルゴリズムが知られている．

1. Blum, Kalai, Wasserman [BKW03] の BKW アルゴリズム,
2. Arora, Ge [AG11] の「再線形化」アルゴリズム,
3. シンドローム復号問題として解くアルゴリズム

である．

4.3.1　BKW アルゴリズムおよびその改良

Blum, Kalai, Wasserman [BKW03] は BKW アルゴリズムと呼ばれるアルゴリズムを提案した．

基本アイデアは以下である．オラクルからのサンプル $(\vec{a}, b)$ が常に $\vec{a} = (1, 0, \ldots, 0)$ という形であれば, $b = s_1 + e$ となる．このようなサンプルを大量に集めれば, s_1 を多数決法で求めることが出来る．一般に $\vec{u}_j$ を j 番目の単位ベクトルとして, $(\vec{u}_j, b)$ という形のサンプルを集めれば s_j を多数決法で求められる．そこでオラクル $\mathcal{O}_{\vec{s},\tau}$ からのサンプルを用いて, 上記のようなサンプルを生成することを目指す．

■BKW アルゴリズムの概要:　$(t-1)k < n \leq tk$ を満たす適当な自然数 t, k を固定する．以下では,

$$A_{\vec{s},\delta,i} := \{\vec{a} \leftarrow \mathbb{F}_2^{n-ik} \times \{0\}^{ik}, e \leftarrow \mathsf{Ber}_{(1+\delta)/2} : (\vec{a}, \vec{s}\cdot\vec{a}^\top + e)\}$$

というオラクルを考える．$A_{\vec{s},\delta,i}$ から得たサンプル $(\vec{a}, b)$ は $\vec{a}$ の末尾から ik 個の要素が必ず 0 である．$i = 0$, $\delta = 1-2\tau$ とすれば, $A_{\vec{s},\delta,i} = \mathcal{O}_{\vec{s},\tau}$ となる．

基本アルゴリズムは以下である．

1. $A_{\vec{s},\delta,i}$ からのサンプルを L_0 個用意する．
2. $i = 0, 1, \ldots, t-2$ について, サイズ L_i の $A_{\vec{s},\delta,i}$ からのサンプルを用いて, $O(L_i)$ 時間でサイズ $L_{i+1} = L_i - 2^k$ の $A_{\vec{s},\delta^2,i+1}$ からのサンプルを構成する．
   - サンプル $(\vec{a}, b) \in L_i$ について, $\vec{a} = (a_1, a_2, \ldots, a_{n-ik}, 0, \ldots, 0) \in \mathbb{F}_2^n$ の $(a_{n-(i+1)k+1}, a_{n-(i+1)k+2}, , \ldots, a_{n-ik}) \in$

$\mathbb{F}_2^k$ に従って分類を行う.

- 各組で代表を一つとり, それを $(\vec{a}^*, b^*)$ とする.
- 各組の代表以外の要素 $(\vec{a}, b)$ を $(\vec{a} \oplus \vec{a}^*, b \oplus b^*)$ で置き換える.
- 全組をまとめてサイズ $L_i - 2^k$ の $A_{\vec{s},\delta^2,i+1}$ からのサンプルとする.

最終的に, サイズ $L_{t-1} = L - (t-1)2^k$ の $A_{\vec{s},\delta^{2^{t-1}},t-1}$ からのサンプルが得られる.

3. 得られた L_{t-1} 個の $A_{\vec{s},\delta^{2^{t-1}},t-1}$ からのサンプルを用いて, s_j を投票で決める.
   - $j = 1, 2, \ldots, n-(t-1)k$ について, $\vec{u}_j$ を $\mathbb{F}_2^n$ の標準基底 j 番目の単位ベクトルとする. サンプル $\{(\vec{a}_i, b_i)\}_{i=1,2\ldots,m}$ から ℓ 個のベクトルを $\vec{a}_{i_1} + \vec{a}_{i_2} + \ldots + \vec{a}_{i_\ell} = \vec{u}_j$ となるようにうまく選ぶ. このとき, $b_{i_1} + b_{i_2} + \ldots + b_{i_\ell} = s_j + e_{i_1} + \ldots + e_{i_\ell}$ となり, 誤差が 0 になる確率は $\Pr[e_{i_1} + e_{i_2} + \ldots + e_{i_\ell} = 0] > 1/2 + (1-2\delta^{2^{t-1}})^\ell/2$ で与えられる. 適当な回数この試行を行い, s_j を多数決投票で決めれば良い.

Blum らの見積もりでは, サンプル数および計算ステップ数は $\delta = 1-2\tau$ として, $\mathrm{poly}\left(\delta^{-2^t}, 2^k\right)$ であった. $\tau < 1/2$ を定数とし, $t = \frac{1}{2}\log n$, $k = 2n/\log n$ とすれば, $2^{O(n/\log n)}$ を得る.

■LF アルゴリズム: Levieil と Fouque [LF06] は BKW アルゴリズムの一部アルゴリズムを改良し LF アルゴリズムを提案した.

簡単のために $n = tk$ を仮定する. BKW アルゴリズムでは基本アルゴリズムのステップ 3 において $\vec{s}$ の各要素を 1 ビットずつ決定している. ステップ 3 において得られたサンプルは, $A_{\vec{s},\delta^{2^{t-1}},t-1}$ からのサンプルであるため, $((a_1, a_2, \ldots, a_k, 0, \ldots, 0), b)$ という形をしている. このとき, $b = \sum_{i=1}^k a_i s_i + e$ となり, サンプルに影響を与えるのは, $\vec{s}$ の k ビット分である. LF アルゴリズムでは, $s_1, s_2, \ldots, s_k$ を総当りで計算する.

Levieil と Fouque は BKW アルゴリズムおよび LF アルゴリズムが必要とするサンプル数および計算ステップ数を, 以下のように詳細に解析した.

定理 4.8 $n = tk$ とし, $\delta = 1-2\tau$ とする.

- BKW アルゴリズムはクエリ数 $m = 20\ln(4n)2^k\delta^{-2^t}$, ステップ数 $t = O(ntm)$, メモリ量 $M = nm$, 成功確率 $\theta = 1/2$ で $\mathsf{LPN}_{n,m,\tau}$ 問題を解く.
- LF アルゴリズムはクエリ数 $m = (8k+200)\delta^{-2^t} + (t-1)2^k$, ステップ数 $t = O(ntm)$, メモリ量 $M = nm + k2^k$. 成功確率 $\theta = 1/2$ で $\mathsf{LPN}_{n,m,\tau}$ 問題を解く.

彼らの報告によれば, LF アルゴリズムと一部のヒューリスティクな手法を用いて $n = 99$, $\tau = 1/4$, $m = 10000$ の LPN 問題を CPU: Pentium 4 (3GHz), RAM: 1GB のマシンで解くことが可能である.

■Kirchner の指摘: Kirchner [Kir11] はランダムに選ばれた $\vec{s}$ よりは Ber_τ に従って選ばれる誤りベクトル $\vec{e}$ の方が, ハミング重みが小さくバリエーションが少ないことに着目した. LPN 問題を Sparse-LPN 問題に置き換えた上で問題を解くことを提案している.

Kirchner の手法は以下のようにまとめられる.

1. Applebaum ら [ACPS09] と同様の手法を用いて, $\mathcal{O}_{\vec{s},\chi}$ というオラクルを $\vec{e}' \leftarrow \mathsf{Ber}_\tau^n$ とランダムに選んだ場合の $\mathcal{O}_{\vec{e}',\chi}$ というオラクルに変換する.
2. BKW アルゴリズムや LF アルゴリズムと同様に基本アルゴリズムのステップ 1, 2 を行い, $A_{\vec{e}',\delta^{2^{t-1}},t-1}$ からのサンプルを得る.
3. ステップ 3 で, k ビットを決定する際に, $\vec{e}'$ の該当部分の重みが少ないことを考慮して総当りを行う.

表 4.2　Becker らによる確率 1/2 以上で SD 問題を解く場合のパラメータ例 [BJMM12]

	log(時間計算量)/m	log(空間計算量)/m	備考
Lee-Brickel	0.05752	–	[LB88]
Stern	0.05564	0.0135	[Ste88]
BLP	0.05549	0.0148	[BLP11b]
MMT	0.05364	0.0216	[MMT11]
BJMM	0.04934	0.0286	[BJMM12]

一般の $\vec{s}$ であれば, 総当りに必要な回数は 2^k となる. 一方, $\vec{e}'$ はスパースであることが期待される. $d \geq 1$ を固定し k が十分に大きいとする. このとき, 圧倒的な確率の下で, ハミング重みは $(1+1/d)\tau k$ 以下である. よって, $\vec{e}'$ の候補数は $\binom{k}{(1+1/d)\tau k}$ 以下となり, 総当りに必要な回数が削減される.

■Ring-LPN 問題への応用:　Bernstein と Lange [BL12] は Levieil と Fouque の高速化手法および Kirchner のアイデアを用いることにより, Ring-LPN 問題の解法が高速化できることを示している.

■GJL アルゴリズム:　Guo, Johansson, Löndahl [GJL14] は, covering codes と呼ばれる符号を用いて Kirchner の手法の高速化を提案している. Kirchner の手法ではステップ 3 で, $A_{\vec{e}',\delta 2^{t-1},t-1}$ からのサンプル $\{(\vec{a}_i, b_i)\}$ が得られる. この $\vec{a}_i$ を covering code の受信語とみなすことで探索空間の圧縮を行い, 高速化に成功している.

■サンプル数が少ない場合:　これまでに挙げてきた BKW アルゴリズムおよびその改良では, サンプルが $O(2^{n/\log n})$ 個必要であった. Lyubashevsky [Lyu05] はサンプル数が $n^{1+\epsilon}$ 個と少ない場合であっても, BKW アルゴリズムを適用できるような指数個のサンプルの構成法を示している. また, 上中谷と國廣 [KK15] は BKW アルゴリズムと Lyubashevsky の方法とを補間するようなアルゴリズムを提案している.

4.3.2　Arora-Ge アルゴリズム

Arora と Ge [AG11] は多変数多項式問題で古くから用いられている再線形化と呼ばれる手法を用いて, LPN 問題を解くことを考えた. このアルゴリズムを $\mathsf{LPN}_{n,m,\tau}$ に用いた場合, $w = \tau m$ として, $\mathrm{poly}(n^w)$ 時間で解くことができる. $\mathrm{poly}(n^w) = 2^{O(\tau m \log n)}$ であるから, $\tau = o(n/m \log n^2)$ であれば, BKW アルゴリズムよりも効率が良い.

4.3.3　SD 問題を経由するアルゴリズム

$\mathsf{LPN}_{n,m,\tau}$ に対応するシンドローム復号問題を考える. 対応するシンドローム復号問題での重みを w とする.

この問題を総当りで解く場合には, 重みが w の m 次元ベクトル $\vec{e}$ を列挙すればよい. そのため, 時間計算量は $O(\binom{m}{w})$ となる.

より効率的な手法として, “Information set decoding” と呼ばれる手法が McEliece [McE78] によって提案されている. 近年その高速化が進んでおり, 時間計算量は $2^{m/20}$ にまで引き下げられている. Becker, Joux, May, Meurer [BJMM12] らによる評価例を表 4.2 に示す. この表は, 時間計算量を最小化した場合の $R = n/m$ の最悪時についてまとめられている. 問題のパラメータによっては, 表の数値よりも速く解くことが可能となる.

パラメータ設定によっては, $\mathsf{LPN}_{n,m,\tau}$ 問題を $\mathsf{SD}_{m-n,m,w}$ 問題に置き換えることで, これらの SD 問題用アルゴリズムも検討する必要がある.

4.3.4 量子アルゴリズムへの耐性

現在のところ多項式時間で LPN 問題を解く量子アルゴリズムは提案されていない. [BJLM13] などで一定の高速化は行われているため, 今後も継続して注視する必要がある.

4.4 まとめ

LPN 問題は学習理論や符号理論から派生した問題である. 誤り確率 τ が十分大きい場合の LPN 問題を多項式時間で効率的に解くことは困難であると予想されている.

共通鍵や公開鍵の分野で多くの方式が LPN 問題に基づいて提案されている. LWE 問題と比較した場合, 利点としては,

- $\mathbb{F}_2$ およびその拡大体を基に構成するため、ハードウェア構成との相性が良い点
- 誤差分布としてベルヌーイ分布やその一般化した分布を用いるため、誤差のサンプリングが容易である点

が挙げられる. 一方、欠点として,

- 鍵や暗号文のサイズが大きくなりやすい点
- ID ベース暗号や完全準同型暗号といった発展的な応用が少ない点

が挙げられる.

暗号方式のパラメータ設定の際には, 4.3 節で挙げたさまざまなアルゴリズムを考慮する必要がある. アルゴリズムの高速化について盛んに研究されており、動向を注視する必要がある. また、攻撃に用いられるアルゴリズムの研究は理論的なものが多く, 攻撃実験報告は小さいパラメータに対して行ったものが多い. そのため, 攻撃実験に関する研究もこれから非常に重要である.

第 4 章の参照文献

[Ale11] Michael Alekhnovich, “More on average case vs approximation complexity,” Computational Complexity 20(4): 755-786 (2011).

[App13] Benny Applebaum, “Garbling XOR gates “For Free” in the standard model,” TCC 2013: 162-181.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai, “Fast cryptographic primitives and circular-secure encryption based on hard learning problems,” CRYPTO 2009: 595-618.

[AHI11] Benny Applebaum, Danny Harnik, and Yuval Ishai, “Semantic security under related-key attacks and applications,” ICS 2011: 45-60.

[AG11] Sanjeev Arora and Rong Ge, “New algorithms for learning in presence of errors,” ICALP (1) 2011: 403-415.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer, “Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding,” EUROCRYPT 2012: 520-536.

[BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg, “On the inherent intractability of certain coding problems,” IEEE Transactions on Information Theory 24(3): 384-386 (1978).

[BJLM13] Daniel J. Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer, “Quantum algorithms for the subset-sum problem,” PQCrypto 2013: 16-33.

[BLP10] Daniel J. Bernstein, Tanja Lange, and Christiane Peters, “Wild McEliece,” SAC 2010: 143-158.

[BLP11a] Daniel J. Bernstein, Tanja Lange, and Christiane Peters, “Wild McEliece incognito,” PQCrypto 2011: 244-254.

[BLP11b] Daniel J. Bernstein, Tanja Lange, and Christiane Peters, “Smaller decoding exponents: Ball-collision decoding,” CRYPTO 2011: 743-760.

[BL12] Daniel J. Bernstein and Tanja Lange, “Never trust a bunny,” RFIDSec 2012: 137-148.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton, “Cryptographic primitives based on hard learning problems,” CRYPTO 1993: 278-291.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman, “Noise-tolerant learning, the parity problem, and the statistical query model,” J. ACM 50(4): 506-519 (2003).

[DP12] Ivan Damgård and Sunoo Park, “Is public-key encryption based on LPN practical?” IACR Cryptology ePrint Archive 2012: 699 (2012). 20140126:205953 版.

[DvdGMQN12] Rafael Dowsley, Jeroen van de Graaf, Jorn Muller-Quade, and Anderson C. A. Nascimento, “Oblivious Transfer Based on the McEliece Assumptions,” IEICE Transactions 95-A(2): 567-575 (2012).

[DNdS12] Bernardo Machado David, Anderson C. A. Nascimento, and Rafael T. de Sousa Jr., “Efficient Fully

Simulatable Oblivious Transfer from the McEliece Assumptions," IEICE Transactions 95-A(11): 2059-2066 (2012).

[DNMQ12] Bernardo Machado David, Anderson C. A. Nascimento, and Jorn Müller-Quade, "Universally Composable Oblivious Transfer from Lossy Encryption and the McEliece Assumptions," ICITS 2012: 80-99.

[DMQN12] Nico Döttling, Jorn Müller-Quade, and Anderson C. A. Nasciment, "IND-CCA secure cryptography based on a variant of the LPN problem," ASIACRYPT 2012: 485-503.

[DDMQN12] Nico Döttling, Rafael Dowsley, Jorn Müller-Quade, and Anderson C. A. Nasciment, "A CCA2 secure variant of the McEliece cryptosystem," IEEE Transactions on Information Theory 58(10): 6672-6680 (2012).

[FGOPT13] Jean-Charles Faugère, Valérie Gauthier-Umaña, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich, "A distinguisher for high-rate McEliece cryptosystems," IEEE Transactions on Information Theory 59(10): 6830-6844 (2013).

[FS96] Jean-Bernard Fischer and Jacques Stern, "An efficient pseudo-random generator provably as secure as syndrome decoding," EUROCRYPT 1996: 245-255.

[FGK+13] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev, "More constructions of lossy and correlation-secure trapdoor functions," J. Cryptology 26(1): 39-74 (2013).

[GRS08] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin, "HB$^{\#}$: Increasing the security and efficiency of HB+," EUROCRYPT 2008: 361-378.

[GRS08] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin, "How to encrypt with the LPN problem," ICALP (2) 2008: 679-690.

[GJL14] Qian Guo, Thomas Johansson, Carl Löndahl, "Solving LPN Using Covering Codes," ASIACRYPT (1) 2014: 1-20.

[Hås01] Johan Håstad, "Some optimal inapproximability results," J. ACM 48(4): 798-859 (2001).

[HKL+12] Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak, "Lapin: An efficient authentication protocol based on ring-LPN," FSE 2012: 346-365.

[HB01] Nicholas J. Hopper and Manuel Blum, "Secure human identification protocols," ASIACRYPT 2001: 52-66.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes, "Commitments and efficient zero-knowledge proofs from learning parity with noise," ASIACRYPT 2012: 663-680.

[KK15] 上中谷 健, 國廣 昇 "LPN 問題に対する BKW アルゴリズムの拡張," SCIS2015, 3E1-3, 2015.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith, "Parallel and concurrent security of the HB and HB+ protocols," J. Cryptology 23(3): 402-421 (2010).

[KPC+11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. "Efficient authentication from hard learning problems," EUROCRYPT 2011: 7-26.

[Kir11] Paul Kirchner, "Improved generalized birthday attack," IACR Cryptology ePrint Archive 2011: 377 (2011).

[KI01] Kazukuni Kobara and Hideki Imai, "Semantically secure McEliece public-key cryptosystems – conversions for McEliece PKC," PKC 2001: 19-35.

[LB88] Pil Joong Lee and Ernest F. Brickell, "An observation on the security of McEliece's public-key cryp-

tosystem," EUROCRYPT 1988: 275-280.

[LF06] Éric Levieil and Pierre-Alain Fouque, "An improved LPN algorithm," SCN 2006: 348-359.

[LDW94] Yuan Xing Li, Robert H. Deng, and Xin Mei Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," IEEE Transactions on Information Theory 40(1): 271-273 (1994).

[Lyu05] Vadim Lyubashevsky, "The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem," APPROX-RANDOM 2005: 378-389

[MVVR12] K. Preetha Mathew, Sachin Vasant, Sridhar Venkatesan, and C. Pandu Rangan, "An efficient IND-CCA2 secure variant of the Niederreiter encryption scheme in the standard model," ACIPS 2012: 166-179.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae, "Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$," ASIACRYPT 2011: 107-124.

[McE78] Robert J. McEliece, "A public-key cryptosystem based on algebraic coding theory," Jet Propulsion Laboratory DSN Progress Report 42-44: 114-116 (1978).

[Nie86] Harald Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory 15: 19-34. Problemy Upravleniia i Teorii Informatsii 15: 159-166 (1986).

[NIKM08] Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov, "Semantic security for the McEliece cryptosystem without random oracles," Designs, Codes and Cryptography 49: 289-305 (2008).

[Per13] Edoardo Persichetti, "Improving the Efficiency of Code-Based Cryptography," University of Auckland, 2013.

[Pie12a] Krzysztof Pietrzak, "Subspace LWE," TCC 2012: 548-563.

[Pie12b] Krzysztof Pietrzak, "Cryptography from learning parity with noise," SOFSEM 2012: 99-114.

[Reg09] Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography," J. ACM, 56(6): 1–40 (2009).

[Ste88] Jacques Stern, "A method for finding codewords of small weight," Coding Theory and Applications 1988: 106-113.

第 5 章

Approximate Common Divisor 問題

5.1 Approximate Common Divisor 問題の概説

5.1.1 Approximate Common Divisor 問題とは

Approximate Common Divisor 問題 (ACDP) は, CaLC2001 において, Howgrave-Graham により, 導入された問題である [HG01]. いくつかの暗号方式の安全性評価が, この問題を経由することにより行われている. Approximate Common Divisor (ACD) 問題は, 次のように定式化される.

定義 5.1 (ACD 問題 (その 1)) p を未知の素数とし, p の倍数 N は, 既知であるとする. r を, その絶対値が N^α 以下の整数とする. q を N/p 程度の乱数として,

$$x = pq + r$$

とする. x が与えられた時に, r を求める問題である.

この問題に対して, 法を p として簡約したものを考えることが多い. すなわち, 次の問題を ACD 問題とみなすことも多い.

定義 5.2 (ACD 問題 (その 2)) N を合成数として, p は, N の未知の素因数とする. ただし, $p \approx N^\beta$ とする. a を与えられた整数として,

$$a + x \equiv 0 \pmod{p}$$

をみたす x を求める問題である. ただし, $\alpha \leq \beta$ に対して, 解 x は, $|x| < N^\alpha$ を満たしているとする.

5.1.2 Approximate Common Divisor 問題の拡張

ACD 問題は, いくつかの拡張問題を持つ. ここでは, 以下の問題を考える.

定義 5.3 (複数 ACD 問題 (その 1)[CMNT11]) p を未知の素数とする. q を十分大きい自然数として, $N = pq$ とする. この N は既知であるとする. r_i を絶対値が N^α 以下の整数とする. q_i を q 程度の乱数として,

$$\begin{cases} x_1 &= pq_1 + r_1 \\ x_2 &= pq_2 + r_2 \\ &\vdots \\ x_n &= pq_n + r_n \end{cases}$$

とする. $x_1, x_2, \ldots, x_n$ が与えられた時に, $r_1, r_2, \ldots, r_n$ を求める問題である.

同様に, 以下のようにも定式化される.

定義 5.4 (複数 ACD 問題 (その２)) N を合成数として, p は, N の未知の素因数とする. ただし, $p \approx N^{\beta}$ とする. $a_1, a_2, \ldots, a_n$ を与えられた整数として,

$$\begin{cases} a_1 + x_1 & \equiv 0 \pmod{p} \\ a_2 + x_2 & \equiv 0 \pmod{p} \\ & \vdots \\ a_n + x_n & \equiv 0 \pmod{p} \end{cases}$$

をみたす $x_1, x_2, \ldots, x_n$ を求める問題である. ただし $\alpha_1, \alpha_2, \ldots, \alpha_n \leq \beta$ となる α_i に対して, 解 $x_1, x_2, \ldots, x_n$ は, $|x_i| < N^{\alpha_i}$ を満たしているとする. 簡単のため, $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$ であるとする.

N が与えられない問題を, General ACD 問題 (GACD 問題) と呼ぶ. この問題と区別するため, N が与えられる問題を Partial ACD 問題 (PACD 問題) と呼ぶこともある. 明らかに, 同一の n に対して, GACD 問題の方が, PACD 問題よりも困難である. 複数 GACD 問題は, 以下のように定義される.

定義 5.5 (複数 GACD 問題) p を未知の素数, N を γ ビットの自然数として, $p \approx N^{\beta}$ とする. q_i を 0 から N/p の間の乱数とし, r_i を絶対値が N^{α} 以下の整数とする. $x_1, x_2, \ldots, x_n$ を

$$\begin{cases} x_1 & = pq_1 + r_1 \\ x_2 & = pq_2 + r_2 \\ & \vdots \\ x_n & = pq_n + r_n \end{cases}$$

とする. $x_1, x_2, \ldots, x_n$ が与えられた時に, $r_1, r_2, \ldots, r_n$ を求める問題である.

5.1.3 Approximate Common Divisor 問題のアプリケーション

van Dijk ら [DGHV10] は, 複数 GACD 問題の困難さを安全性の根拠として持つ, 整数上での完全準同型暗号を提案している. さらに, 仮定を複数 PACD 問題の困難さに強めることにより, 効率的になることを述べている. ついで, Coron らは, 公開鍵サイズを削減する方式を提案している [CMNT11]. 彼らの方式も, 複数 PACD 問題の困難さを安全性の根拠としている. さらに, [CCK+13] では, 中国人の剰余定理を用いる事により, バッチ処理が可能な方式を提案している. この論文では, 新たに, 判定 Approximate GCD 問題を導入し, この問題の困難さを安全性の根拠とした方式を提案している. さらに, 提案方式をベースに, 128 ビット AES 回路の実装を行っている. 72 ビットセキュリティを担保した上で, 13 分以内で, 暗号化の処理が終了すると報告している. この論文では, 後に述べる [CN11] による攻撃を考慮した上で, パラメタ設定を行っている.

以下, 順に, van Dijk らの方式 [DGHV10], Cheon らの方式 [CCK+13] を説明する. ただし, 記述を容易にするため, 完全準同型方式ではなく, somewhat 準同型方式を記載する.

5.1.3.1 van Dijk らの方式 [DGHV10]

正の奇数 p に対して, 以下のように, γ ビットの整数上の分布 $\mathcal{D}_{\gamma,\rho}(p)$ を導入する.

$$\mathcal{D}_{\gamma,\rho}(p) = \{ \text{ choose } q \leftarrow \mathbb{Z} \cap [0, 2^{\gamma}/p), r \leftarrow \mathbb{Z} \cap (-2^{\rho}, 2^{\rho}) : \text{ output } x = pq + r \}$$

KeyGen(λ)

秘密鍵： η ビットの奇数 p

公開鍵： x_i を $D_{\gamma,\rho}(p)$ から $\tau+1$ 個取り, それらを $x_0, x_1, \ldots, x_\tau$ とする. ただし, x_0 が最大とする. x_0 は奇数で, $x_0 \bmod p$ は偶数であるとし, そうでなければ, あらためて, x_i を取り直す. 公開鍵 pk は, $(x_0, x_1, \ldots, x_\tau)$ である.

Enc($pk, m \in \{0,1\}$)

Step1　ランダムな部分集合 $S \subseteq \{1, 2, \ldots, \tau\}$ を選ぶ.

Step2　$r \leftarrow \mathbb{Z} \cap (-2^{\rho'}, 2^{\rho'})$ を選ぶ.

Step3　暗号文

$$c \leftarrow (m + 2r + 2\sum_{i \in S} x_i) \bmod x_0$$

とする.

Dec(sk, c)

$m' \leftarrow (c \bmod p) \bmod 2$ を計算し, m' を出力する.

5.1.3.2　CCK+13 方式 [CCK+13]

簡単のため, 論文中メッセージ空間は, 二進系列の場合のみを記述する. 一般の場合の記述は, [CCK+13] を参照されたい.

KeyGen(λ)

秘密鍵： η ビットの異なる奇数 $p_0, p_1, \ldots, p_{l-1}$

公開鍵： $\pi = \prod_{i=0}^{l-1} p_i$ とする. q_0 を, 0 から $2^\gamma/\pi$ の間の整数をランダムに選び, $x_0 = q_0\pi$ とする. ただし, q_0 は, 2^{λ^2}-rough であるとする.

$$x_i \bmod p_j = 2r_{i,j}, r_{i,j} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) \text{ for } 1 \le i \le \tau$$
$$x'_i \bmod p_j = 2r'_{i,j} + \delta_{i,j}, r'_{i,j} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) \text{ for } 0 \le i \le l-1$$

ここで, $\delta_{i,j}$ は, $i = j$ のときに, 1 であり, $i \neq j$ とき, 0 を取る. 公開鍵 pk は, $(x_0, x_1, \ldots, x_\tau, x'_0, x'_1, \ldots, x'_{l-1})$ である.

Enc($pk, \boldsymbol{m} = (m_0, m_1, \ldots, m_{l-1}) \in \{0,1\}^l$)

Step1　$\boldsymbol{b} = (b_1, b_2, \ldots, b_\tau) \in \{0,1\}^\tau$ をランダムに選ぶ.

Step2　暗号文

$$c \leftarrow (\sum_{i=0}^{l-1} m_i x'_i + \sum_{i=1}^{\tau} b_i x_i) \bmod x_0$$

とする.

Dec(sk, c)

$m_j \leftarrow (c \bmod p_j) \bmod 2$ を計算し, $\boldsymbol{m} = (m_0, m_1, \ldots, m_{l-1})$ を出力する.

5.1.4 安全性の根拠となる問題

[DGHV10] では, Approximate GCD 問題を次のように定義している. (ρ, η, γ)-approximate GCD 問題とは, ランダムに選ばれた η ビットの奇数 p に対して, $\mathcal{D}_{\gamma,\rho}(p)$ からの多項式個のサンプルが与えられた時に, p を求める問題である.

[DGHV10] で提案された somewhat 準同型暗号方式の安全性は, 以下のように示されている. ここで, 用いるパラメタを $(\rho, \rho', \eta, \gamma, \tau)$ とする. このとき, advantage ϵ で方式を破る攻撃者 A は, (ρ, η, γ)-approximate GCD 問題を, 確率 $\epsilon/2$ 以上で, 解くアルゴリズム B に変換することができる. アルゴリズム B の動作時間は, A の動作時間, $\lambda, 1/\epsilon$ の多項式である.

[CMNT11] では, Error-free Approximate GCD 問題を次のように定義している. 正の奇数 p, q_0 に対して, 整数上の分布 $\mathcal{D}'_\rho(p, q_0)$ を, 次のように定義する.

$$\mathcal{D}'_\rho(p, q_0) = \{ \text{ choose } q \leftarrow \mathbb{Z} \cap [0, q_0), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : \text{ output } x = pq + r\}$$

(ρ, η, γ)-error-free approximate GCD 問題とは, ランダムに選ばれた η ビットの奇数 p とランダムに選ばれた square-free かつ 2^λ-rough で, 0 から $2^\gamma/p$ の間の整数 q_0 に対して, $x_0 = q_0 p$ と $\mathcal{D}'_\rho(p, q_0)$ からの多項式的に多くのサンプルが与えられた時に, p を求める問題である.

[CMNT11] で提案された somewhat 準同型暗号方式の安全性は, 以下のように示されている. ここで, 用いるパラメタを $(\rho, \rho', \eta, \gamma, \tau)$ とする. このとき, advantage ϵ で方式を破る攻撃者 A は, (ρ, η, γ)-error-free approximate GCD 問題を, 確率 $\epsilon/2$ 以上で, 解くアルゴリズム B に変換することができる. アルゴリズム B の動作時間は, A の動作時間, $\lambda, 1/\epsilon$ の多項式である.

[CCK+13] で提案された somewhat 準同型暗号方式は, 判定 Approximate GCD 問題の困難さに安全性の根拠をおいている. この問題は, 以下のように定式化される.

1. $\mathcal{D}'_\rho(p, q_0)$ から多項式個のサンプルを受け取った上で,
2. $z = x + rb \bmod x_0$ を受け取った時に, $b \in \{0, 1\}$ を判定する問題である. ここで, $x \leftarrow \mathcal{D}'_\rho(p, q_0)$ と $r \leftarrow \mathbb{Z} \cap [0, x_0)$ である.

以上の記述では, 原論文での記述を採用している. そのため, 用いるパラメタが異なっているが, $\eta = \beta \log N, \rho = \alpha \log N, \gamma = \log N$ という関係にあることに注意されたい.

5.2 ACD 問題に対する評価

PACD 問題は, N の素因数分解を経由することにより, 容易に解くことができる. 具体的には, 以下の手順による. まず, N を素因数分解をすることにより, p を求める. 求めた p を用いることにより, x を求めることができる. 法が既知の一次方程式 $a + x \equiv 0 \pmod{p}$ を解くことは, 容易であるためである. これ以降, N の素因数分解を, 直接的には, 経由しないアルゴリズムを考察する.

N の素因数分解を直接的には経由しないアルゴリズムを, 以下の二つに大別して説明をする.

1. 組み合わせ論に基づくアルゴリズム
2. 格子理論に基づくアルゴリズム

前者のアルゴリズムは, 指数関数時間アルゴリズムではあるが, 解に制約は存在しない. すなわち, どのような α に対

しても, 解を求めることが可能である. しかし, 計算量は, α に依存する. その一方で, 後者のアルゴリズムは, 解くことができる解に制約が存在するものの, 解がその制約をみたせば, 多項式時間で求解が可能である. すなわち, 任意の α に対して, 解を求めることができる訳ではなく, 制限が存在するが, 十分高速に解を求めることができる. そのため, 求める問題に応じて, 適切なアルゴリズムの選択が重要である.

5.2.1 組み合わせ論に基づくアルゴリズム

PACD 問題を解く最も素朴なアルゴリズムは, 全数探索アルゴリズムである. 解 x の可能な値は, $2N^{\alpha}$ 個であるので, 全数探索により, $\tilde{O}(N^{\alpha})$ の計算量で解の探索が可能である. これは, ビット長 $\log N$ に対して, 指数関数時間必要である.

Chen と Nguyen は, 全数探索よりも効率的に, 解を求めるアルゴリズムを提案している [CN11]. 彼らは, multipoint evaluation of univariate polynomials というテクニックを導入することにより, 効率化に成功している. まず, このテクニックについて説明する. 整数係数でモニックな 1 変数 n 次多項式 $f(x)$ を考える. $a_1, a_2, \ldots, a_n$ を整数として, $f(a_1), f(a_2), \ldots, f(a_n)$ の値全てを計算したい状況を考える. 素朴なアルゴリズムでは, この計算には, $O(n^2)$ の計算量が必要である. これに対して, 彼らは, $\tilde{O}(n)$ の計算量で, $f(a_1), f(a_2), \ldots, f(a_n)$ の全てを計算するアルゴリズムを提案している. すなわち, 平方根の高速化が実現している. 彼らは, PACD 問題を, multipoint evaluation of univariate polynomials に帰着した上で, このアルゴリズムを適用することにより, PACD 問題を解くアルゴリズムを構成している. 実際の計算量は,

$$\tilde{O}(N^{\alpha/2})$$

で与えられる.

Chen と Nguyen[CN11] は, 提案アルゴリズムを実装することにより, Coron らの論文 [CMNT11] 中で提示された推奨パラメタに対して, 安全性の再評価を行っている. 再評価結果を表 5.1 に記す. 表中, 「Securtiy Level」の欄は, 総当たりの攻撃により, 見積もられた Security Level である. その一方で, 「新しい Security Level」の欄は, Chen-Nguyen の攻撃により見積もられた Security Level である. 従来の見積もりよりも, 安全性が低下していることが確認できる.

表 5.1 Chen–Nguyen アルゴリズムによる評価 ([CN11] より)

Name	Toy	Small	Medium		Large	
Security Level	52	61	72		100	
計算時間の見積もり	1.6 分	7.1 時間	190 日	76 日	2153 年	9 年
使用メモリ量	$\leq$ 130 Mb	$\leq$ 15 Gb	$\leq$ 72 Gb	$\approx$ 240 Gb	$\leq$ 72 Gb	$\approx$ 25 Tb
新しい Security Level	$\leq$ 37.7	$\leq$ 45.7	$\leq$ 55	$\leq$ 54	$\leq$ 67	$\leq$ 59

5.2.2 格子理論に基づくアルゴリズム

一般に, 暗号の安全性解析において, 格子理論にもとづくアルゴリズム [Cop95, Cop96, Cop97, HG97] は, 重要なツールである. ここでは, 格子理論を用いた ACD 問題を解くアルゴリズムについて説明する. Partial ACD 問題を解く格子理論に基づくアルゴリズムの中で, 現状で最も優れたアルゴリズムは, Howgrave-Graham によるアルゴリズムである [HG01]. このアルゴリズムでは, α と β が,

$$\alpha < \beta^2 \tag{5.1}$$

を満たすときに, 多項式時間で解を求めることが可能である.

この結果を用いると, よく知られた以下の結果を, 容易に導くことができる [Cop96:A].

RSA タイプの合成数 $N = pq$ に対して, p の上位半分のビットがわかった時に, 素因数分解が可能である.

RSA 型の合成数 $N = pq$ に対して, p の近似値 $\tilde{p}$ がわかった場合を考える. $x = p - \tilde{p}$ とおくと, $\tilde{p} + x \equiv 0 \pmod{p}$ が成り立つ. このため, PACD 問題が解ければ, 素因数分解が可能となる. $p \approx N^{1/2}$ の時, すなわち, $|p - \tilde{p}| < N^{1/4}$ の時には, 素因数分解が可能となる. 具体的には, p の上位半分がわかれば, 素因数分解が可能である.

5.2.3 量子アルゴリズムへの耐性

前述のように, Partial ACD 問題は, N の素因数分解ができれば, 簡単に解くことができる. 量子計算機を用いることができれば, Shor のアルゴリズム [Shor94] により, 多項式時間で素因数分解を行うことができるため, PACD 問題を解くことは容易である.

5.2.4 ACD 問題に対する評価のまとめ

以上の議論をまとめる. Partial ACD 問題は,

1. 解の大きさに $\alpha < \beta^2$ という制限がある場合には, 多項式時間で解くことができる.
2. その一方で, 解の大きさに制限がない場合には, $\tilde{O}(N^{\alpha/2})$ の計算量で解を求めることが可能である.

問題の設定により, 最適なアルゴリズムが異なるため, 適切な選択が必要である.

5.3 複数 ACD 問題に対する評価

5.3.1 組み合わせ論に基づくアルゴリズム

複数 ACD 問題に対しても, 最も素朴なアルゴリズムは, 全数探索アルゴリズムである. 解 $x_1, x_2, \ldots, x_n$ のうち, 一つでも値を求めることができれば, p を求めることができるため, $x_1, x_2, \ldots, x_n$ の全てを求めることが可能である. このため, x_1 をまず求めることにする. このとき, x_1 の取りうる値の可能な個数は, $2N^{\alpha_1}$ である. そのため, 複数 ACD 問題を全数探索アルゴリズムにより解く計算量は, $\tilde{O}(N^{\alpha_1})$ で与えられる.

同様に, Chen-Nguyen のアルゴリズム [CN11] により, $\tilde{O}(N^{\alpha_1/2})$ の計算量で, この問題を解くことができる. このアルゴリズムでは, 複数の方程式が与えられていることを有効に活用できていない.

5.3.2 格子理論に基づくアルゴリズム

5.3.2.1 Coppersmith 流のアルゴリズム

格子理論に基づくアルゴリズムにより, 複数 ACD 問題を多項式時間で解くことができる条件を示す. 前述の Howgrave-Graham アルゴリズム [HG01] を用いることにより, $\alpha_1 < \beta^2$ であれば, 解を求めることができる. このアルゴリズムでも, 方程式が複数個得られていることを活用していない.

ANTS2012 において, Cohn と Heninger は, $\beta \gg \frac{1}{\sqrt{\log N}}$ という条件下で,

$$\frac{\alpha_1 + \alpha_2 + \cdots + \alpha_n}{n} < \beta^{(n+1)/n}$$

の時に, 多項式時間で解を求めるこができることを示している [CH11]. 各 α_i が, 全て等しく α であるとする. このと

き, $\alpha < \beta^{(n+1)/n}$ の時に, 解を求めることができる.

その一方で, Cohn と Heninger の結果は, α_i が等しく無い場合には, 必ずしも最適ではない. これに対して, Takayasu と Kunihiro は, 解くことができる条件の改良を行っている [TK13]. 彼らは,

$$\sqrt[n]{\alpha_1\alpha_2\cdots\alpha_n} < \beta^{(n+1)/n}$$

の時に多項式時間で解を全て求めることができることを示している. 常に,

$$\frac{\alpha_1+\alpha_2+\cdots+\alpha_n}{n} \geq \sqrt[n]{\alpha_1\alpha_2\cdots\alpha_n}$$

が成り立つため, 彼らの条件は, Cohn–Heninger の条件の改良となっている. ただし, 各 α_i が, 全て等しく α であるときには, この条件は, $\alpha < \beta^{(n+1)/n}$ となり, Cohn–Heninger の結果と一致する.

この結果の妥当性を検証する. $\alpha_1 = \beta^2$ であれば, $2 \leq i \leq n$ となる i に対して, $\alpha_i = \beta$ と取ることができるため, Takayasu–Kunihiro の結果は, Howgrave-Graham の結果の自然な拡張となっている.

5.4 GACD 問題の格子理論を用いたアルゴリズム

GACD 問題を解くアルゴリズムに関して, 議論する.

5.4.1 組み合わせ論に基づくアルゴリズム

Chen と Nguyen は, PACD 問題を解くアルゴリズムを拡張により, General ACD 問題を, $\tilde{O}(N^{3\alpha/2})$ で解くことができることを示している [CN11]. 総当たりのアルゴリズムでは, $\tilde{O}(N^{2\alpha})$ の計算量が必要であるため, 指数関数の高速化を実現している. このとき, 必要となるメモリ量は, $\tilde{O}(N^{\alpha/2})$ である.

Chen と Nguyen のアルゴリズムは, GACD の 2 個のサンプルしか用いていないが, 積極的に複数個のサンプルを用いることにより計算量の削減が可能である. Coron らは, [CNT12] において, 計算量 $\tilde{O}(N^{\alpha})$, メモリ量 $\tilde{O}(N^{\alpha})$ のアルゴリズムを提案している.

5.4.2 格子理論に基づくアルゴリズム

次に, 格子理論に基づくアルゴリズムを述べる, Coppersmith の手法に基づくように, 十分大きい法に対して成り立つ関係式を用いて, 法を外し, 整数上の方程式に変換してから解く方法と, 解を最短ベクトルに埋め込むことにより解く方法を紹介する. この二つの方法の一般論に関しては, [K11] に詳しい.

5.4.2.1 Coppersmith の手法に基づく解析

Howgrave-Graham は, $n = 2$ の時の解析を行っている [HG01]. $n = 2, \alpha_1 = \alpha_2 := \alpha$ の時は,

$$\alpha < 1 - \frac{1}{2}\beta - \sqrt{1 - \beta - \frac{1}{2}\beta^2}$$

であれば, 解を求めることができることを示している. 一般の n の状況に関しては, Cohn と Heninger は,

$$\alpha < \frac{1 - 1/n^2}{n^{1/(n-1)}}\beta^{n/(n-1)}$$

のときに, 多項式時間で解を求めることができることを示している [CH11].

5.4.2.2 最短ベクトルに埋め込む解法

次に, 解きたい解を短いベクトルに埋め込む手法を用いた場合の解析について説明する. [DGHV10] では, Lagarias の同時 Diophantine 近似 (SDA) 問題を解くアルゴリズムを利用することにより, 複数 ACD 問題が難しくなるかを評価している. 今, サンプルは, $t+1$ 個用いるとする. $t+1 < \gamma/\eta$ の時には, 解を埋め込んだベクトルが最短ベクトルにならないことを指摘している. そのため, LLL アルゴリズムなど格子簡約アルゴリズムなどを用いても, 解を見つけることができない. その一方で, t が大きいときには, 埋め込んだベクトルが最短になりやすくなる. しかし, この場合, 用いる格子の次元が大きくなりすぎるため, 効率的に解を求めることができない. 経験的に, 最短ベクトルの 2^k の近似精度でベクトルを求めるためには, $2^{t/k}$ の計算時間が必要である. そのため, $t \geq \gamma/\eta$ の時には, 2^η の近似精度を実現するためには, およそ $2^{\gamma/\eta^2}$ の計算時間が必要である. そのため, γ/η^2 を $\log \lambda$ 程度に設定をすれば, 全体の計算時間は指数関数時間になる.

さらに, [DGHV10] では, Nguyen と Stern による orthogonal 格子を用いた場合の解析も行っている. SDA 問題を経由するときと同様に, 解を求めるためには, $2^{\gamma/\eta^2}$ 程度の計算量が必要であることを述べている.

5.4.3 完全準同型暗号の安全性への影響

いずれの攻撃においても, 適切にパラメタが設定された状況では, 攻撃に成功するのに, 指数関数時間が必要であり, 脆弱性は発見されていない. しかし, いずれも, 理論上の解析であるため, 数値実験により安全性の検証をする必要がある.

5.5 関連問題 co-ACD 問題の安全性評価

Cheon らは, ACM CCS2014 において, ACD 問題の関連問題として, co-ACD 問題を導入し, この問題の困難さに安全性の根拠をおく加法準同型暗号を提案している [CLS14]. この加法準同型暗号方式は, 同様の機能を持つ Paillier 暗号と比べて, 高速に演算が可能であるという性質を持つ. さらに, co-ACD 問題の安全性を議論し, ACD 問題に対するアルゴリズムを適用した場合には, 十分, 安全であることを示している.

以下に, co-ACD 問題の定義を記す. まず, 分布 $\hat{\mathcal{D}}_{\rho,Q}$ を, 以下のように定義する. 素数 $(p_1, p_2, \ldots, p_k)$ として, $e \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$ とし,

$$(eQ \bmod p_1, eQ \bmod p_2, \ldots, eQ \bmod p_k)$$

を出力する. 計算 co-ACD 問題は, $\hat{\mathcal{D}}_{\rho,Q}$ からの多項式個のサンプルが与えられたときに, $\prod_{i=1}^{k} p_i$ の非自明な因数を求める問題である.

しかし, 最近になり, co-ACD 問題に特化した攻撃手法が提案されている [FLT15]. [FLT15] は, 短い平文に対する暗号文を複数得られた状況で, Nguyen-Stern の直交格子解読手法, グレブナー基底手法, Coppersmith アルゴリズムを用いることにより, 効率的に平文の復元が可能であると主張している.

5.6 まとめ

この節の議論をまとめる. 現状において, ACD 問題は, パラメタを適切に選ぶ事により, 現実的な時間で解を求めることは不可能である. つまり, 法に対して, 解が, ある制限よりも小さいときには, 多項式時間で解くことができるものの, その一方で, 解が十分大きいときには, 解を求めることができない. 組み合わせ論に基づくアルゴリズムを用いた場

合では, 依然, 指数関数時間の計算量が必要であるが, 全数探索アルゴリズムの平方根の計算量で解を求めることができる. Chen–Nguyen のアルゴリズムは, 暗号の提案時には, 考慮されていなかった攻撃であり, 実際に, 提案論文で書かれた推奨パラメタのいくつかは, 解読されることが示されている. また, ACD 問題に関連した問題 co-ACD 問題は, 当初の想定よりも弱いことが明らかになっている. これらの結果は, ごく最近に示されたものであり, 今後の研究の動向に注視する必要がある.

第 5 章の参照文献

[CN11] Y. Chen and P. Q.Nguyen, “Faster Algorithms for Approximate Common Divisors: Breaking Fully-Homomorphic-Encryption Challenges over the Integers,” EUROCRYPT 2012, LNCS 7237, pp 502–519, 2012.

[CCK+13] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi and A. Yun, “Batch Fully Homomorphic Encryption over Integers,” EUROCRYPT 2013, LNCS 7881, pp 315–335, 2013.

[CLS14] J. H. Cheon, H. T. Lee and J. H. Seo. “A New Additive Homomorphic Encryption based on the co-ACD Problem,” ACM CCS2014, pp. 287–298, 2014.

[CH11] H. Cohn, N. Heninger, “Approximate common divisors via lattices,” Proc. of ANTS 2012.

[Cop95] D. Coppersmith, “Factoring with a hint,” IBM Research Report RC 19905, 1995.

[Cop96] D. Coppersmith, “Finding a Small Root of a Univariate Modular Equation,” Advances in Cryptology – Eurocrypt ’96, LNCS 1070, Springer-Verlag, pp. 155–165, 1996.

[Cop96:A] D. Coppersmith, “Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known,” Advances in Cryptology – Eurocrypt ’96, LNCS 1070, Springer-Verlag, pp. 178–189, 1996.

[Cop97] D. Coppersmith, “Small solutions to polynomial equations, and low exponent RSA vulnerabilities,” Journal of Cryptology 10: 233, 260, 1997.

[CMNT11] J. -S. Coron, A. Mandal, D. Naccache, M. Tibouchi, “Fully homomorphic encryption over the integers with shorter public keys,” CRYPTO 2011, LNCS 6841, pp. 487–504, Springer-Verlag, 2011.

[CNT12] J. -S. Coron, D. Naccache, M. Tibouchi, “Public key compression and modulus switching for fully homomorphic encryption over the integers,” EUROCRYPT 2012, LNCS 7237, pp. 446–464, Springer-Verlag, 2012.

[DGHV10] M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan, “Fully homomorphic encryption over the integers,” Proc. of Eurocrypt 2004, pp. 14–27. LNCS 6110. Springer-Verlag, Berlin, Heideberg , 2010. Longer version available as Report 2009/616 in the Cryptology ePrint Archive(http://eprint.iacr.org/2009/616).

[FLT15] ピエール=アラン・フーク, タンクレード・ルポワン , メディ・ティブシ, “Co-ACD 仮定とそれを基にした準同型暗号方式の安全性評価, ” SCIS2015, 3E4-4, 2015.

[HG97] N. Howgrave-Graham, “Finding small roots of univariate modular equations revisited,” Proc. of Cryptography and Coding, LNCS 1355, pp. 1331–142, 1997.

[HG01] N. Howgrave-Graham, “Approximate integer common divisors,” Proceedings of CALC 2001, LNCS 2146, pp. 51–66, Springer, 2001.

[K11] 國廣 昇, “格子理論を用いた暗号解読の最近の研究動向,” 電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review, Vol. 5, no. 1, pp. 42-55, 2011.

[Shor94] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proc. of 35nd Annual Symposium on Foundations of Computer Science, pp. 124–134, 1994.

[TK13] A. Takayasu and N. Kunihiro, "Better Lattice Constructions for Solving Multivariate Linear Equations Modulo Unknown Divisors," in Proc. of ACISP2013, LNCS 7959, pp. 118–135, 2013.

資料 1
別添 4

2014 年度 暗号技術調査 WG(軽量暗号)活動報告

1. 活動目的

軽量暗号 WG は、軽量暗号技術が求められるサービスにおいて、電子政府のみならず利用者が適切な暗号方式を選択でき、容易に調達できることをめざして設置された。昨年度は、これまでに提案されている軽量暗号の現状調査、アプリケーションに関する調査、実装評価等を行った。今年度はこれらをふまえてさらに検討を行い、CRYPTREC における今後の活動方針を検討し、暗号技術評価委員会に提言を行う。

2. 委員構成

主査　本間尚文(東北大学)

委員　青木和麻呂(NTT)、岩田哲(名古屋大学)、小川一人(NHK)、
崎山一男(電気通信大学)、渋谷香士(ソニー)、鈴木大輔(三菱電機)、
成吉雄一郎(ルネサスエレクトロニクス)、峯松一彦(日本電気)、三宅秀享(東芝)、
渡辺大(日立製作所)

3. 活動概要

3.1. スケジュール及び審議概要

2014 年 8 月 29 日　第 1 回軽量暗号 WG

- 本年度活動内容の審議・承認
- 軽量暗号に関する議論 (既存暗号に対するアドバンテージ、CRYPTREC で扱う軽量暗号のスコープ、軽量暗号で達成すべき安全性)

「暗号技術調査 WG(軽量暗号)報告書」の更新方法、執筆担当委員などについて合意した。

- 今後の活動方針に関する議論

2014 年 11 月 12 日　第 2 回軽量暗号 WG

- 今年度調査に関する中間報告
- 今後の活動方針に関する議論

A)「暗号技術ガイドライン（軽量暗号の最新動向)」の発行、

B)「暗号技術ガイドライン（軽量暗号の詳細評価)」の発行、

C) 軽量暗号に関する技術公募の実施

などが出されていたが、審議の結果、A) もしくは B) として技術ガイドラインをまとめる方向で進めていくこととなった。

2015 年 2 月 2 日　第 3 回軽量暗号 WG

- 暗号技術評価委員会への報告内容の審議

  「4. 暗号技術評価委員会への報告」に示す提言を暗号技術評価委員会に提出することで合意した。

- 「暗号技術調査 WG(軽量暗号)報告書」の内容確認・議論
- 次年度から開始する詳細評価の対象に関する議論

3.2. 「暗号技術調査 WG(軽量暗号)報告書」各技術分類の執筆担当委員

表 1. 「暗号技術調査 WG(軽量暗号)報告書」各技術分類の執筆担当委員

第 2 章 軽量暗号アルゴリズム調査		
ブロック暗号	安全性	実装性能
	青木 和麻呂 委員	渋谷 香士 委員
ストリーム暗号	渡辺 大 委員	
ハッシュ関数	三宅 秀享 委員	
メッセージ認証コード	渡辺 大 委員	
認証暗号	安全性	実装性能
	峯松 一彦 委員	鈴木 大輔 委員
第 3 章 軽量暗号に関わる新しい技術動向		
低レイテンシ	﨑山 一男 委員	
サイドチャネル攻撃耐性	成吉 雄一郎 委員	
CAESAR プロジェクト	岩田 哲 委員	
軽量暗号の活用事例 および標準化動向	小川 一人 委員	

4. 暗号技術評価委員会への報告

4.1. CRYPTREC で扱う軽量暗号のスコープ

- 軽量暗号 WG では、「実装性能と安全性のトレードオフを勘案した上で、従来の暗号技術に対して特定の性能指標で優位性（軽量性）を持つように設計され

た暗号技術」を軽量暗号のスコープとし、用途が想定される代表的な性能指標に対して優位性を主張する暗号を主な対象とする。

・ 本報告書では共通鍵暗号系の軽量暗号を対象としている。

表 2 ：代表的な性能指標

性能指標		アプリケーションの例
ハードウェア実装	回路規模（消費電力、コスト）	RFID、低コストセンサー
	消費電力量	医療機器、バッテリ駆動デバイス
	レイテンシ（リアルタイム性能）	メモリ暗号化、車載機器、 産業向け I/O デバイス制御
ソフトウェア実装	メモリサイズ（ROM/RAM）	家電機器、センサー、車載機器

4.2. 既存の暗号に対して優位性を持つ分野

① 回路規模

・ 回路規模の視点では、現在提案されている軽量暗号とAESの差は数k gate程度[1]である。

・ ミューチップのようなダイサイズが 50μm 角クラスのチップでは数 k gate の差がクリティカルで、暗号機能の搭載可否に影響を与えうる。500μm 角クラスのチップでも 180nm など古いプロセスにおいては実装可否に影響を与える可能性がある。

② 消費電力量

・ 一般に回路規模が小さいほど消費電力あるいは消費電力量は減る傾向にあり、軽量暗号を利用することで消費電力あるいは消費電力量に関する設計条件を緩和する効果が期待できる。

③ レイテンシ

・ AES に対して 2 倍の応答速度をおよそ 1/10 の回路規模で実現できる軽量暗号が存在する。(20k gate で 10ns 以下での暗号演算が可能。AES では 200k gate で 15ns 要する)

・ 産業向け I/O デバイス制御に代表されるような μs オーダーのリアルタイム性

1 現在、モバイル向け SoC や GPU で主流の 40nm 以下のプロセスでは、数 k gate 程度の回路規模は暗号の優劣の指標とはなりえない。

が求められる用途において、チップへのコストインパクトなしで暗号技術を利用できる。

④ メモリサイズ

- 組込み機器向けソフトウェア実装のプログラムサイズにおいて、AES に対しておよそ 1/4 の ROM サイズで実装可能な軽量暗号が存在する。（産業分野や自動車などで利用されているマイコン RL78 上で 220 バイトで実装可能）
- レガシー製品に暗号機能を搭載する場合、残された ROM 領域に実装する必要があり、軽量暗号でないと搭載できないケースが起こりうる。
- 新規搭載の場合も ROM 領域が削減できれば、製品単価の安いチップを選定することができる。

［軽量暗号に見込める将来に向けた期待］

- 2020 年、センサー1 兆個、IoT 機器 500 億個がつながる時代に、ローエンドマイコンを搭載する機器に暗号技術が必要になることが予想される。
- 自動運転の実用化、工場やプラントがクラウドとシームレスにつながる時代に、現時点で暗号技術が利用されていない領域にも利用が広がることが予想される。
- 現時点で暗号技術を搭載していない、想定すらしていない機器やシステムにおいて、将来的に実装面での制約を緩和する効果を期待できる。

4. 3. 軽量暗号で達成可能な安全性

- 電子政府推奨暗号リストおよび推奨候補暗号リストに掲載されている暗号技術は、安全性、実装性能が確認された方式であり、カテゴリ毎に想定されている利用の範囲で安全性の問題が生じない、実装性能では実装環境ごとの差が少ないバランスのとれた方式である。
- 軽量暗号は特定の性能指標で優位性をもつように設計されており、提案されている軽量暗号は上記暗号技術よりも安全性が低くなる、もしくは条件付きの安全性になる傾向にある。
- 例えば、64 ビットブロック暗号では同じ鍵で 2^{32} ブロック(32GB)以上のデータを処理すると、高い確率で無作為に選んだビット列と区別できることが知られている。近年は平文ビット列を導出できることも明らかになってきた。
- 上記への対策として、①一つの鍵で処理するデータ量を減らす、②CENC のようなモードの利用、③Abdalla-Bellare の方法を活用するなどリスクを回避する利用方法もある。

- 提案されている軽量ブロック暗号の中には、関連鍵攻撃に対する耐性が保証されていない方式もあるが、関連鍵攻撃が起きないような鍵管理がされていれば許容できる。
- ブロック長に関する安全性指標については研究が進んでいるが、それ以外の指標については明らかになっていないことがほとんどである。
- 電子政府推奨暗号や推奨候補暗号でもリスクなしでの運用は困難であり、軽量暗号でも、利用に応じたリスクを考慮しながらの運用が必要である。

4.4. 今後の活動方針に対する提言

軽量暗号WGでは、2015 年度以降の CRYPTREC での活動方針として、以下の案 (A) (B) (C) を検討してきた。それぞれの概要と目的、期待される効果を示す。

(A) 暗号技術ガイドライン（軽量暗号の最新動向）	(B) 暗号技術ガイドライン（軽量暗号の詳細評価）	(C) 軽量暗号に関する技術公募の実施
− 軽量暗号の最新技術動向をまとめた技術レポート − 軽量暗号の利用促進	− 軽量暗号の安全性と実装性能を統一的に評価した技術レポート − 軽量暗号を選択・利用する際の技術的判断材料として活用 − 軽量暗号の利用促進 − 第三者評価レポートとして活用	− CRYPTREC 暗号リストへの掲載を視野に、軽量暗号の公募・詳細評価・選定 − 電子政府システム等での最適な方式の選択と調達

軽量暗号 WG での議論の結果、今後の活動方針に対して以下のように提言する。

- 軽量暗号は、特定の性能指標における優位性が認められ、次世代のネットワークサービスでの活用が期待される一方、電子政府推奨暗号リスト掲載の暗号技術ほど高い安全性を保証していない方式もあり、利用において留意すべき点がある。
- 軽量暗号を選択・利用する際の技術的判断に資することや今後の利用促進をはかることを目的として暗号技術ガイドラインを発行することが有効と考えられる。
- 暗号技術ガイドラインの発行（(A)または(B)）について、軽量暗号全体とな

ると膨大であり、また技術分類によって状況が異なる。詳細評価が望ましい分野と、現時点では既存文献のサーベイでよいと思われる分野がある。よって、(A)と(B)のハイブリッド案で軽量暗号に関する暗号技術ガイドラインを作成するのがよいと思われる。

- 詳細評価を行う技術分類は、新規評価の必要性（既存文献で十分な評価結果が得られるかどうか)、当該技術分野における我が国の技術の将来性、当該技術分野の現時点での注目度・重要度、評価結果から期待される学術的貢献等を鑑みて決定するのがよいと考えられる。
- 軽量暗号は、現時点では直ちに電子政府システムで活用される段階ではないと考えるが、今後関連する次世代ネットワークサービスに搭載される可能性があることから、上記の活動は長期的には電子政府システムの安全性向上にも資すると期待される。

4. 今後の活動方針

軽量暗号 WG から提示された今後の活動方針に対する提言に沿い、A) 軽量暗号の最新動向 および、B) 軽量暗号の詳細評価 のハイブリッドの形で暗号技術ガイドラインの作成を行う。

2015 年度の活動内容(案)

- (A) 軽量暗号の最新動向
  - 必要に応じ、本年度作成した「暗号技術調査 WG(軽量暗号)報告書」の内容を更新する。
- (B) 軽量暗号の詳細評価
  - (a) 詳細評価を行う対象技術分類の選定
    軽量認証暗号、軽量 MAC、軽量ブロック暗号を検討の対象とする。
  - (b) 詳細評価対象技術分類に関する評価内容の策定
  - (c) 具体的な詳細評価の実施

詳細評価 の対象技術分類の有力候補である軽量認証暗号について、関連する CAESAR プロジェクトの動向を鑑み、次年度は、第一回暗号技術評価委員会の開催に先んじて第一回軽量暗号 WG を開催する可能性もある。

以上

資料1
別添5

暗号技術調査 WG (軽量暗号) 報告書（案）

CRYPTREC 軽量暗号 WG

2015 年 3 月 24 日版

目次

はじめに

本報告書は、暗号技術調査 WG(軽量暗号) が 2013 年度および 2014 年度に調査・検討した内容をまとめたものである。

1 章で総括として、CRYPTREC で扱うスコープ、既存暗号に対して優位性をもつエリア、軽量暗号で達成可能な安全性、今後の CRYPTREC での活動方針について提言をまとめている。

2 章で、軽量暗号に関する現状調査として、軽量暗号技術において、産業上のニーズがあり、具体的な暗号アルゴリズムの設計、安全性評価、実装評価が学会等で発表されている技術分類について代表的な軽量暗号アルゴリズムの現状調査（サーベイ）を行った結果をまとめている。

3 章では、軽量暗号に関わる新しい技術動向や関連する外部動向についての調査、軽量暗号の活用事例および標準化動向についてまとめた。

4 章では、軽量暗号のアプリケーションとして、自動車セキュリティおよび制御システムへの応用についてヒアリングを行った内容をまとめている。

5 章では、特に軽量ブロック暗号について、実装詳細評価を行った結果をまとめた。

以上の調査は、2014 年 12 月までに入手できる情報を対象とした。但し、2015 年 1 月に開催された 2015 年暗号と情報セキュリティシンポジウム (SCIS2015) で発表された内容を一部含む。

本報告書は下記に示す軽量暗号 WG 委員で執筆を行った。

主査	本間　尚文	国立大学法人東北大学　大学院　情報科学研究科　情報基礎科学専攻　准教授
委員	青木　和麻呂	日本電信電話株式会社　NTT セキュアプラットフォーム研究所　主任研究員
委員	岩田　哲	国立大学法人名古屋大学　大学院工学研究科　計算理工学専攻　准教授
委員	小川　一人	NHK 放送技術研究所　ハイブリッド放送システム研究部　上級研究員
委員	崎山　一男	国立大学法人電気通信大学　大学院　情報理工学研究科　教授
委員	渋谷　香士	ソニー株式会社　システム研究開発本部　アプリケーション・プラットフォーム設計部門　セキュリティ技術推進部
委員	鈴木　大輔	三菱電機株式会社　情報技術総合研究所　情報セキュリティ技術部　開発第 1 グループ　主席研究員
委員	成吉　雄一郎	ルネサスエレクトロニクス株式会社　CPU システムソリューション部　主任技師
委員	峯松　一彦	日本電気株式会社　クラウドシステム研究所　主任研究員
委員	三宅　秀享	株式会社東芝　研究開発センター　コンピュータアーキテクチャ・セキュリティラボラトリー　研究主務
委員	渡辺　大	株式会社日立製作所　横浜研究所　エンタープライズシステム研究部　主任研究員

第 1 章

総括：軽量暗号の現状と今後の活動方針

1.1 CRYPTREC で扱う軽量暗号のスコープ

近年、リソースの限られたデバイスにも実装可能な「軽量暗号」(Lightweight Cryptography) の研究開発が進んでいる。これまで多くのアルゴリズムが発表され、国際標準化 (ISO/IEC 29192 など) も進んでいる。欧州では 2004 年から European Commission の第 6-7 次 Framework Programme の研究プロジェクト ECRYPT I, ECRYPT II のテーマとしても取り上げられてきた。日本も小型ハードウェア実装に適した暗号技術等で強みをもっている分野である。

低コスト・低消費電力で動作可能な軽量暗号技術は、今後もセンサー、車載機器、医療機器をはじめさまざまな用途での利用が期待されており、M2M (Machine to Machine), IoT (Internet of Things), CPS (Cyber Physical System) といった次世代のネットワークサービスを構築する上で有効なセキュリティ技術の一つとなることが期待される。

CRYPTREC では、主として電子政府で利用する暗号技術について検討を行っているが、電子政府のみに閉じることなく、さまざまな領域で利用される暗号技術についても技術調査を行い、社会に役立つ形で情報提供を行うことを目指している。軽量暗号技術が求められる製品やサービスにおいて、利用者が最適な暗号方式を選択でき、容易に調達できることを目指し、2013 年度より CRYPTREC 暗号技術評価委員会の下に軽量暗号 WG が設置された。

軽量暗号としてこれまで提案されてきた暗号技術には、ハードウェア実装のサイズ、消費電力量、組み込みソフトウェア実装で必要なメモリサイズ等さまざまな性能指標で最適化されたものがあり、「軽量暗号」に対して一般的に合意されている定義はない。また、性能と安全性のトレードオフもあり、実際には色々な扱いが可能な幅がある。本 WG では、以上の状況を鑑み、「実装性能と安全性のトレードオフを勘案した上で、従来の暗号技術に対して特定の性能指標で優位性（軽量性）を持つように設計された暗号技術」をスコープとし、用途が想定される代表的な性能指標に対して優位性を主張する暗号を主な対象とする。

また、現時点で、公開鍵暗号系において「軽量暗号」として広くコンセンサスがとれている方式はほとんどないため、本報告書では共通鍵暗号系の軽量暗号を対象としている。

軽量暗号が求められるアプリケーションでの要求条件のうち、代表的な性能指標として、本報告書では下記に注目する。

- ハードウェア実装における
  - 回路規模
  - 消費電力量
  - レイテンシ（リアルタイム性能）
- 組み込みソフトウェア実装における

– メモリサイズ（ROM/RAM）

ハードウェア実装の回路規模は、半導体のコストに直結し、また、消費電力 (Power) の指標にもなり得ることが知られている。回路規模の小型化は、RFID をはじめとする回路実装面積の要求条件が厳しいアプリケーションで重要な要件である。また、バッテリや外部供給電源がなく、電磁誘導等で駆動するデバイスにおいても重要な要件である。

消費電力量（Energy）の低減は、人体へ埋め込まれたり密着装備される医療機器をはじめ、バッテリで駆動するあらゆるデバイスで求められる要件である。

レイテンシ（遅延時間）は 1 回の暗号化（復号）処理に必要な時間である。低遅延性はメモリ暗号化や車載機器などのリアルタイム性が求められるアプリケーションで必須の要件である。

組み込みソフトウェア実装では、組み込みマイコン上で実現されるさまざまなアプリケーションの一部として、暗号機能を実装することが多い。組み込みマイコンでは、ROM や RAM のサイズが限られており、小さく実装できる暗号ほど、選択できるマイコンの品種が増える、コストを下げられる等の利点がある。組み込みマイコンは家電機器やセンサー、車載向け等で広く利用されており、実装に必要なメモリサイズ (ROM/RAM) が少ないことはこれらのアプリケーションで重要な要件である。

性能指標	アプリケーションの例
回路規模 (消費電力, コスト)	RFID、低コストセンサー
消費電力量	医療機器、バッテリ駆動デバイス
レイテンシ（リアルタイム性能）	メモリ暗号化、車載機器、産業向け I/O デバイス制御
メモリサイズ (ROM/RAM)	家電機器、センサー、車載機器

1.2 既存暗号に対して優位性をもつ分野

ここで性能指標の視点から軽量暗号が既存暗号に対して優位性を持ちうる分野について述べる。

LSI への実装を想定した回路規模の視点では、現在提案されている軽量暗号と AES の差は数 kgate 程度である。2014 年現在、モバイル向けの SoC (System on a Chip) や GPU で主流となっている 40nm 以下のプロセスで設計される LSI においては、典型的なダイサイズは 50mm^2 から 150mm^2 であり、1mm^2 あたり 1.6Mgate 程度も搭載できるため [1]、数 kgate 程度の回路規模が暗号の優劣の指標とはなりえない。これは、回路全体の 0.1% 未満に関するゲート数削減の議論となるためである。一方で、文献 [2] のミューチップのような、ダイサイズが 50μm 角 (250μm^2) クラスのチップでは数 kgate の差がクリティカルな課題となり、暗号機能の搭載可否に影響を与えうる。現時点では、このサイズのチップはアセンブリの難易度が高いため、500μm 角程度のチップが RFID では主流となっているが、このケースにおいても利用するプロセスが 180nm など古いプロセスであれば、数 kgate の回路規模の差が実装可否に影響を与える可能性がある。

また一般に、回路規模が小さいほど、消費電力あるいは消費電力量は減る傾向にある。環境発電に代表される低消費電力が求められるアプリケーションにおいては、様々な観点で低消費電力化を図る設計が必要となる。軽量暗号を利用することで消費電力あるいは消費電力量に関する設計条件を緩和する効果が期待できる。

次にレイテンシ（リアルタイム性能）の視点では、AES に対して 2 倍の応答速度をおよそ 1/10 の回路規模で実現できる軽量暗号が存在する [3]。この文献の例では、20kgate 程度の回路を用いれば 10ns 以下で暗号演算が可能とされている。一方、AES で同様のリアルタイム性能を得るためには、200kgate 使ったとしても 15ns 必要である。現時点で、産業向け I/O デバイス制御に代表されるような μs オーダーのリアルタイム性能が求められる通信路において暗号技術は利用されていないが、このようなアルゴリズムを利用することで、チップへのコストインパクトなしに暗号技術を利

用できる可能性がある。

最後にソフトウェア実装における軽量暗号の性能指標について述べる。プログラムサイズの観点での AES に対する軽量暗号の優位性として、AES に対しておよそ 1/4 の ROM サイズで実装可能な軽量暗号が存在する [4]。この文献の例では、ルネサスエレクトロニクス社製の組み込みマイコン RL78 を用いた性能評価が行われている。RL78 は産業分野や自動車など幅広く利用されているマイコンの一つであるが、文献 [4] ではそのプラットフォーム上で 220 Bytes の ROM サイズで暗号演算が可能であることが示されている。長期間にわたって利用されてきたレガシー製品に対して、暗号機能を新たに搭載するといったアップデートを施す場合、残された ROM 領域に暗号を実装する必要があり、軽量暗号でなければ搭載できないケースが起こりうる。また、新規に暗号機能を搭載する製品を開発する場合であっても、暗号が使用する ROM 領域の削減が実現できれば、製品単価の安いチップを選定することができる。たとえば RL78 では、ROM サイズを 1KB から 512KB までの間から選ぶことができる。

2020 年にはセンサー 1 兆個、IoT 機器 500 億台の時代が到来すると言われており、前述のようなローエンドのマイコンが利用されている機器においても暗号技術が必要になることが予想される。また、自動運転が実用化され、工場やプラントがクラウドとシームレスにつながる時代が来ると予想されている。このような時代においては、現時点で暗号技術が利用されていない領域であっても、今後活用の必要性が高まると考えられる。軽量暗号は、現時点で暗号技術を搭載していない、あるいは実装上の制約から想定すらしていない機器やシステムにおいて、将来的に実装面での制約を緩和する効果を期待できる。

1.3 軽量暗号で達成可能な安全性

世の中に提案されている様々な暗号技術は様々な性能指標により評価できる。CRYPTREC では様々な暗号技術を評価し、CRYPTREC 暗号リストを維持している。CRYPTREC 暗号リストのうち、電子政府推奨暗号リスト及び推奨候補暗号リストは CRYPTREC により安全性及び実装性能が確認された方式である。これはカテゴリ毎で想定されている範囲でどのような利用がなされたとしても、安全性の問題が生じないとされており、速度などの実装性能についても実装環境毎の差が少ないバランスのよいものを意味している。もちろん、リスト中に注釈がついているものはその注釈の限定の範囲での話である。以下、この節ではそのような方式は議論対象外とする。

一方、軽量暗号は前節で述べられたように従来の暗号技術に対して特定の性能指標で優位性を持つように設計されている。それぞれの性能指標の間には一般にトレードオフが存在することから、提案されている軽量暗号の中には安全性が電子政府推奨暗号や推奨候補暗号より低くなっている方式も存在する。例えば関連鍵攻撃について安全かどうかは保証せず、その分、速度を稼いでいると主張している方式もある。とはいえ、利用場面によってはこのような高い安全性は不要であり、電子政府推奨暗号や推奨候補暗号では高い安全性が消費電力など別の性能指標の足を引っ張っている場合もあることから、安全性の一部に目をつぶった軽量暗号の方が有利な場合もある。よって軽量暗号は利用法によっては有効な技術であるが、設計者が主張するもしくは第三者による安全性評価結果については十分に注意する必要がある。

軽量暗号と謳っている方式の多くはハードウェア実装の回路規模が小さいものが多い。ブロック暗号を実装するためには、ブロック長のビット数に応じた中間状態を保持することが必須であるため、軽量ブロック暗号はブロック長として 128 ビットより小さなものが選ばれるものが多い。例えば 64 ビットブロック長の暗号を CTR モードで利用した場合については、鍵を変更せずに 2^{32} ブロックすなわち 32GB 以上のデータを処理すると高い確率で無作為に選んだビット列と区別できることが知られている。さらに最近の研究 [5] によると具体的にビット列を導出できることも明らかになってきた。逆に、64 ビットブロック暗号を CTR モードで利用したとしても、ひとつの鍵で処理するデータ量が十分に小さければ、無作為に選んだビット列と区別できる確率が十分に小さいため、そのリスクを許容できる場合は効率的

な利用法となり得るだろう。また、標準的ではないが CTR モードの代わりに CENC モード [6] や Abdalla-Bellare の方法 [7] を利用することによりリスクを減らしたり回避したりできることもある。さらに、利用プロトコルもしくはシステム中で関連鍵攻撃が起きないような鍵管理がされている場合は、関連鍵攻撃耐性のない方式を使うことにより、効率をあげることが出来得る。

ブロック長に関する安全性指標については、ここにあげた通り、限界がかなりのところまで知られている。しかし、その他の安全性に関する性能指標については残念ながら分かっていないことが多い。例えば選択平文攻撃は出来ないが既知平文攻撃は想定の必要があるといった場合には明らかとなっていないことが殆んどである。暗号技術の安全性について「どんな攻撃に対しても何も起きない」といった「最強」の安全性についての評価の研究は進んでいるが、一部の軽量暗号で達成しようとしているような条件付きの安全性については研究結果が少なく、あまり明らかになっていないというのが実情である。電子政府推奨暗号や推奨候補暗号を利用したとしてもリスクなしでの運用は困難であり、軽量暗号の利用でも、利用に応じたリスクを考慮しながらの運用が必要である。また、軽量暗号といっても、全てにおいて電子政府推奨暗号や推奨候補暗号より劣っているわけではない。64 ビットブロック長なら、それに応じた安全性、関連鍵攻撃耐性を考慮しないなら、それに応じた安全性が達成されているので、必要な安全性とリスクを考慮した軽量暗号の利用が求められる。

1.4 今後の活動方針に対する提言

軽量暗号 WG では、2015 年度以降の軽量暗号に関する CRYPTREC での活動方針として、以下のような案 (A)(B)(C) を検討してきた（図 1.1 参照)。

それぞれの活動の目的と意義をまとめると下記のようになる。

(A)「暗号技術ガイドライン（軽量暗号の最新動向)」の発行
　軽量暗号の最新技術動向をまとめた技術レポートであり、軽量暗号に関する情報や専門的知見を得るのに活用されることを目的とする。

(B)「暗号技術ガイドライン（軽量暗号の詳細評価)」の発行
　代表的な軽量暗号アルゴリズムの安全性及び実装性能を統一的に評価した技術レポートであり、ユーザが軽量暗号アルゴリズムを選択・利用する際の技術的判断材料として活用できることを目的とする。これにより、軽量暗号の利用が促進されたり、軽量暗号に関する第三者評価レポートとして国際標準化等への寄書として活用されることが期待できる。

(C) 軽量暗号に関する技術公募の実施
　CRYPTREC 暗号リストへの掲載を視野に、軽量暗号の公募・詳細評価を行い、選定を行う。これにより、軽量暗号が CRYPTREC 暗号リストへ新技術として追加され、電子政府システム等で最適な方式を選択でき、容易に調達できるようになることが期待される。

■今後の活動方針　軽量暗号は、特定の性能指標において既存技術と比べて優位性を持ち、M2M, IoT, CPS といった次世代のネットワークサービスを構築する上で有効なセキュリティ技術と期待される一方、電子政府推奨暗号リスト掲載の暗号技術ほど高い安全性を保証していない方式も存在しており、利用において留意すべき点がある。よって、軽量暗号を選択・利用する際の技術的判断の一助となり、今後の利用促進をはかることを目的として暗号技術ガイドラインを発行するのが有益と考えられる。

軽量暗号に関連する技術分野は多岐にわたり、分野ごとに研究開発の状況が異なる。ガイドライン作成にあたって

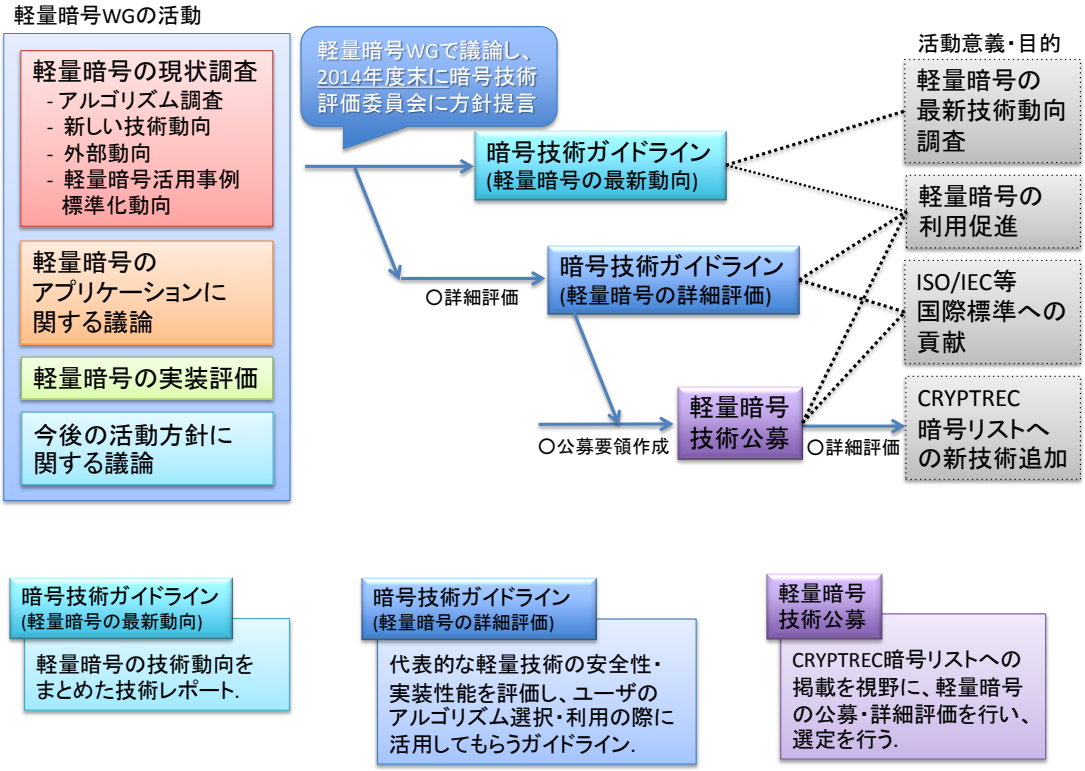


図 1.1　今後の活動方針案

は、詳細評価が望ましい分野や現時点では既存文献のサーベイで十分な分野など、各分野の状況を精査した上で、(A) と (B) のハイブリッド案でまとめるのが妥当と考えられる。詳細評価を行う技術分類は、新規評価の必要性（既存文献で十分な評価結果が得られるかどうか)、当該技術分野における我が国の技術の将来性、当該技術分野の現時点での注目度・重要度、評価結果から期待される学術的貢献等を鑑みて決定することが望ましい。

軽量暗号は、現時点では直ちに (C) の技術公募を行う段階ではないと考えるが、今後、IoT などの次世代ネットワークサービスで活用される可能性があることから、本 WG での検討が、長期的には電子政府システムの安全性向上にも資することが期待される。

参考文献

[1] STMicroelectronics, “CMP annual users meeting,” http://cmp.imag.fr/aboutus/slides/Slides2013/05_ST_2013.pdf, 2013.

[2] M. Usami, H. Tanabe, A. Sato, I. Sakama, Y. Maki, T. Iwamatsu, T. Ipposhi and Y. Inoue, “A 0.05 × 0.05 mm^2 RFID Chip with Easily Scaled-Down ID-Memory,” ISSCC 2007, Digest of Technical Papers, pp. 482-483, 2007.

[3] M. Kneevi, V. Nikov, and P. Rombouts, “Low-Latency Encryption – Is “Lightweight= Light+ Wait”?” CHES 2012, pp. 426-446, 2012.

[4] M. Matsui and Y. Murakami, “Minimalism of Software Implementation - Extensive Performance Analysis of Symmetric Primitives on the RL78 Microcontroller,” FSE 2013, pp. 393-409, 2013.

[5] David A. McGrew, ”Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes,” preproceedings of FSE 2013, Session 4-1.

[6] Tetsu Iwata, “New Blockcipher Modes of Operation with Beyond the Birthday Bound Security”, FSE 2006, pp.310-327, 2006.

[7] Michel Abdalla, Mihir Bellare, “Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques,” ASIACRYPT 2000, pp. 546-559, 2000.

第 2 章

軽量暗号に関する現状調査: 軽量暗号アルゴリズム

2.1 軽量暗号に関する現状調査の概要

2013 年度および 2014 年度、軽量暗号 WG では、軽量暗号技術において、産業上のニーズがあり、具体的な暗号アルゴリズムの設計、安全性評価、実装評価が学会等で発表されている技術分類について、現状調査 (サーベイ) を行った。また、軽量暗号技術に関わる新しい技術動向や関連する外部動向についての調査や、軽量暗号の活用事例および標準化動向も行った。

これらの軽量暗号に関する現状調査を 2 章および 3 章にまとめる。

2 章の執筆担当者は下記の通りである.

第 2 章　軽量暗号アルゴリズム	
2.2 章　軽量ブロック暗号	青木委員, 渋谷委員
2.3 章　軽量ストリーム暗号	渡辺委員
2.4 章　軽量ハッシュ関数	三宅委員
2.5 章　軽量メッセージ認証コード	渡辺委員
2.6 章　認証暗号	峯松委員, 鈴木委員

2.2 軽量ブロック暗号

2.2.1 軽量ブロック暗号の安全性

本章では軽量暗号に分類されるブロック暗号の安全性に関する調査報告を行なう。軽量暗号に求められる安全性は暗号研究者の間でさえも合意されていない。もともと「『軽量』暗号」の名前の通り、安全性ではなく実装性能用件から始まった研究対象であり、「軽量とはいえ通常の暗号と同等の安全性が必要」という意見や、「通常利用しないような用途の安全性を犠牲にして軽量化を行なう」、また「通常利用しない用途の安全性は当然考慮せず、さらに通常使う用途に対しても長期間の安全性を保証せず、ぎりぎりを狙う」といった方式まである。従って、軽量暗号の利用に際しては、設計指針としてどこまでの安全性を考慮しているのかを理解して利用することが重要である。つまり従来型のブロック暗号の安全性と異なる部分が利用にあたって重要であることから本章では通常目的のブロック暗号に求められる安全性を調査する。

そもそも「何が『軽量ブロック暗号』か」という問に対して、軽量暗号という名前自体 buzz word と化しており難しい。広い意味で「軽量ブロック暗号」とされるものは [1] に詳しくあげられているが、AES など従来型のブロック暗号も含まれている。AES は電子政府推奨暗号であり、さらに事実上の世界標準であることから、本章では原則 AES より「軽量」なものを「軽量暗号」とした。軽量暗号の標準としては既に ISO/IEC で定められていることから ISO/IEC 29192 から中心に調査対象方式を選び、その他、共通鍵暗号の研究者の多くが「軽量」として引用している方式を調査対象とした[*1]。ここで、TDES, Camellia, CLEFIA については、平成 25 年に公表された CRYPTREC 暗号リストに掲載されており、安全性が十分に確認されている方式である。また、その後、本報告作成までの 2 年間の間に大きな問題は報告されていないので本章では調査対象外とする。

なお本章では、純粋にアルゴリズムそのものについての攻撃に対する安全性のみの調査を行ない、サイドチャネル攻撃や故障利用攻撃などは含めないこととする。また、秘密鍵の全数探索を高速化する手法、特に biclique を利用した中間一致攻撃的な手法 [4] がいくつかの暗号に対して提案されているが、効果は限定的であり、暗号の脆弱性として認められるのかどうかについても暗号研究者間で合意が得られていないのでこれも取り扱わないこととする。

表 2.1　軽量ブロック暗号の安全性評価

名称	提案文献	ブロック長	鍵長	仕様段数	攻撃可能段数	備考
LBlock	[21]	64	80	32	23	[6]
LED	[11]	64	64 ∼ 128	8/12	3/8	LED-64 と LED-128 に対応 [10]
Piccolo	[6]	64	80/128	25/31	9/11	whitening 鍵あり [16]
PRINCE	[2]	64	128	12	8	[8]
PRESENT	[5]	64	80/128	31	25(26)	26 段攻撃は全平文 [9]
PRINTcipher	[12]	48/96	80/160	48/96	48/96	[13] は弱鍵攻撃、[14] は関連鍵攻撃
TWINE	[18, 19]	64	80/128	36	—/25?	攻撃 [7] とそれに対する疑問 [20]

[*1] 近年提案された Simon と Speck[3] については軽量暗号と見做されることが多い。提案論文そのものでは安全性評価が行なわれていないことから、解析論文が次々と出ている状態である。さらに、これらの方式は、パラメータが非常に多く、解析結果もそれぞれのパラメータに対して多数存在し、ここで最新情報を載せてもすぐに更新される可能性が高いことから今回は掲載を見送った。なお、現在 (2014 年 12 月) のところ、推奨パラメータでは破れていない。

参考文献

[1] Alex Biryukov and and Léo Perrin. State of the Art in Lightweight Cryptography. `http://cryptolux.org/index.php/Lightweight_Cryptography`, 2014.

[2] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[3] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. IACR Cryptology ePrint Archive 2013/404.

[4] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology — ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer-Verlag, Berlin, Heidelberg, 2011.

[5] Andrey Bogdanov, Lars Ramkilde Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and Charlotte H. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer-Verlag, Berlin, Heidelberg, New York, 2007.

[6] Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and SIMON. In Palash Sarkar and Tetsu Iwata editors, *Advances in Cryptology — ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 179–199. Springer-Verlag, Berlin, Heidelberg, 2014.

[7] Özkan Boztas, Ferhat Karakoç, and Mustafa Çoban. Multidimensional Meet-in-the-Middle Attacks on Reduced-Round TWINE-128. LightSEC 2013.

[8] Anne Canteaut, María Naya-Plasencia, and Bastien Vayssière. Sieve-in-the-middle: Improved MITM attacks. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology — CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 222–240, Berlin, Heidelberg, 2013. Springer-Verlag.

[9] Joo Yeon Cho. Linear cryptanalysis of reduced-round PRESENT. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2008: The Cryptographers' Track at the RSA Conference 2010*, volume 5985 of *Lecture Notes in Computer Science*, pages 302–317, Berlin, Heidelberg, New York, 2010. Springer-Verlag.

[10] Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key recovery attacks on 3-round Even-Mansour,

8-step LED-128, and full AES^2. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology — ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 337–356. Springer-Verlag, Berlin, Heidelberg, 2013.

[11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer-Verlag, Berlin, Heidelberg, New York, 2011.

[12] Lars Ramkilde Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. PRINTcipher: A block cipher for ic-printing. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems — CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer-Verlag, Berlin, Heidelberg, New York, 2010.

[13] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A cryptanalysis of PRINTcipher: The invariant subspace attack. In Phillip Rogaway, editor, *Advances in Cryptology — CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221, Berlin, Heidelberg, 2011. Springer-Verlag.

[14] Yuseop Lee, Kitae Jeong, Changhoon Lee, Jaechul Sung, and Seokhie Hong. Related-key cryptanalysis on the full PRINTcipher suitable for IC-printing. *International Journal of Distributed Sensor Networks*, 2014(Article ID 389476), 2014.

[15] David A. McGrew. Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. In Shiho Moriai, editor, *preproceedings of Fast Software Encryption Workshop 2013 (FSE 2013)*, Singapore, 2013.

[16] 芝山直喜, 金子敏信. Piccolo の新しい高階差分特性. 信学技報 ISEC2014-34, 2014.

[17] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 342–357. Springer-Verlag, Berlin, Heidelberg, New York, 2011.

[18] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight, versatile block cipher. In Gregor Leander and François-Xavier Standaert, editors, *ECRYPT Workshop on Lightweight Cryptography*, pages 146–169. ECRYPT II, 2011.

[19] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE : A lightweight block cipher for multiple platforms. In Lars Ramkilde Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Workshop, SAC 2012, Windsor, Ontario, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354, Berlin, Heidelberg, 2013. Springer-Verlag.

[20] Long Wen, Meiqin Wang, Andrey Bogdanov, and Huaifeng Chen. Note of Multidimensional MITM Attack on 25-Round TWINE-128. IACR Cryptology ePrint Archive 2014/425.

[21] Wenling Wu and Lei Zhang. LBlock: A Lightweight Block Cipher. In Javier Lopez and Gene Tsudik, *Applied Cryptography and Network Security — 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, volume 2011 of *Lecture Notes in Computer Science*, pages 327–344. Berlin, Heidelberg, 2012. Springer-Verlag.

2.2.2 軽量ブロック暗号の実装性能

本章では、軽量暗号技術の現状調査として、主要な軽量ブロック暗号アルゴリズム、および CRYPTREC 暗号リストの電子政府推奨暗号リストに含まれるブロック暗号アルゴリズムの実装性能 (ハードウェア、ソフトウェア) 調査結果をまとめる。

2.2.2.1 調査対象

調査対象とした軽量ブロック暗号アルゴリズムは、ISO/IEC 29192 軽量暗号のパート 2 ブロック暗号に記載されているブロック暗号 (PRESENT、CLEFIA)、および主要国際学会で発表されており、現段階で有力な攻撃法が発見されておらず、かつ十分な実装性能を持つと考えられるアルゴリズム (LED、Piccolo、TWINE) とした。また、参考として、CRYPTREC 暗号リストの “電子政府推奨暗号リスト” に含まれるブロック暗号 (3-key Triple DES、AES、Camellia) の実装性能も調査した。さらに、類似の実装特性を持つ低レイテンシ暗号 PRINCE の実装性能も調査した。表 2.2 にこれら調査対象アルゴリズムをまとめる。実装性能の調査を行う論文としては、十分信頼が置けるデータが得られることを考慮し、各対象アルゴリズムの提案論文、および主要国際会議で発表された論文を中心に調査を行った。調査結果については、様々な資料から得られた評価値をできる限り公平になるように並べた。しかしながら、全ての評価が同じ環境で行われているわけではなく、評価環境や実装者によって評価値が変化する可能性があるため、本調査の数値は参考程度である点に注意されたい。

表 2.2　調査対象ブロック暗号アルゴリズム基本情報

Algorithm	Block size [bit]	Key size [bit]	# rounds	Structure	Ref.
3-Key Triple DES	64	168	48	Feistel	電子政府推奨暗号
AES	128	128/192/256	10/12/14	SPN	電子政府推奨暗号
Camellia	128	128/192/256	18/24/24	Feistel	電子政府推奨暗号
PRESENT	64	80/128	31	SPN	ISO/IEC29192-2
CLEFIA	128	128/192/256	18/22/26	GFN	ISO/IEC29192-2
LED	64	64/65 128	32/48	SPN	[11]
Piccolo	64	80/128	25/31	GFN	[28]
TWINE	64	80/128	36	GFN	[30]
PRINCE	64	128	12	SPN	[6]

2.2.2.2 ハードウェア実装性能調査

ハードウェア実装性能調査としては、十分な評価が行われていると考えられる ASIC での実装性能評価を調査した。実装性能の評価指標は、大別すると、主に自身で電源を持たないような機器 (passive device) 向けの指標として消費電力 (Power)、自身で電源を持つような機器 (active device) 向けの指標として消費電力量 (Energy) の 2 つがある。このうち、消費電力 (Power) における効率を示す指標としてはゲート規模がよく知られている。一方、消費電力量 (Energy)

の効率を示す指標としては、((ゲート規模)×(1-block 処理に必要なサイクル数)/(ブロックサイズ)) によって計算される energy per bit や、(((1-cycle で処理するビット数)×10^9)/(ゲート規模)^2) によって計算される FOM(Figure of Merit) が知られている。これらの調査結果を表 2.3-2.5 にまとめる。表中、Mode は暗号化関数のみを実装している場合は Enc と記載し、暗号化関数、復号関数をともに実装している場合は Enc/Dec と記載している。また、Area の評価として用いている GE は gate equivalent の略であり、ゲート規模を表す。Cycles/block は 1-block の演算に必要なサイクル数を表し、Throughput は、100[kHz] での Throughput のみを調査している。また、表中 LED* は LED の推定値による評価結果を示している。

表 2.3　128 bit ブロック暗号のハードウェア実装性能

Algorithm	Key size [bit]	Mode	Area [GE]	Cycles/ block	Throughput @100kHz [kbps]	Tech. [μm]	Ref.
AES	128	Enc/Dec	3,400	1,032/1,165	12.4/11.0	0.35	[10]
		Enc	2,400	226	56.6	0.18	[20]
		Enc/Dec	12,454	11	1,163.6	0.13	[27]
		Enc/Dec	5,398	54	237.0	0.13	
		Enc	3,100	160	80.0	0.13	[12]
Camellia	128	Enc/Dec	6,511	44	290.9	0.13	[27]
		Enc/Dec	6,264	44	290.9	0.18	
CLEFIA	128	Enc	2,488	328	39.0	0.13	[2]
		Enc/Dec	2,604	328/320	39.0/40.0	0.13	
		Enc	2,678	176	72.7	0.13	
		Enc/Dec	4,950	36	355.6	0.09	[29]
		Enc/Dec	5,979	18	711.1	0.09	

2.2.2.3　ソフトウェア実装性能調査

ソフトウェア実装性能調査として、ハイエンド CPU、およびローエンド CPU による実装評価の調査を行った。結果を表 2.6-2.8 にまとめる。ハイエンド CPU では実行速度として、Cycles/byte (1-byte の演算に必要なサイクル数) を調査し、ローエンド CPU では、Cycles/byte、および ROM、RAM 使用量をそれぞれ調査した。表 2.6 における Type は実装手法を表しており、それぞれ、Table による表引きを主に使用した実装を Table、VPI(Vector Permute Instruction) を利用した実装を VPI、bitslice 実装を Bitslice と記述している。Bitslice 実装における block 数の記述は並列実行ブロック数を表しており、例えば 8-block と記述があるものは、8-block 並列実行の bitslice 実装を表している。また、TWINE の実装手法における Single は通常の 1-block を実行する実装手法、Double は 2-block 並列に実行する実装手法を表す。

2.2.2.4　まとめ

本章では、軽量暗号技術の現状調査として、128-bit ブロック暗号アルゴリズム AES、Camellia、CLEFIA、および 64-bit ブロック暗号アルゴリズム 3-Key Triple DES、LED、Piccolo、TWINE、PRINCE の軽量暗号用途でのハー

ドウェア、ソフトウェア実装性能を公知の論文から調査した結果をまとめた。

表 2.4　64-bit ブロック暗号のハードウェア実装性能 (flexible-key setting)

Algorithm	Key size [bit]	Mode	Area [GE]	Cycles/ block	Throughput @100kHz [kbps]	Tech. [μm]	Ref.
Triple-DES	168	Enc/Dec	5,504	48	133.3	0.13	[27]
PRESENT	80	Enc	1,000	563	11.4	0.35	[26]
			1,570	32	200.0	0.18	[5]
	128	Enc	2,587	63	101.6	0.35	[26]
			2,681	39	164.1	0.35	[26]
			1,391	559	11.4	0.18	[24]
			1,886	32	200.0	0.18	[5]
LED	64	Enc	966	1,248	5.1	0.18	[11]
	128	Enc	1,265	1,872	3.4		
LED* (推定値)	64	Enc	2,695	32	200.0	0.18	[1]
	80	Enc	1,040	1,872	3.4	0.18	[11]
		Enc	2,780	48	133.3	0.18	[1]
	96	Enc	1,116	1,872	3.4	0.18	[11]
		Enc	2,866	48	133.3	0.18	[1]
	128	Enc	3,036	48	133.3	0.18	[1]
Piccolo	80	Enc	1,048	432	14.8	0.13	[14, 28]
		Enc/Dec	1,109	432	14.8		
		Enc	1,499	27	237.0		
		Enc/Dec	1,638	27	237.0		
	128	Enc	1,338	528	12.1		
		Enc/Dec	1,397	528	12.1		
		Enc	1,776	33	193.9		
		Enc/Dec	1,942	33	193.9		
TWINE	80	Enc	1,503	36	177.8	0.09	[30]
		Enc/Dec	1,799	36	177.8		
		Enc	1,011	393	16.3		
	128	Enc	1,866	36	177.8		
		Enc/Dec	2,285	36	177.8		
PRINCE	128	Enc/Dec	3,491	12	533.3	0.13	[6]
			2,953	12	533.3	0.13	[3]
			8,577	1	6,400		

表 2.5　64-bit ブロック暗号のハードウェア実装性能 (fixed-key setting)

Algorithm	Key size [bit]	Mode	Area [GE]	Cycles/ block	Throughput @100kHz [kbps]	Tech. [μm]	Ref.
LED	64	Enc	688	1,280	5.0	0.18	[11]
	128	Enc	700	1,872	3.4		
LED* (推定値)	64	Enc	2,354	32	200.0	0.18	[1]
	80	Enc	690	1,872	3.4	0.18	[11]
			2,354	48	133.3	0.18	[1]
	96	Enc	695	1,872	3.4	0.18	[11]
			2,354	48	133.3	0.18	[1]
	128	Enc	2,354	48	133.3	0.18	[1]
Piccolo	80	Enc	616	432	14.8	0.13	[14, 28]
		Enc/Dec	675	432	14.8		
		Enc	1,051	27	237.0		
		Enc/Dec	1,199	27	237.0		
	128	Enc	654	528	12.1		
		Enc/Dec	721	528	12.1		
		Enc	1,083	33	193.9		
		Enc/Dec	1,249	33	193.9		

表 2.6　128-bit ブロック暗号 (AES、Camellia) のソフトウェア実装性能 (ハイエンド CPU)

Algorithm	Key size [bit]	Type	Cycles/byte	Platform	Ref.
AES	128	VPI (Enc/Dec)	6.66/9.12	Core i5 U560	[30]
			7.42/9.44	Core i7 2600S	
			10.28/12.37	Core i3 2120	
			14.72/17.82	Xeon E5620	
			12.16/14.39	Core2Quad Q9550	
			22.04/25.82	Core2Duo E6850	
		Table (Enc/Dec)	14.26/19.27	Core i5 U560	
			14.04/21.17	Core i7 2600S	
			19.03/28.68	Core i3 2120	
			31.60/42.69	Xeon E5620	
			22.74/30.94	Core2Quad Q9550	
			22.43/30.76	Core2Duo E6850	
		Bitslice (8-block)	9.32	Core2Quad Q6600	[16]
			7.59	Core2Quad Q9550	
			6.92	Core i7 920	
	128	Bitslice (1/2/16-block)	10.7/7.8/5.4	PowerPC G4	[13]
	192		12.8/9.3/6.7		
	256		14.9/10.8/7.9		
Camellia	128	Bitslice (128-block)	9.19	Core2Duo E6400	[19]

表 2.7　64-bit ブロック暗号のソフトウェア実装性能 (ハイエンド CPU)

Algorithm	Key size [bit]	Type	Cycles/byte	Platform	Ref.
PRESENT	80/128	Bitslice (8/16/32-blk)	8.46/6.52/4.73	Xeon E3-1280	[17]
			10.88/7.26/5.79	Core i7 870	
			13.55/10.98/7.55	Xeon E5410	
	80	Table/VPI/Bitslice	72.6/35.0/17.4	Core i3 2367M	[4]
			65.7/42.1/20.7	Xeon X5650	
			59.5/42.3/21.0	Core2Duo P8600	
	128	Table/VPI/Bitslice	72.5/35.0/18.9	Core i3 2367M	
			65.7/42.1/24.1	Xeon X5650	
			59.5/42.4/24.1	Core2Duo P8600	
LED	64	Table/VPI/Bitslice	76.0/36.0/12.2	Core i3 2367M	[4]
			70.9/48.1/13.1	Xeon X5650	
			62.8/47.4/14.2	Core2Duo P8600	
	128	Table/VPI/Bitslice	113.3/54.6/17.6	Core i3 2367M	
			105.9/67.4/19.0	Xeon X5650	
			93.5/68.7/20.2	Core2Duo P8600	
Piccolo	80	Bitslice (16-blk)	4.57	Xeon E3-1280	[17]
			5.69	Core i7 870	
			6.85	Xeon E5410	
	128		5.52	Xeon E3-1280	
			6.80	Core i7 870	
			8.23	Xeon E5410	
	80	Table/VPI/Bitslice	83.9/33.3/9.2	Core i3 2367M	[4]
			71.0/37.4/9.7	Xeon X5650	
			67.1/38.3/10.7	Core2Duo P8600	
	128		103.6/41.6/10.9	Core i3 2367M	
			87.5/47.4/12.5	Xeon X5650	
			83.6/47.2/13.0	Core2Duo P8600	
TWINE	80/128	Single/Double	9.47/4.77	Core i5 U560	[30]
			11.10/5.55	Core i7 2600S	
			15.06/7.55	Core i3 2120	
			13.62/6.87	Xeon E5620	
			15.16/7.93	Core2Quad Q9550	
			26.85/14.85	Core2Duo E6850	

表 2.8　ブロック暗号のソフトウェア実装性能 (ローエンド CPU)

Algorithm	Block size [bit]	Key size [bit]	ROM [byte]	RAM [byte]	Cycles/ byte [Enc/Dec]	Platform	Ref.
AES	128	128	1,912	432	125/181	ATmega163	[7]
			1,659	33	287.5/4381	ATtiny45	[9]
			970	84	7,743/10,862	RL78	[18]
			1,989	64	3,917/5,911		
			2,380	64	3,865/5,706		
Camellia	128	128	1,020	78	39,357/152,023	RL78	[18]
			2,033	64	4,337/4,477		
			2,047	74	4,125/4,244		
CLEFIA	128	128	1,309	76	18,062/18,759	RL78	[18]
			2,026	64	7,768/7,799		
			2,040	86	6,208/6,740		
PRESENT	64	80	2,398	528	1,199/1,228	ATmega163	[24]
			1,000	18	1,412.5/1,700	ATtiny45	[9]
			936	0	1,340.4/1404.3	ATtiny45	[22]
			1,794	18	1090.1/-	ATtiny45	
			426	18	11,340.6/12,728.1	ATtiny45	
			512	62	61,634/60,834	RL78	[18]
			1,009	54	13,883/14,014		
			1,855	48	9,007/8,920		
TWINE	64	80	1,304	414	271/271	ATmega163	[30]
			728	335	2,350/2,337		
			792	191	2,350/2,337		
			2,294	386	163/163		
PRINCE	64	128	2,382	220	225.4	ATtiny85	[23]

参考文献

[1] The LED block cipher, December 2013. Available from `https://sites.google.com/site/ledblockcipher/hardware`.

[2] Toru Akishita and Harunaga Hiwatari. Very compact hardware implementations of the blockcipher CLEFIA. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pages 278–292. Springer, 2011.

[3] Lejla Batina, Amitabh Das, Baris Ege, Elif Bilge Kavun, Nele Mentens, Christof Paar, Ingrid Verbauwhede, and Tolga Yalçin. Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures. In Hutter and Schmidt [15], pages 103–112.

[4] Ryad Benadjila, Jian Guo, Victor Lomné, and Thomas Peyrin. Implementing lightweight block ciphers on x86 architectures. In Tanja Lange, Kristin E. Lauter, and Petr Lisonek, editors, *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*, pages 324–351. Springer, 2013.

[5] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Paillier and Verbauwhede [21], pages 450–466.

[6] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[7] Joppe W. Bos, Dag Arne Osvik, and Deian Stefan. Fast implementations of AES on various platforms. *IACR Cryptology ePrint Archive*, 2009:501, 2009.

[8] Christophe Clavier and Kris Gaj, editors. *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*. Springer, 2009.

[9] Thomas Eisenbarth, Zheng Gong, Tim Güneysu, Stefan Heyse, Sebastiaan Indesteege, Stéphanie Kerckhof, François Koeune, Tomislav Nad, Thomas Plos, Francesco Regazzoni, François-Xavier Standaert, and Loïc van Oldeneel tot Oldenzeel. Compact implementation and performance evaluation of block ciphers in ATtiny devices. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *Progress in Cryptology - AFRICACRYPT*

2012 - 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings, volume 7374 of *Lecture Notes in Computer Science*, pages 172–187. Springer, 2012.

[10] Martin Feldhofer, Johannes Wolkerstorfer, and Vincent Rijmen. AES implementation on a grain of sand. *Information Security, IEE Proceedings*, 152(1):13–20, 2005.

[11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Preneel and Takagi [25], pages 326–341.

[12] Panu Hämäläinen, Timo Alho, Marko Hännikäinen, and Timo D. Hämäläinen. Design and implementation of low-area and low-power AES encryption hardware core. In *Ninth Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD 2006), 30 August - 1 September 2006, Dubrovnik, Croatia*, pages 577–583. IEEE Computer Society, 2006.

[13] Mike Hamburg. Accelerating AES with vector permute instructions. In Clavier and Gaj [8], pages 18–32.

[14] Harunaga Hiwatari, Kyoji Shibutani, Takanori Isobe, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Compact hardware implementations of the ultra-lightweight block cipher Piccolo. Proceedings of the ECRYPT Workshop on Lightweight Cryptography, 2011.

[15] Michael Hutter and Jörn-Marc Schmidt, editors. *Radio Frequency Identification - Security and Privacy Issues 9th International Workshop, RFIDsec 2013, Graz, Austria, July 9-11, 2013, Revised Selected Papers*, volume 8262 of *Lecture Notes in Computer Science*. Springer, 2013.

[16] Emilia Käsper and Peter Schwabe. Faster and timing-attack resistant AES-GCM. In Clavier and Gaj [8], pages 1–17.

[17] Seiichi Matsuda and Shiho Moriai. Lightweight cryptography for the cloud: Exploit the power of bitslice implementation. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*, pages 408–425. Springer, 2012.

[18] Mitsuru Matsui and Yumiko Murakami. Minimalism of software implementation - extensive performance analysis of symmetric primitives on the RL78 microcontroller. In Shiho Moriai, editor, *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume 8424 of *Lecture Notes in Computer Science*, pages 393–409. Springer, 2013.

[19] Mitsuru Matsui and Junko Nakajima. On the power of bitslice implementation on Intel Core2 processor. In Paillier and Verbauwhede [21], pages 121–134.

[20] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the limits: A very compact and a threshold implementation of AES. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 69–88. Springer, 2011.

[21] Pascal Paillier and Ingrid Verbauwhede, editors. *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*. Springer, 2007.

[22] Konstantinos Papagiannopoulos and Aram Verstegen. Speed and size-optimized implementations of the PRESENT cipher for tiny AVR devices. In Hutter and Schmidt [15], pages 161–175.

[23] Kostas Papapagiannopoulos. High throughput in slices: The case of PRESENT, PRINCE and KATAN64

ciphers. In Nitesh Saxena and Ahmad-Reza Sadeghi, editors, *Radio Frequency Identification: Security and Privacy Issues - 10th International Workshop, RFIDSec 2014, Oxford, UK, July 21-23, 2014, Revised Selected Papers*, volume 8651 of *Lecture Notes in Computer Science*, pages 137–155. Springer, 2014.

[24] Axel Poschmann. Lightweight cryptography - cryptographic engineering for a pervasive world. *IACR Cryptology ePrint Archive*, 2009:516, 2009.

[25] Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.

[26] Carsten Rolfes, Axel Poschmann, Gregor Leander, and Christof Paar. Ultra-lightweight implementations for smart devices - security for 1000 gate equivalents. In Gilles Grimaud and François-Xavier Standaert, editors, *Smart Card Research and Advanced Applications, 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, London, UK, September 8-11, 2008. Proceedings*, volume 5189 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 2008.

[27] Akashi Satoh and Sumio Morioka. Hardware-focused performance comparison for the standard block ciphers AES, Camellia, and Triple-DES. In Colin Boyd and Wenbo Mao, editors, *Information Security, 6th International Conference, ISC 2003, Bristol, UK, October 1-3, 2003, Proceedings*, volume 2851 of *Lecture Notes in Computer Science*, pages 252–266. Springer, 2003.

[28] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: an ultra-lightweight blockcipher. In Preneel and Takagi [25], pages 342–357.

[29] Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher CLEFIA (extended abstract). In Alex Biryukov, editor, *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, volume 4593 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2007.

[30] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight block cipher for multiple platforms. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer, 2012.

2.3 軽量ストリーム暗号

本章では、軽量（lightweight）ストリーム暗号について報告する。

2.3.1 ECRYPT Stream Cipher project (eSTREAM)

eSTREAM は、EU における暗号技術研究 ECRYPT の一環として 2004〜2008 年に実施されたプロジェクトである。プロジェクトの中では、ストリーム暗号アルゴリズムの公募、および実装性能と安全性の両面から評価が行われた。eSTREAM プロジェクトでは、AES よりも性能が顕著であることを公募要件として求めており、ソフトウェア実装が高速であること (Profile I)、ハードウェア実装が軽量な暗号 (Profile II) のそれぞれの要件に特化したアルゴリズムを公募した。特にハードウェア実装の軽量性を追求した Profile II は、軽量暗号研究の流行を生んだ。CRYPTREC 軽量暗号 WG においては、low-latency 暗号など、処理速度の観点で軽量 (高速) と謳う暗号についても俎上に乗っているため、本報告では Profile I についても調査を行った。

表 2.9　eSTREAM Portfolio [1]

Profile 1 (ソフトウェア向け)	Profile 2 (ハードウェア向け)
HC-128 (128-bit), HC-256 (256-bit)	Grain-v1 (80-bit)
Rabbit (128-bit)	MICKEY 2.0 (80-bit, 128-bit)
Salsa20/12 (128-bit, 256-bit)	Trivium (80-bit)
SOSEMANUK (128-bit)	

当初の Portfolio では、Grain-128 が含まれていたが、2012 年の報告 [2] では Grain-128 を除外している。[2] によれば、Grain-128 は開発者自身がサポートしなくなったと報告されている。これは [11] で Grain-128 に弱鍵が存在する、およびセキュリティマージンが小さい、の 2 点が報告されたことが原因である。

2.3.1.1 Profile I (**ソフトウェア実装が高速な暗号**)

Profile I は PC、サーバ上で高速なソフトウェア向け暗号を指向しており、鍵長は 128 ビット以上である。eCRYPT II の 1 プロジェクトである VAMPIRE の成果である eBACS [9] で、さまざまな環境での処理速度を確認することができる。表 2.10 は、Intel Core i5（64 ビットモード）における評価結果である (詳細：Intel Core i5-2400S; 4 x 2495MHz; sandy, supercop-20120908)。

また、2009 TI Sitara AM3703 500MHz (ARM Cortex A8) 上での処理性能は表 2.11 のとおりである (詳細：armeabi (v7-A, Cortex A8); 2009 TI Sitara AM3703; 1 x 500MHz; h7silver, supercop-20130126)。

Profile I に属するアルゴリズムは、いずれも AES(AES-NI 不使用の場合) に比べて 3〜5 倍のスループットを実現している。AES 命令が実装されていない環境では利用に適するケースもある。現在、Salsa20 は TLS 用の暗号スイートとして提案が進められている [7]。

アルゴリズム構造は算術演算を用いるもの (Rabbit, Salsa20/12)、大きな内部状態を持ち、初期化に時間をかけるもの (HC-128, SOSEMANUK) の 2 系統に分かれる。後者のアルゴリズムは短いデータの処理には適していない。また、Profile I に属するアルゴリズムは、ハードウェア実装したときに論理規模が大きくなるケースが多いと考えられる。

HC-128, Rabbit は組み込み機器向けの SSL/TLS 実装 ChaSSL に実装されている [2]。また、Rabbit は

表 2.10　eSTREAM Portfolio Profile I アルゴリズムのソフトウェア実装性能（Intel Core i5）[8]

	処理速度 (cycle/B)				
	長いメッセージ	4096B	576B	64B	8B
HC-128	2.32	7.13	36.44	309.25	2472.00
Rabbit	4.41	4.58	5.41	13.06	80.00
Salsa20/12	2.40	2.44	2.70	4.94	56.50
SOSEMANUK	3.54	3.81	5.72	20.56	164.50
AES	11.33	11.41	11.78	15.75	77.50
KCipher-2(*)	4.01	4.22	5.50	17.45	111.51

CRYPTREC 推奨ストリーム暗号との性能比較のため、KCipher-2 の処理性能を [14] に記載されている性能から見積もった。なお、[14] の評価環境は Intel Core2Duo である。

表 2.11　eSTREAM Portfolio Profile I アルゴリズムのソフトウェア実装性能（ARM Cortex A8）[8]

	処理速度 (cycle/B)				
	長いメッセージ	4096B	576B	64B	8B
Salsa20/12	5.52	5.84	8.14	28.50	264.75
AES	19.28	20.36	29.59	111.83	852.38

ISO/IEC 18033-4 [5] および RFC 4503 [7] に記載されている。

2.3.1.2　Profile II (**ハードウェア実装規模／消費電力が小さい暗号**)

Profile II は軽量なハードウェア実装向け暗号を指向しており、鍵長は 80 ビット以上である。軽量暗号の実装では、状態を保持するレジスタが論理回路の大半を占めることから、回路規模削減のために、Profile I に比べて短い鍵長を許容しているものと考えられる。鍵長 128 ビットレベルセキュリティを持つアルゴリズムに比べると、安全性が低く設定されているので、用途は限定されるべきである。

表 2.12 および図 2.1 は、文献 [10] から抜粋した Profile II (および AES) のハードウェア実装性能である。いずれのアルゴリズムも、論理規模、処理速度の両面で AES に比べて顕著な軽量性を実現している。軽量実装では、回路の動作周波数が低く抑えられているケースが多いと想定されるため、[10] では動作周波数を 10MHz, 100KHz に固定した場合の消費電力評価も行われている。結果として、消費電力はアルゴリズムによらず、論理規模に比例して増加する傾向が見られた。

2.3.2　ISO/IEC 29192-3

ISO/IEC JTC 1/SC 27 では、一般的な暗号アルゴリズムの標準を定めた ISO/IEC 18033 に加えて、軽量暗号の標準を ISO/IEC 29192 で定めている。ストリーム暗号は 2012 年に発行された Part 3 に収められており、eSTREAM Portfolio に掲載された Trivium (鍵長 80 ビット) と、CRYPTREC 推奨候補暗号に掲載された Enocoro (鍵長 80 ビットおよび 128 ビット) の 2 つのアルゴリズムが収録されている。Enocoro-80, -128v2 のハードウェア実装性能を表 2.13 にまとめる。Trivium の実装性能については紹介済みなので割愛する。Enocoro の性能は eSTREAM Portfolio II に

表 2.12　eSTREAM Portfolio Profile II アルゴリズムのハードウェア実装性能 [10]

アルゴリズム	出力	最大動作周波数	スループット	論理規模	実装プロセス
	(bit/cycle)	(MHz)	(Mbps)	(kgate)	(μm)
Grain	1	724.6	724.6	1.3	0.13
	8	632.9	5063.2	2.2	
Trivium	1	327.9	327.9	2.6	
	8	471.7	3773.6	3.0	
Mickey 2.0	1	454.5	454.5	3.2	
Enocoro-80	8	274.7	2197.6	2.7	0.18
AES	2.37	131.2	311.0	5.4	0.11
	0.124	80.0	10.0	3.4	0.35

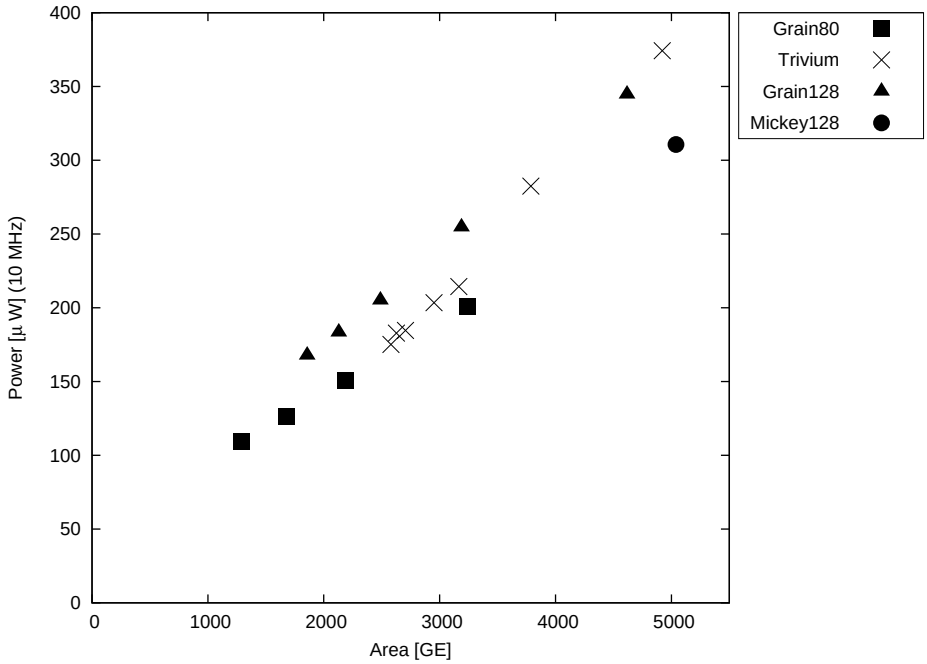


図 2.1　eSTREAM Portfolio Profile II アルゴリズムのハードウェア実装規模と消費電力 [10]

掲載のアルゴリズムと同程度である。消費電力に関する情報は見つかっていない。

表 2.13　Enocoro のハードウェア実装性能

アルゴリズム	最大動作周波数 (MHz)	スループット (Mbps)	論理規模 (kgate)	実装プロセス (μm)
Enocoro-80 [12]	274.7	2197.6	2.7	0.18
Enocoro-128v2 [13]	440.0	3520.0	4.1	0.09

参考文献

[1] Steve Babbage, Christophe De Cannière, Anne Canteaut, Carlos Cid, Henri Gilbert, Thomas Johansson, Matthew Parker, Bart Preneel, Vincent Rijmen, and Matthew Robsha, “The eSTREAM Portfolio (rev. 1),” September 8, 2008.

[2] Carlos Cid and Matt Robshaw, The eSTREAM Portfolio in 2012, ECRYPT II, European Network of Excellence in Cryptology II, 2012.

[3] M. Robshaw and O. Billet, editors. New Stream Cipher Designs: The eSTREAM Finalists, LNCS 4986, pp. 267–293. Springer.

[4] M. Boesgaard, M. Vesterager, and E. Zenner. A Description of the Rabbit Stream Cipher Algorithm. Network Working Group, Request for Comments 4503. `http://tools.ietf.org/html/rfc4503`

[5] ISO/IEC 18033-4. Information technology – Security techniques – Encryption algorithms – Part 4: Stream ciphers. 2005.

[6] ISO/IEC 29192-3:2012, Information technology – Security techniques – Lightweight cryptography – Part 3: Stream ciphers, 2012.

[7] A Description of the Rabbit Stream Cipher Algorithm, RFC4503.

[8] “The Salsa20 Stream Cipher for Transport Layer Security,” draft-josefsson-salsa20-tls-04, November 26, 2013.

[9] eBASC: ECRYPT Benchmarking of Stream Ciphers

[10] T. Good and M. Benaissa, “Hardware performance of eStream phase-III stream cipher candidates,” SASC2008, Feb 2008.

[11] I. Dinur and A. Shamir. Breaking Grain-128 with Dynamic Cube Attacks. In A. Joux, editor, Proceedings of FSE 2011, LNCS 6733, pp. 167–187, Springer, 2011.

[12] Watanabe, D., Ideguchi, K., Kitahara, J., Muto, K., Furuichi, H., Kaneko, T. “Enocoro-80: A Hardware Oriented Stream Cipher”. Third International Conference on Availability, Reliability and Security, pp. 1294–1300, IEEE Press, New York, 2008.

[13] 日立製作所, 「ストリーム暗号 Enocoro 評価書」, 2010.

[14] KDDI 研究所, 「ストリーム暗号 KCipher-2」, CRYPTREC シンポジウム 2010, 2010.

2.4 軽量ハッシュ関数

本章では、軽量暗号技術の現状調査として、代表的なハッシュ関数の安全性と実装性能に関する調査結果を報告する。

2.4.1 調査対象アルゴリズム

調査対象とするハッシュ関数アルゴリズムは、軽量ハッシュ関数として主要国際会議で提案された PHOTON、SPONGENT、QUARK、および、CRYPTREC 暗号リストに含まれる SHA-2、SHA-3 として選定された Keccak の 5 方式とする。調査文献として、SHA-2、SHA-3 については NIST の SHA-3 Competition から、その他のアルゴリズムについては提案論文や主要国際会議の論文を中心に調査した。

2.4.2 安全性

アルゴリズムの構造に基づいた Generic attacks に対する安全性に関し、各アルゴリズムの preimage attack、2nd-preimage attack、collision attack に対する計算複雑度を表 2.14 に示す。Sponge 構造である Keccak、PHOTON、SPONGENT、QUARK については、“Parameter” の “n” は hash size を、“c” は capacity を、“r” は rate を表している。Merkle-Damgard 構造である SHA-1、SHA-2 に関しては、それぞれ hash size、internal state size、message block size を表している。アルゴリズム特有の性質を利用した攻撃手法については、SHA-2 や Keccak に対しては数多くの報告があるが（CRYPTREC 技術報告書 [1],[2] 参照)、それ以外のアルゴリズムに対してはまだ十分に議論されていないため安全性に欠陥がある可能性がある。

2.4.3 ハードウェア実装性能

ハードウェア実装性能に関する調査結果を表 2.15, 2.16 に纏める。これらの評価値は様々な文献から抽出したものであり、同一環境で評価されたものではないことに注意されたい。参考情報として AES ベースのハッシュ関数の実装性能（推定値を含む）についても記載した。表中の “Area” はゲート規模を、“Latency” は internal permutation P (または internal block-cipher E) のクロック数を、“FOM(Figure of Merit)” はエネルギー効率を表す指標（スループットとゲート規模の二乗の比）を、“Power” は平均消費電力を表している（動作周波数 100kHz での性能を示す)。この結果より、軽量ハッシュ関数と呼ばれるアルゴリズムは、回路規模を数 kGE 程度に収めることを優先し、スループットはあまり高くない設計のものが多いことが分かる。

2.4.4 ソフトウェア実装性能

ソフトウェア実装性能については、軽量暗号という観点から非力な CPU（Atmel AVR ATtiny45 8-bit RISC microcontroller）での性能を示した CARDIS2012 の発表論文 [12] を引用する（表 2.17)。

2.4.5 まとめ

CRYPTREC 暗号リストに記載の SHA-2 と SHA-3 として選定された Keccak、および軽量ハッシュ関数に分類される PHOTON、SPONGENT、QUARK について、提案論文を中心に安全性と実装性能について調査した。軽量ハッシュ関数は 64〜128 ビットセキュリティの安全性での実装性能を重視したものが多く、ハードウェア実装性能において

表 2.14　各アルゴリズムの安全性

Algorithm	Hash [bit]	Parameter[bit] n	c	r	Security[bit] Pre	2nd-Pre	Col	Source
SHA-1	160	160	160	512	160	160	80	[8]
SHA-256	256	256	256	512	256	256	128	[8]
Keccak-f[200][*1]	128	200	128	72	128	128	64	[5]
Keccak-f[400][*1]	160	400	256	144	160	160	80	[5]
PHOTON-80	80	80	80	20	64	40	40	[8]
PHOTON-128	128	128	128	16	112	64	64	[8]
PHOTON-160	160	160	160	36	124	80	80	[8]
PHOTON-224	224	224	224	32	192	112	112	[8]
PHOTON-256	256	256	256	32	224	128	128	[8]
SPONGENT-88	88	88	80	8	80	40	40	[10]
SPONGENT-128	128	128	128	8	120	64	64	[10]
SPONGENT-160	160	160	160	16	144	80	80	[10]
SPONGENT-224	224	224	224	16	208	112	112	[10]
SPONGENT-256	256	256	256	16	240	128	128	[10]
U-QUARK	128	136	128	8	128	64	64	[11]
D-QUARK	160	176	160	16	160	80	80	[11]
S-QUARK	224	256	224	32	224	112	112	[11]

SHA-2 と比較すると、回路規模の面で大きな優位性があるものの、速度面では必ずしも優れているわけではなく、レイテンシは勝るものもあるがスループットに関しては概ね劣っていることが分かった。以上の観点から、今回調査した軽量ハッシュ関数は、特に回路規模に制限があるデバイスや低レイテンシが要求されるアプリケーションでの利用が適していると考えられる。

[*1] Keccak-f[] は置換関数であることに注意

表 2.15　ハードウェア実装性能

Algorithm	Area [GE]	Latency [clk]	Throughput [kbps]	FOM	Power [uW]	Proc. [nm]	Source
SHA-1	6,812	450	113.78	24.52	11.0	250	[3]
SHA-256	8,588	490	104.48	14.17	11.2	250	[4]
KECCAK-f[200]	2,520	900	8.00	12.60	5.60	130	[5]
	4,900	18	400.0	166.6	27.6	130	[5]
KECCAK-f[400]	5,090	1,000	14.40	5.56	11.5	130	[5]
	10,560	20	720.00	64.57	78.1	130	[5]
KECCAK-f[1600]	20,790	1,200	90.66	2.10	44.9	130	[5]
	47,630	24	4,533	19.98	315.1	130	[5]
AES-based DM scheme-128	>4,400	-	<12.4	-	-	-	[7]
AES-based Hirose scheme-256	>9,800	-	<12.4	-	-	-	[7]
PHOTON-80/20/16	865	708	2.82	37.73	1.59	180	[8]
	1,168	132	15.15	111.13	2.70	180	[8]
	1,067	708	2.82	24.77	14.0	45	[9]
	1,567	132	15.15	61.70	39.9	45	[9]
PHOTON-128/16/16	1,122	996	1.61	12.78	2.29	180	[8]
	1,708	156	10.26	35.15	3.45	180	[8]
	1,394	996	1.61	8.29	17.2	45	[9]
	2,172	156	10.26	21.75	49.6	45	[9]
PHOTON-160/36/36	1,396	1332	2.70	13.87	2.74	180	[8]
	2,117	180	20.00	44.64	4.35	180	[8]
	1,741	1332	2.70	8.91	19.4	45	[9]
	2,849	180	20.00	24.64	65.8	45	[9]
PHOTON-224/32/32	1,735	1716	1.86	6.19	4.01	180	[8]
	2,786	204	15.69	20.21	6.50	180	[8]
	2,142	1716	1.86	4.05	22.6	45	[9]
	3,586	204	15.69	12.20	78.8	45	[9]
PHOTON-256/32/32	2,177	996	3.21	6.78	4.55	180	[8]
	4,362	156	20.51	10.78	8.38	180	[8]
	2,675	996	3.21	4.49	51.6	45	[9]
	5,335	156	20.51	7.21	248.	45	[9]

表 2.16　ハードウェア実装性能 (続)

Algorithm	Area [GE]	Latency [clk]	Throughput [kbps]	FOM	Power [uW]	Proc. [nm]	Source
SPONGENT-88	738	990	0.81	14.9	1.57	130	[10]
	1,127	45	17.78	139	2.31	130	[10]
	869	990	0.81	10.7	16.5	45	[9]
	1,237	45	17.78	116	38.7	45	[9]
SPONGENT-128	1,060	2,380	0.34	3.03	2.20	130	[10]
	1,687	70	11.43	40.2	3.58	130	[10]
	1,257	2,380	0.34	2.15	21.1	45	[9]
	1,831	70	11.43	34.1	53.2	45	[9]
SPONGENT-160	1,329	3,960	0.40	2.26	2.85	130	[10]
	2,190	90	17.78	37.1	4.47	130	[10]
	1,572	3,960	0.40	1.62	24.6	45	[9]
	2,406	90	17.78	30.7	73.5	45	[9]
SPONGENT-224	1,728	7,200	0.22	0.7	3.73	130	[10]
	2,903	120	13.33	15.8	5.97	130	[10]
	2,070	7,200	0.22	0.5	31.4	45	[9]
	3,220	120	13.33	12.9	96.0	45	[9]
SPONGENT-256	1,950	9,520	0.17	0.45	4.21	130	[10]
	3,281	140	11.43	10.6	6.62	130	[10]
	2,323	9,520	0.17	0.32	34.2	45	[9]
	3,639	140	11.43	8.63	110.	45	[9]
U-QUARK	1,379	544	1.47	7.73	2.44	180	[11]
	2,392	68	11.76	20.6	4.07	180	[11]
	1,744	544	1.47	4.83	51.2	45	[9]
	3,215	68	11.76	11.4	89.4	45	[9]
D-QUARK	1,702	704	2.27	7.84	3.10	180	[11]
	2,819	88	18.18	22.9	4.76	180	[11]
	2,200	704	2.27	4.69	58.6	45	[9]
	3,695	88	18.18	13.3	87.7	45	[9]
S-QUARK	2,296	1,024	3.13	5.94	4.35	180	[11]
	4,640	64	50.0	23.2	8.39	180	[11]
	3,001	1,024	3.13	3.48	81.6	45	[9]
	6,155	64	50.0	13.2	146	45	[9]

表 2.17　ソフトウェア実装性能

Algorithm	Digest size [bits]	Code size [bytes]	RAM data [bytes]	RAM state & others [bytes]	RAM stack	Cycle count (8byte msg)	Cycle count (50byte msg)	Cycle count (100byte msg)	Cycle count (500byte msg)
SHA-256	256	1090	64	73	6	33,600	33,600	66,815	266,105
Keccak[r=40,c=160]	160	752	5	45	3	58,063	162,347	278,269	1,205,627
Keccak[r=144,c=256]	256	608	18	92	4	90,824	181,466	317,221	1,313,291
Keccak[r=1088,c=512]*	256	868	136	240	4	178,022	178,022	179,494	716,483
PHOTON-160/36/36	160	764	9	39	11	620,921	1,655,364	2,793,265	11,999,914
PHOTON-256/32/32	256	1,244	4	68	10	254,871	486,629	787,896	3,105,396
SPONGENT-160/160/80	160	598	10	60	6	795,294	2,783,241	4,771,186	20,674,746
SPONGENT-256/256/128	256	364	16	96	5	1,542,923	3,856,916	6,170,900	25,454,100
D-QUARK	176	974	2	42	5	631,871	1,516,685	2,570,035	10,996,835
S-QUARK	256	1106	4	60	5	708,783	1,417,611	2,339,023	9,427,023

参考文献

[1] 盛合志帆, ハッシュ関数の安全性に関する技術調査報告書, CRYPTREC 技術報告書 No.0213, http://www.cryptrec.go.jp/estimation.html#2004, 2004.

[2] 金子敏信, SHA-256/-384/-512 の評価報告, CRYPTREC 技術報告書 No.0503, http://www.cryptrec.go.jp/estimation.html#2005, 2005.

[3] Mooseop Kim and Jaecheol Ryou, Power Efficient Hardware Architecture of SHA-1 Algorithm for Trusted Mobile Computing. In Sihan Qing, Hideki Imai, and Guilin Wang, editors, *Information and Communications Security, 9th International Conference — ICICS 2007*, volume 4861 of *Lecture Notes in Computer Science*, pages 375-385, Springer-Verlag, Berlin, Heidelberg, 2007.

[4] Mooseop Kim, Jaecheol Ryou, and Sungik Jun, Efficient Hardware Architecture of SHA-256 Algorithm for Trusted Mobile Computing. In Moti Yung, Peng Liu, and Dongdai Lin, editors, *Information Security and Cryptology — ISC 2008*, volume 5487 of *Lecture Notes in Computer Science*, pages 240-252, Springer-Verlag, Berlin, Heidelberg, 2009.

[5] Elif Bilge Kavun and Tolga Yalcin, A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications, In Siddika Berna Ors Yalcin, editor, *Radio Frequency Identification: Security and Privacy Issues — RFIDsec 2010*, volume 6370 of *Lecture Notes in Computer Science*, pages 258-269, Springer-Verlag, Berlin, Heidelberg, 2010.

[6] Luca Henzen, Pietro Gendotti, Patrice Guillet, Enrico Pargaetzi, Martin Zoller, and Frank K. Gürkaynak, Developing a Hardware Evaluation Method for SHA-3 Candidates, In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems — CHES 2012*, volume 6225 of *Lecture Notes in Computer Science*, pages 248-263, Springer-Verlag, Berlin, Heidelberg, 2010.

[7] Andrey Bogdanov, Gregor Leander, Christof Paar, Axel Poschmann, Matt J. B. Robshaw, and Yannick Seurin, Hash Functions and RFID Tags: Mind the Gap, In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 283-299, Springer-Verlag, Berlin, Heidelberg, 2008.

[8] Jian Guo, Thomas Peyrin, and Axel Poschmann, The PHOTON Family of Lightweight Hash Functions, In Phillip Rogaway, editor, *Advances in Cryptology — CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 222-239, Springer-Verlag, Berlin, Heidelberg, 2011.

[9] Andrey Bogdanov, Miroslav Knežević, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede, SPONGENT: The Design Space of Lightweight Cryptographic Hashing, volume 62, issue 10, pages 2041-2053, IEEE Transactions on Computers, 2013.

[10] Andrey Bogdanov, Miroslav Knežević, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede,

SPONGENT: A Lightweight Hash Function, In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 312-325, Springer-Verlag, Berlin, Heidelberg, 2011.

[11] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Mara Naya-Plasencia, Quark: A Lightweight Hash, In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems — CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 1-15, Springer-Verlag, Berlin, Heidelberg, 2010.

[12] Josep Balasch, Barış Ege, Thomas Eisenbarth, Benoit Gérard, Zheng Gong, Tim Güneysu, Stefan Heyse, Stéphanie Kerckhof, François Koeune, Thomas Plos, Thomas Pöppelmann, Francesco Regazzoni, François-Xavier Standaert, Gilles Van Assche, Ronny Van Keer, Loïc van Oldeneel tot Oldenzeel, and Ingo von Maurich, Compact Implementation and Performance Evaluation of Hash Functions in ATtiny Devices, In Stefan Mangard, editor, *Smart Card Research and Advanced Applications — CARDIS 2012*, volume 7771 of *Lecture Notes in Computer Science*, pages 158-172, Springer-Verlag, Berlin, Heidelberg, 2013.

2.5 軽量メッセージ認証コード

本章では、軽量なメッセージ認証コード (Message Authentication Code: MAC) に関する調査報告を行う。

MAC は、nonce 入力の有無によって、probabilistic MAC と deterministic MAC に分けられる。nonce 入力を持つ probabilistic MAC は、リプレイ攻撃に耐性を持つ。また、Wegman と Carter が提案した、universal hash function から MAC を構成する方法 [9] は nonce 入力が必須である。universal hash function をベースとする MAC は、代数的演算を用いることを特徴としてお、64 ビットレジスタ上での演算などを高速に行うことができる実装環境では優れた処理性能を発揮する。

一方、ブロック暗号から構成する CMAC [18] や、ハッシュ関数から構成する HMAC [2, 26] は deterministic MAC として定義される（nonce を prefix とすることで probabilistic MAC として用いることもできる)。このような、暗号プリミティブからモードとして MAC を定義する場合、実装環境に合わせて軽量な暗号プリミティブを使用することで、軽量な MAC となることが期待できる。

2.5.1 universal hash function を用いる構成法

Wegman と Carter により、ユニバーサルハッシュ関数 h から安全な MAC を構成できることが知られている [27]。Wegman-Carter 方式による MAC の構成は $MAC(m,k,r) = h(m,k) + b(r)$ で定義される。ただし、ここで $b(r)$ は one-time key である。主要なユニバーサルハッシュ関数はいずれも代数的な演算で構成されており、これらの演算が高速に実行できる環境では優れた処理性能を実現する。

まず、Wegman らが提案した多項式を用いる方式 (polynomial hashing) では、ユニバーサルハッシュ関数は $h(m,k) = \sum_i m_i k^i$ で定義される。polynomial hashing のアルゴリズム例として GMAC [13] や Poly1305 [3] がある。GMAC の演算は標数 2 の拡大体 GF(2^{128}) 上で、また、Poly1305 の演算は素体 $GF(2^{130}-5)$ 上で定義される。Saarinen は GMAC に弱鍵があることを指摘している [25]。また、Procter らは、この脆弱性が多項式の取り方に依らず存在すること、および任意の鍵を弱鍵と見做せることを示した [24]。しかし、Procter の攻撃では、弱鍵を検出する識別子のメッセージ長と、弱鍵の空間の大きさが等価であるため、現実的な脅威ではないと考えられる。

[15] によれば、Intel Haswell アーキテクチャ上では GMAC (GHASH) の漸近的な処理速度は 0.4 cycle/Byte である。また、表 2.18 は、[4] で提供されている Poly1305-AES の処理性能から抜粋したものである。

表 2.18　Poly1305-AES の処理速度 [4](単位：cycles/Byte)

	データ長			
	64	256	1024	long
Pentium III	16.3	6.9	5.1	4.4
Pentium 4	18.7	8.0	5.3	4.5
Athlon	13.1	5.7	3.7	3.2

また、Halevi と Krawczyk は内積を用いる方式 MMH を提案した [17]。MMH はメッセージ $m = \{m_1, \ldots, m_n\}$ と等長の鍵ストリーム $k = \{k_1, \ldots, k_n\}$ に対して $h(m,k) = \sum_i m_i \cdot k_i$ 定義される。UMAC [7] や Badger [8] は MMH と同じく内積方式であるが、MMH が有限体上の演算を用いて定義されているのに対して、ソフトウェア実装に適した $Z/2^wZ$ 上の演算を用いる点が異なる。

MMH 方式では、一般に鍵をメッセージと等長のビット列に伸長して内積を計算する。したがって、他の方式に比べて事前処理に要するコストが大きくなる。また、拡大鍵を保持するためのメモリ使用量が増大する傾向にあり、複数の相手と通信を行うようなケースでは、メモリを圧迫する可能性がある。[7] や [8] では、安全に拡大鍵を使い回す方法や、tree-hash との組み合わせにより拡大鍵の量を削減する方法が紹介されている。

表 2.19 は、[21] で報告されている UMAC の処理性能である。報告されている数値からの推定になるが、UMAC の性能には、少くとも鍵を伸長する事前処理は含まれていないと考えられる。

表 2.19　Pentium 4 上での UMAC の処理速度 [21](単位：cycles/Byte)

	データ長			
タグ長	64	256	1024	long
32	8.3	2.4	0.9	0.6
64	12.0	3.5	1.4	1.0
96	15.1	4.5	1.9	1.5

また、表 2.20 は [8] で報告されている、タグ長が 64 ビットの Badger の処理速度である。

表 2.20　Pentium III および Pentium 4 上での Badger の処理速度 [8]

	事前処理	メッセージ処理	最終処理
Pentium III	4,093 cycles	2.2 cycles/Byte	433 cycles
Pentium 4	5,854 cycles	1.3 cycles/Byte	800 cycles

上に挙げた方式の実装性能はいずれも、CPU が 64 ビットアーキテクチャやベクトル演算を利用可能な環境、もしくは多大な事前計算テーブルをメモリに展開できる環境において実現されたものであり、計算機能力が貧弱な環境には適していない可能性が高い。

2.5.2　暗号プリミティブを用いる構成法

暗号プリミティブから MAC を構成する方法として、ブロック暗号から構成する CMAC [18] や、ハッシュ関数から構成する HMAC [2, 26] がある。ISO/IEC 9797 [28, 29] には、CMAC や HMAC の他にも、CBC-MAC のバリエーションなどが規定されている。Bertoni らが [5] でスポンジ関数を提案して以降、置換をベースとする暗号機能の研究がさかんになった。MAC の構成法としては、secret-prefix 方式が一般的であり、Bertoni らにより、その安全性が証明されている [6]。多くの軽量ハッシュ関数はスポンジ関数から構成されているので、上記の secret-prefix 方式を用いることが可能である。スポンジ関数の secret-prefix 方式は最終処理が不要であるため、メッセージ長が短い場合には、HMAC に比べて処理時間が短いことが期待される。

暗号プリミティブが疑似ランダム関数 (疑似ランダム置換) であることを利用するのではなく、その写像の一様性のみを利用する方式も存在する。Daemen らは、メッセージ処理を行う関数として、AES のラウンド関数 4 段 (鍵無し) を用いる Pelican を提案した [10, 11]。Pelican 2.0 [11] の安全性は証明されていない。しかし、現実的な攻撃も報告されていない。Minematsu らは、同じく AES のラウンド関数 4 段 (鍵付き) を用いる PC-MAC-AES を提案した [22]。PC-MAC-AES は、ベースとなる関数の最大差分確率を前提として安全性が証明されている。したがって、安全性の

観点では、Pelican よりも PC-MAC-AES が優れている。いずれのアルゴリズムも、AES 以外の軽量ブロック暗号をベースに構成することが可能であるが、事前に最大差分確率の評価が必須である。

実装性能では、Pelican や PC-MAC-AES は、いずれも漸近的な性能が CMAC-AES の 2.5 倍である。ただし、いずれも事前処理や最終処理に AES の暗号化 1 回以上の処理を行うため、メッセージ長が短い場合にはアドバンテージが小さくなる。また、PC-MAC-AES の処理速度は拡大鍵の量とトレードオフの関係にあり、漸近的な処理速度に近づくためには、メモリ使用量が増大する。したがって、実装性能の観点では Pelican が優位である場合が多い。

これらの他に、独自の暗号プリミティブを用いる方式として、Mouha らは非線形置換を用いる Chaskey を提案した [23]。Chaskey は 1-key Even-Mansour ブロック暗号の CMAC と解釈することが可能である。また、非線形置換は Skein, SipHash [1] と同様、ARX 演算をベースにしている。事前処理、最終処理が無いため、短いメッセージに対して効率的であると考えられる。表 2.21 は [23] で報告されている、Chaskey の処理速度である。

表 2.21　Cortex-M 上での Chaskey の処理速度 (cycles/Byte) [23]

	データ長	
	16	128
Cortex-M0	21.3	18.3
Cortex-M3/M4	10.6	7.0

参考文献

[1] Jean-Philippe Aumasson and Daniel J. Bernstein. “SipHash: A Fast Short-Input PRF”. *INDOCRYPT*, LNCS 7668, pages 489–508, Springer, 2012.

[2] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. “Keying Hash Functions for Message Authentication”. *Advances in Cryptology, CRYPTO'96*, LNCS 1109, pages 1–15, Springer, 1996.

[3] Daniel J. Bernstein. “The Poly1305-AES Message-Authentication Code”. *Fast Software Encryption, FSE'05*, LNCS 3557, pages 32–49, Springer, 2005.

[4] Daniel J. Bernstern. “Poly1305-AES speed tables”. `http://cr.yp.to/mac/speed.html`.

[5] G. Bertoni, J. Daemen, M. Peeters and G. Van Assche, “Sponge functions,” ECRYPT Hash Workshop, May 2007.

[6] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, “On the security of the keyed sponge construction”. *Symmetric Key Encryption Workshop, SKEW'11*, 2011.

[7] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. “UMAC: Fast and Secure Message Authentication”. *Advances in Cryptology, CRYPTO'99*, LNCS 1666, pages 216–233. Springer, 1999.

[8] M. Boesgaard, T.Christensen and E. Zenner, “Badger – A fast and provably secure MAC.” *Applied Cryptgraphy and Network Security*, LNCS 3531, pagets 176–191, Springer, 2005.

[9] J. Lawrence Carter and Mark N. Wegman. “Universal Classes of Hash Functions”. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[10] Joan Daemen and Vincent Rijmen. “A New MAC Construction ALRED and a Specific Instance ALPHA-MAC”. *Fast Software Encryption, FSE'05*, LNCS 3557, pages 1–17, Springer, 2005.

[11] Joan Daemen and Vincent Rijmen. “The MAC Function Pelican 2.0”. *IACR Cryptology ePrint Archive*, 2005:88, 2005. `https://eprint.iacr.org/2005/088.pdf`.

[12] Morris Dworkin. “Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication”. NIST special publication 800-38b, May 2005. `http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf`.

[13] Morris Dworkin. “Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC”. NIST special publication 800-38d, November 2007. `http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf`.

[14] Shimon Even and Yishay Mansour. “A Construction of a Cioher From a Single Pseudorandom Permutation”. *Advances in Cryptology, ASIACRYPT*, LNCS 739, pages 210–224. Springer, 1991.

[15] Shay Gueron. “AES-GCM software performance on the current high end CPUs as a performance baseline for CAESAR competition.” *Directions in Authenticated Ciphers, DIAC 2013*. 2013.

[16] Niels Ferguson. "Authentication weaknesses in GCM". Comments submitted to NIST Modes of Operation Process, May 2005.

[17] Shai Halevi and Hugo Krawczyk, "MMH: Software message authentication in the Gbit/second rate". *Fast Software Encryption, FSE'97*, LNCS 1267, pages 172–189, Springer, 1997.

[18] Tetsu Iwata and Kaoru Kurosawa. "OMAC: One-Key CBC MAC". *Fast Software Encryption, FSE'03*, LNCS 2887, pages 129–153. Springer, 2003.

[19] Antoine Joux. "Authentication Failures in NIST version of GCM". Comments submitted to NIST Modes of Operation Process, June 2006.

[20] Ted Krovetz. "Message Authentication on 64-Bit Architectures". *Selected Areas in Cryptography*, LNCS 4356, pages 327–341. Springer, 2006.

[21] Ted Krovetz. "UMAC Performance". `http://web.cs.ucdavis.edu/~rogaway/umac/2004/perf04.html`

[22] K. Minematsu and Y. Tsunoo. "Provably Secure MACs From Differentially-uniform Permutations and AES-based Implementations," *Fast Software Encryption, FSE'06*, LNCS 4047, pp. 226–241, 2006.

[23] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. "Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers." *Selected Areas in Cryptography, SAC'14*, 2014.

[24] Gordon Procter and Carlos Cid. "On Weak Keys and Forgery Attacks Against Polynomial-Based MAC Schemes," *Fast Software Encryption, FSE'13*, LNCS 8424, pp. 287–304, Springer, 2014.

[25] Markku-Juhani O. Saarinen. "Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes." *Fast Software Encryption, FSE'12*, 2012.

[26] James M. Turner. "The Keyed-Hash Message Authentication Code (HMAC)". FIPS PUB 198-1, National Institute of Standards and Technology, July 2008. `http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf`.

[27] Mark N. Wegman and J. Lawrence Carter. "New Hash Functions and Their Use in Authentication and Set Equality". *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[28] ISO/IEC 9797-1:2011, Information technology – Security tecniques – Message authentication codes (MACs) – Part 1: Mechanisms using a block cipher, 2011.

[29] ISO/IEC 9797-2:2011, Information technology – Security tecniques – Message authentication codes (MACs) – Part 2: Mechanisms using a dedicated hash-function, 2011.

[30] ISO/IEC 9797-3:2011, Information technology – Security tecniques – Message authentication codes (MACs) – Part 3: Mechanisms using a universal hash function, 2011.

2.6 認証暗号

2.6.1 認証暗号の安全性

2.6.1.1 はじめに

共通鍵暗号を用いて、送りたい平文の暗号化と、改ざん検知のための認証タグの付与とを同時に行う、認証暗号（Authenticated Encryption, AE）と呼ばれる機能が知られている。本章では、AE について、特に安全性の側面から調査した結果を報告する。

まず、AE の形式的な定義、およびその安全性の概念として知られているものを整理し 2.6.1.2 章で報告する。基本的に AE の安全性は、暗号文が平文の関する情報をどれだけ隠しているか（秘匿性）と、不正な暗号文をどれだけ正確に検知できるか（真正性）の二つの軸で評価されている [7, 46]。ただし、これらを複合した単一評価軸の存在や、平文以外の入力変数である初期ベクトルなどの形式、生成モデルの違いによりいくつかバリエーションを生じている。これらの違いを意識した安全性の比較が必要である。

次に 2.6.1.6 章以降にて具体的な構成方法を紹介する。AE の構成方法は多岐にわたり、特に用いる暗号学的プリミティブによって全体構成が大きく変化する。ここではもっとも普及しているアプローチの一つである、ブロック暗号をベースとした方式、すなわちブロック暗号利用モード（以下モード）により実現される方式に絞って説明を行い、これらが満たす安全性を示す。

2.6.1.2 認証暗号の形式と安全性

■基本的な入出力　まず、認証暗号の入出力について解説する。認証暗号の処理は一般に暗号化と復号からなる。秘密鍵 K を共有する 2 者間において、暗号化関数の入力は、もっとも典型的な場合、

- 初期ベクトル（Initial Vector, IV）N
- 平文 M
- ヘッダ H

となる。ここで、初期ベクトル N は暗号化のために補助的に用いる変数であり、通常暗号文と共に通信される（従って受信側は初期ベクトルを同期する必要がない）。初期ベクトルの長さは固定の場合も可変長の場合もある。典型的な生成方法は乱数によるものか、暗号化側が保持し、逐次更新する状態変数（カウンターなど）を用いるもの、あるいはその両方の組み合わせによるものである。

平文 M は暗号化の対象となる情報であり、一般に可変長の系列である。

ヘッダ H は associated data (AD) とも呼ばれ、暗号化はされないものの改ざんは防ぎたい情報のことを指す。例えば通信プロトコルのバージョン、パラメータ、中継ポイントでのルーティング情報などがある。こちらも一般に可変長の系列である。

なお厳密にはヘッダの存在しない方式を AE と呼び、ヘッダがある方式を AEAD（AE with AD）と呼ぶことがあるが、本稿では区別せず AE と呼ぶ。AEAD は方式によってはヘッダが存在せず、長さ 0 の変数と解釈して処理を行うことが可能であり、その意味では AE を包含する概念といえる。さらに、平文 M が存在しない場合を認める方式もあり、この場合の意図する処理はヘッダ H に対する、IV 付きのメッセージ認証コード（Message Authentication Code, MAC）となる。

暗号化処理の出力は、

- 暗号文 C
- タグ T

となる。暗号文 C の長さは通常 M と同じであり、タグ T は固定長である。送信する情報は (N, A, C, T) の 4 つ組となる。

復号処理の入力は上記４つ組であり、出力結果は、もし送信された情報が改ざんされていないと判断（受理）された場合には復号された平文 M となり、改ざんがあったと判断した場合は、単一のエラーメッセージとなる。

■入出力形式のバリエーション　基本的な AE には IV は必須であるが、方式によってはこれを不要とするものがある。例えば ANSI のスマートメータ関連規格（C12.22）において定義されている EAX-prime という方式では、IV とヘッダを組み合わせた変数を Cleartext と呼んでいる。また、いわゆる Deterministic AE (DAE), On-line AE (OAE) と呼ばれる AE のクラスにおいては、IV は存在せず、もし存在する場合には暗黙にヘッダに含まれるものとされていることが多い。

2.6.1.3　安全性の概念 – IV 付きの場合

上述のように、安全性の概念は典型的に秘匿性（Privacy）と完全性（Authenticity・Integrity）に分けて説明される。秘匿性とは、送信内容である (N, A, C, T) を得た攻撃者が元の平文 M に対する情報を得ることの困難性を表す指標であり、より端的には、暗号化関数の出力である (C, T) と同じ長さの乱数との判別困難性をもって表される。完全性とは、攻撃者が改ざんに成功することの困難性を表す指標である。ここで、改ざんとは、観測した正規の (N, A, C, T) とは異なる $(N', A', C', T') \neq (N, A, C, T)$ を、鍵を知ることなく生成し、これを受信者が受理する事象を指す。完全性は改ざん成功確率を攻撃者のクラスに関して最大値をとることで評価される。

よりフォーマルに記載するために、以下の表記を導入する。まず、$\mathcal{A}^{O_1, O_2, \ldots, O_c}$ を攻撃者 $\mathcal{A}$ が c 個のオラクル $O_1, \ldots, O_c$ に任意の順序でアクセスする環境を示すものとする。次に $\mathsf{AE}[\tau]$ を、τ-bit のタグを持つ AE であるとし、その暗号化と復号の関数をそれぞれ $\mathsf{AE}\text{-}\mathcal{E}_\tau$ と $\mathsf{AE}\text{-}\mathcal{D}_\tau$ とする。秘匿性の定義は以下で与えられる。まず $\mathsf{AE}[\tau]$ への nonce-respecting な q 選択平文攻撃とは $\mathsf{AE}\text{-}\mathcal{E}_\tau$ に対して $(N_1, H_1, M_1), \ldots, (N_q, H_q, M_q)$ を逐次的・適応的に与えて、$(C_1, T_1), \ldots, (C_q, T_q)$ を得ることをいう。ただしどの $i < j$ についても $N_i \neq N_j$ となることが条件である。ここで \$ を、(N, H, M) が与えられたもとで常に (C, T) と同じ長さの乱数を返す、ランダムビットオラクルであるとする。すると AE へ nonce-respecting な選択平文を行う攻撃者 $\mathcal{A}$ に対する PRIV アドバンテージは

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1].$$

と定義される。

次に完全性を定義する。$\mathcal{A}$ が $\mathsf{AE}[\tau]$ に対する選択暗号文攻撃を行う場合、$\mathsf{AE}\text{-}\mathcal{E}_\tau$ と $\mathsf{AE}\text{-}\mathcal{D}_\tau$ の両方に任意の順序でアクセスできる。$\mathcal{A}$ は nonce-respecting な選択平文クエリを $\mathsf{AE}\text{-}\mathcal{E}_\tau$ へ行うが、$\mathsf{AE}\text{-}\mathcal{D}_\tau$ には IV に関する制約はない。つまり暗号化クエリで用いた IV を復号クエリに用いてもよいし、復号クエリで重複した IV を用いてもよい。ただし自明な答えが返ってくる、暗号化で聞いた結果をそのまま復号に与えることだけは禁じる。このような攻撃者 $\mathcal{A}$ について、AE の完全性は、

$$\mathtt{Adv}^{\mathtt{auth}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau, \mathsf{AE}\text{-}\mathcal{D}_\tau} \text{ forges }]$$

で定義される。ここで右辺は、エラーシンボルである $\perp$ 以外を $\mathsf{AE}\text{-}\mathcal{D}_\tau$ から得るのイベントの確率を指す。

なお PRIV/AUTH といった表記については論文によっては異なる名称をとる場合もあるので注意が必要である。後述の IV なしのケースについても同様。

2.6.1.4 安全性の概念 – IV なしの場合

■Deterministic AE　IV が存在しない場合、暗号化の入力が (H, M)（もし H が存在すれば）で、出力が (C, T)、送信内容が (H, C, T) となる。また復号処理は (H, C, T) を入力とし、受理すれば M を出力、そうでなければ $\perp$ 出力となる。

このような AE の安全性については、大きく二つのバリエーションがある。一つ目は、Privacy については、平文の一致情報以上は漏らさないことを求める方式である。Authenticity については (H, C, T) に対する改ざん困難性を要求する。この概念は最初に Rogaway と Shrimpton によって提案され、Deterministic AE (DAE) と呼ばれることから、DAE security とも呼ばれている。

IV 付きの場合と同様に PRIV/AUTH で評価する場合について述べる。まず秘匿性は、$\mathsf{AE}[\tau]$ を DAE とみなし、AE-$\mathcal{E}_\tau$ へクエリ (H, M) を重複して行わない $\mathcal{A}$ について,

$$\mathtt{Adv}^{\mathtt{dpriv}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1]$$

で評価する。一方完全性は IV 付きのケースと同様

$$\mathtt{Adv}^{\mathtt{dauth}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau, \mathsf{AE}\text{-}\mathcal{D}_\tau} \text{ forges }],$$

で評価する。DAUTH の forge の意味は、non-trivial な復号のクエリ (H, C, T)（すなわち (H, M) を暗号化クエリして (C, T) を得ていない）について $\perp$ 以外のレスポンスを得ることを指す。それぞれの指標を DPRIV,DAUTH とここでは呼ぶことにする。なお Rogaway と Shrimpton は同時にこの二つをまとめた指標として DAE-advantage を提案している。これは、

$$\mathtt{Adv}^{\mathtt{dae}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau, \mathsf{AE}\text{-}\mathcal{D}_\tau} \Rightarrow 1] - \Pr[\mathcal{A}^{\$, \perp} \Rightarrow 1]$$

という指標である。これは、攻撃者が DAE の暗号化と復号に両方アクセスできるもとで、自明な質問をしないで、random-bit oracle と $\perp$-oracle (常に $\perp$ を返すオラクル）の組と、実際の DAE 暗号化、復号の組との判別を行う困難性を示すものである。両者は基本的に等価な関係にあり、DPRIV と DAUTH（の上界）が求まれば、DAE-advantage の上界が求まり、またその逆も可能であることが示されている [48]。単一指標のほうがシンプルな表現ではあるが、従来指標との整合性、および実際の証明手続きを考えると、二軸での指標にも実用的価値が見いだせると思われる。

DPRIV が求めるものは、本質的に暗号文のどのビットも平文全体の情報を反映することであり、従って DAE には原理上平文全体を読み込まない限り暗号文の最初のブロックが計算できず、従ってオンライン処理（1 パス処理）が不可能である。

■On-line AE　もう一つのケースが、秘匿性において異なる平文間の prefix の一致だけ漏れることを許容し、それより後ろは漏らさない、とするものである。このような機能は一般的に On-line Cipher と呼ばれ、Bellare らの研究 [5] に端を発するものである。認証暗号として完全性も満たすよう拡張された方式も提案されており、On-line AE (OAE) と呼ばれている。

まず秘匿性は、$\mathsf{AE}[\tau]$ を OAE とみなし、AE-$\mathcal{E}_\tau$ へクエリ (H, M) を重複して行わない $\mathcal{A}$ について,

$$\mathtt{Adv}^{\mathtt{opriv}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau} \Rightarrow 1] - \Pr[\mathcal{A}^{\$^O} \Rightarrow 1]$$

で評価する。一方完全性は IV 付きのケースと同様

$$\mathtt{Adv}^{\mathtt{oauth}}_{\mathsf{AE}[\tau]}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathsf{AE}\text{-}\mathcal{E}_\tau, \mathsf{AE}\text{-}\mathcal{D}_\tau} \text{ forges }],$$

で評価する。本質的に等価な別の定義として、$\mathtt{Adv}^{\mathtt{oauth}}$ はまた、$\perp$ を常にエラーシンボル $\perp$ を返すオラクルと定義したうえで、$(\mathsf{AE}\text{-}\mathcal{E}_\tau, \mathsf{AE}\text{-}\mathcal{D}_\tau)$ と $(\mathsf{AE}\text{-}\mathcal{E}_\tau, \perp)$ との判別のアドバンテージと定義されることもある [11]。

なお $\$^O$ は random-bits on-line oracle と呼ばれる、prefix が一致する部分のみ同じ乱数を返すオラクルである。より正確には、ヘッダが同じ二つの入力 (H, M), (H, M'), $M \neq M'$ について、M, M' を n-bit ブロックに分割した表現を $M = M[1], \ldots, M[m]$, $M' = M'[1], \ldots, M'[m']$ とし、Length of Longest common prefix (LLCP) を、$\mathrm{LLCP}_n(M, M') = \max_i\{M[1], \ldots, M[i] = M'[1], \ldots, M'[i]\}$ とする。対応する暗号文が $C = C[1], \ldots, C[m]$, $C' = C'[1], \ldots, C'[m']$ のとき、$\mathrm{LLCP}_n(M, M') = i$ なら まず $C[1], \ldots, C[i] = C'[1], \ldots, C'[i]$ がランダムに選択され、残りの系列が独立かつランダムに選ばれる。このようなオラクルは過去のクエリを保持しその都度サンプリング（lazy sampling）を行うことで実現可能である。

また、Fleischmann ら [11] は DAE のケースと同様に、OPRIV と OAUTH をまとめた指標である CCA3-security を提案し、OPRIV と OAUTH との等価性を説明している。

DAE とは異なり、OAE は秘匿性の部分は On-line cipher と同等の安全性要件と同じであるため、オンライン処理が可能である。

■Nonce-misuse との関連　DAE, OAE ともに、IV がヘッダの一部に含まれているケースを考えることが可能である。この場合、上記の安全性基準は、IV が nonce として暗号化に用いられている限りは通常の IV 付き AE の安全性を保証し、IV の重複が暗号化で発生する場合には DAE/OAE 本来の安全性が保証される、ということを意味する。この性質は、特に DAE について Rogaway と Shrimpton により Misuse-resistent AE (MRAE) [48] と呼ばれているが、OAE に関しては達成できる安全性が DAE よりも弱いため、OAE も含めて MRAE と呼称すべきかどうかについては議論がある（例えば CAESAR メーリングリスト [1] の議論参照）。

2.6.1.5 計算量的仮定

上記の安全性概念・基準を達成するにあたり、用いられるブロック暗号に対する計算量的仮定としては以下のものがある。ブロック暗号のブロックサイズを n ビット、またその暗号化関数を E_K, 復号関数を D_K とすると、

- 疑似ランダム関数（Pseudorandom Function, PRF）：選択平文攻撃において n-bit ランダム関数との計算量的判別困難性を有する鍵付き関数。
- 疑似ランダム置換（Pseudorandom Permutation, PRP）：選択平文攻撃において n-bit ランダム置換との計算量的判別困難性を有する鍵付き関数。
- 強疑似ランダム置換（Strong Pseudorandom Permutation, SPRP）：選択暗号文攻撃において n-bit ランダム置換との計算量的判別困難性を有する鍵付き関数。
- 関連鍵安全性（Related-key Security）：攻撃者が関連鍵を入力できる環境における、上記の計算量仮定のいずれか。例えば定数 c を鍵差分として入力できる PRP の場合、$K, K' = K \oplus c$ においてペア $(E_K, E_{K'})$ と ペアの独立なランダム置換 $(\mathsf{P}, \mathsf{P}')$ の判別困難性を意味する。

2.6.1.6 方式説明における記法

次節から具体的な方式を取り上げ、それらの概略と、証明可能安全性について述べる。AE の実現方法は多様であるため、ここではブロック暗号をベースとした暗号利用モードにより実現されている例を中心に取り扱う。仕様の解説はおおまかなものにとどめる。また、安全性の評価を簡潔にするため、以下ではすべて n-bit ブロック暗号を用いるものとし、

- q：暗号化クエリの回数
- q_v：復号クエリの回数
- σ_p：暗号化のクエリ (N, A, M) のトータルのブロック長
- σ_a：暗号化のクエリ (N, A, M) および復号のクエリ (N, A, C, T) のトータルのブロック長
- τ：タグのビット長

というパラメータ群を用いて攻撃者 $\mathcal{A}$ を定義し、$\mathcal{A}$ に対する安全性評価指標（バウンド）を表すことにする。攻撃者 $\mathcal{A}$ の計算量を便宜的に t とするが、本稿におけるバウンドの式では陽には現れないため省略する。また特段断らない限り、バウンド中の定数は略すこととする。実際の定数、および具体的なパラメータの設定においては、必要に応じて引用文献を参照のこと。認証暗号方式 XXX について、$\mathtt{XXX}[E, \tau]$ を、用いるブロック暗号が E で、タグ長が τ ビットとした実現例とする。多くの場合 $1 \leq \tau \leq n$ である。また、$\mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}')$, $\mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}')$ を $\mathcal{A}$ から求まる $\mathcal{A}'$ による、E に対する疑似ランダム置換、および強疑似ランダム置換との判別可能性を表すものとする。$\mathcal{A}'$ のパラメータは、計算量を含め $\mathcal{A}$ から決まるため方式ごとに定義が必要だが、以下では、$\mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}')$ 中の $\mathcal{A}'$ はすべて $O(\sigma_p)$（定数は一般に小さい）回の CPA クエリを行う、計算量 $O(t\sigma_p)$ の攻撃者となる。同様に $\mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}')$ 中の $\mathcal{A}'$ はすべて $O(\sigma_a)$ 回の CCA クエリを行う、計算量 $O(t\sigma_a)$ の攻撃者となる。使うブロック暗号の鍵の数が 2 以上の場合、一般的にこれらの項にも係数が出てくるが、こちらも省略するものとする。

2.6.1.7 IV 付き、レート 2 の方式

平文 M の 1 ブロックあたりの処理に必要なブロック暗号の回数をレートと呼ぶことにする。このような方式は、一般的に安全な暗号化のモード（カウンターモードなど）とメッセージ認証コード（CMAC など）を異なる鍵で適切に組み合わせることで構成可能であり、これを generic composition と呼ぶ。以下で説明するものの中には generic composition と類似した構成も含まれるが、鍵が共通であるため、generic composition の安全性結果（[6, 41] など）を直接引用することはできない。

■CCM　設計者: Housley, Whiting, Ferguson により 2002 年に作られた [54]。
構成：CBC-MAC で (N, H, M) を処理して中間タグ T' を生成したのち、N および H, M の長さ情報からカウンターモード暗号化の IV を生成し、M と T' を連結した系列を暗号化し、暗号文 C とタグ T とする。いわゆる MAC-then-Enc という generic composition の形式をとる（ただし鍵は単一である）。このため、本質的に On-line 処理ができない。IV 長は 1 ブロック未満に制限されている。また、CBC-MAC 入力のフォーマットが本来不要な複雑さを持つ、という問題がある。
安全性：Johnson [23] により以下の安全性証明がなされている。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{CCM}[E,\tau]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{CCM}[E,\tau]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

つまり、クエリの総ブロック長が $2^{n/2}$ より十分小さく、復号クエリの回数が 2^τ より十分小さい限り、CCM は PRIV/AUTH の双方の意味において、用いるブロック暗号の疑似ランダム性に帰着される安全性を有するといえる。
このタイプのバウンドは IV 付き AE の中でもっともよく見られるものである。

■GCM　設計者: McGrew と Viega により 2004 年に作られた [30]。
構成：$n = 128$-bit のブロック暗号によるカウンターモード GCTR と、有限体 $\mathrm{GF}(2^n)$ 上の乗算を用いたユニバーサ

ルハッシュ関数である GHASH とを組み合わせている。全体構成としては Enc-then-MAC の構成に近い。IV N は任意長をとれるが、特に $|N| = 96$ の場合、$I = N$ とし I の下 32-bit をインクリメントした値を初期値とした GCTR で M を暗号化し C を得たのち、GHASH を (A, C) へ適用し、$E_K(I)$ との XOR によりタグ T を生成する。これ以外の長さでは $I = GHASH(N)$ としたのち同様の処理を行う。なお、処理量としては平文 m ブロック、ヘッダ a ブロック、IV x ブロックにつき $m + 1$ 回のブロック暗号コール、$a + m + x$ 回の GF 乗算を必要とする。乗算のコストと実装規模（コードサイズ、事前計算量など）は無視できないため、ブロック暗号のレートとしては 1 であるが、トータルの計算コスト、実装規模は下記のレート 1 の方式と同等ととらえることはできない。

安全性：当初、McGrew, Viega により安全性証明がなされた [31] が、後に岩田らにより誤りが発見され、成功確率は現実的ではないが理論的攻撃が示された [20]。これは 96-bit 以外の IV を用いる時にカウンタ衝突確率の上界評価が当初の証明より大幅に増加することを利用している。同時に、証明の誤りを修正した以下のバウンドが示された。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n} + \frac{2^{22} q \sigma_p \ell_N}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{GCM}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{2^{22} (q + q_v) \sigma_a \ell_N}{2^n} + \frac{q_v \ell_A}{2^\tau}$$

ただし ℓ_N と ℓ_A は暗号化および復号で用いた最大の IV ブロック長と、最大のヘッダブロック長である。上記のバウンドは特別に大きい係数 2^{22} のみ省略せずに記載している。またこの係数は IV を 96-bit に固定することで 1 とすることが可能であるため、安全性を確保する上ではこの設定が望ましい。

■EAX　設計者：Bellare, Rogaway, Wagner により 2004 年に作られた [7]。
構成：CMAC で N を処理した結果 $\widetilde{N}$ を初期値としたカウンターモードで M を暗号化し、C を得たのち、H, C を個別に CMAC で処理した結果の XOR をとり、さらに $\widetilde{N}$ との XOR もとることでタグ T を生成する。N は任意の可変長変数である。CMAC は 3 回コールされるが、それぞれ最初に異なる定数ブロックを挿入することで、独立な疑似ランダム関数として振る舞うようにしている。いわゆる MAC-then-Enc という generic composition の形式をとるが、鍵は単一である。
安全性：Bellare, Rogaway, Wagner により安全性証明がなされている。この証明は AUTH のバウンドが $q_v = 1$ のケースについてのみ扱っており、汎用的な変換方法を用いて $q_v \ge 1$ のケースのバウンドに変換すると、次数 3 の項 $\sigma^2 q_v / 2^n$ が出現するため birthday bound ではなくなることが知られていたが、最近峯松らの結果により改善された [36]。ここでは改善されたバウンドで示す。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{EAX}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{EAX}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

■CLOC と SILC　設計者：Iwata, Minematsu, Guo, Morioka による CLOC [17, 16] と、Iwata, Minematsu, Guo, Morioka, Kobayashi による SILC [18, 19] がある。いずれも CFB と CBC-MAC との組み合わせをベースとし、事前計算を要する入力マスクを無くし、実行中に必要なメモリ量を減らす構造をとり、また 64-bit ブロック暗号の利用も定義するなど、ローエンドデバイスでの動作を意識した方式となっている。CLOC は組み込みソフトウェアを、SILC は小規模ハードウェアを主なターゲットとおいている。安全性：CLOC は [17, 16] により、SILC は [19] により、下記のタイプの標準的なバースデーバウンド安全性が示されている。また、AUTH に関しては Nonce が暗号化で再利用されても安全性が保証されるという特徴を持つ。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{CLOC}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{CLOC}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{SILC}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{SILC}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

■CHM と CIP　ここまで示したのはすべて $2^{n/2}$ ブロックの質問によりバウンドが 1 になり安全性保証がなくなる、安全性に関していわゆるバースデー限界のある方式である。一方岩田 [15, 14] により、バースデー限界を超えた安全性を保証する方式が提案されている。CHM と CIP の二つがあり、いずれもカウンターモードの変種である CENC と、代数的なユニバーサルハッシュ関数との組み合わせである。カウンターモードは鍵ストリーム系列と乱数との判別が $O(2^{n/2})$ ブロック出力させることで可能となるのに対し、CENC は周期的にブロック暗号を追加コールし、その結果をカウンターモード出力へ加算することでバースデー限界を超えた安全性を保証するものである。正整数 w を用いて周期を 2^w（2^w ブロックおきに追加のコールを行う）とした場合、CENC のレートは $1 + 1/2^w$ となり、安全性のバウンドはおおよそ $\sigma^3/2^{2n} + \sigma/2^n$ となる。w は大きいほうがレートが下がるが安全性バウンドと事前計算量などに影響を及ぼすため、$n = 128$ のときは $4 \sim 8$ 程度が推奨される。CHM と CIP についてもほぼ同様の安全性バウンドが得られる。暗号化のレートもほぼ CENC 同様だが、平文ブロック数の GF(2^n) 乗算を要するため、前述のルールに従うとレートは $2 + 1/2^w$ となる。

2.6.1.8　IV 付き、レート 2 未満の方式

■OCB　設計者：正確には 3 つのバージョンが知られており、OCB1,2,3 と呼称される。OCB1 は Rogaway [47] により 2001 年に, OCB2 は同じく Rogaway [45] により 2004 年に、OCB3 は Krovetz と Rogaway [26] により 2011 年に作られた。

構成：ECB 暗号化の上下のブロックをマスク系列で XOR している。マスク系列は、IV N と何番目のブロックかを表すインデックス $i = 1, 2, \ldots$ とをブロック暗号で処理して、i についてシーケンシャルに生成する。平文 M をマスク付き ECB 暗号化した出力が暗号文 C となり、タグ T は平文の全ブロックの XOR（チェックサムと呼ばれる）を特別なマスクを入力側に付けた 1 ブロック ECB で暗号化することで得られる。これはヘッダが存在しないときの処理であり、ヘッダがある場合、並列実行可能な MAC である PMAC をヘッダに適用した結果と上記の T との XOR をタグとする。PMAC は上記のマスク付き ECB の出力全ブロックの XOR をもう一度マスク付き 1 ブロック ECB 暗号化するものである。復号においてはマスク付き ECB の復号をしたのち、得られた平文のチェックサムを暗号化して、タグとの一致をチェックする。この処理にはブロック暗号の復号関数を要する。この構造により、レート 1 を達成している。

OCB の各バージョンでマスク系列生成方式に違いがある。OCB1,3 は Gray code をベースとしており、基準となる n 個のブロック値をブロック暗号を用いて事前計算し、Gray code が示す順序に従って基準のブロック値を逐次的に XOR していくことでマスクを生成するのに対し、OCB2 はほぼ事前計算なしに逐次的に GF(2^n) 上の 2 倍算を繰り返すことでマスクを生成する。

安全性：各提案論文 [47, 45, 26] によりそれぞれのバージョンの安全性証明がなされている。基本的にはいずれも以下

の形で示される。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{OCB}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{OCB}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

ただし、調査した限りでは、OCB1 のみ上記の AUTH バウンドにおいて $q_v = 1$ のケースしか示されていないようである。

なお、最近の結果として、青木と安田 [3] により、OCB の各バージョンとも強疑似ランダム置換よりも弱い条件に帰着できることが示された。上記のバウンドにおいて PRIV は PRP のみで証明できることはほぼ自明であるが、[3] によると、AUTH においてもブロック暗号復号の結果に対する予測不能性（系列全体がランダムでなくとも満たしうる性質）があればよいことが分かる。

■類似方式　ほぼ OCB1 と同時期に発表された方式として Julta [25] による IAPM, IACBC, Gligor, Donescu [12] による XCBC がある。これらは構造的には OCB と同じ（もしくは ECB の内部にさらにブロック間のチェインを挟むものもある）であるが、マスク生成の部分に関して OCB がもっとも洗練されているといえる。

OCB と類似した構造で、マスクを用いずに ECB のブロックをチェインさせて、すなわち CBC 暗号化のような処理を行ってレート 1 の AE を達成しようとする試みもある（例えば PCBC とその変種 [33, 38], IOBC [39],IOC [44]）。OCB 以前に考えられた方法が多い。また [39] によるとそのほぼすべてに攻撃が発見されており、現在のところ安全性証明が与えられた方式はないとみられる。

■CCFB　設計者: Lucks [28] により 2005 年に作られた。
構成：ブロック暗号の入出力の一部のみを用いた CFB モードにより暗号化を行う。CFB で使われない入力部分は処理ブロックのインデックスが与えられ、出力部分は逐次的に XOR をとることでチェックサムとしている。ヘッダが存在しない基本的なバージョンでは、CFB のチェイン値の初期値は IV である。ヘッダが存在するバージョンを CCFB+H と呼ぶが、このバージョンでは、ヘッダを（0^n プリペンドした）CMAC へ適用した結果と IV の XOR をチェインの初期値とする。IV は 1 ブロックの値である。暗号化が終わった時点のチェイン値を暗号化し、チェックサムとの XOR を行いタグとする。タグの長さを τ ビットとすると、チェックサムの長さもこれと等しい。またチェイン値を a-bit とすると $a + \tau = n$ を満たすこととなる。例えば $n = 128$ のケースで $a = 96$, $\tau = 32$ とすることが提案されている。センサーネット系のメッセージ認証コードは 32-bit タグのケースが多く、そのようなケースにフィットすると考えられる。上記の構造により、ブロック暗号 1 回につき a-bit 平文を処理可能であるため、レートは n/a となる。例えば $a = (2/3)n$, $\tau = (1/3)n$ とするとレートは 1.5 となる。原理上は $n/(n-1)$ まで 1 に近づけられるが、タグの短さは AUTH バウンドの劣化に直結するため、適切なバランスを取る必要がある。並列処理が不可能であるが、1 パス暗号化が可能であり、逐次的な処理には適している[*2]。また OCB と異なり、ブロック暗号の暗号化関数のみを用いる。

安全性：Lucks [28] により以下の安全性証明がなされている。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{CCFB}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^a}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{CCFB}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^a} + \frac{1}{2^\tau}$$

ただし AUTH は $q_v = 1$ のケースを扱っている。

[*2] ただし論文のタイトルには Two-pass とある。

■復号関数を用いない方式　OCB がブロック暗号の復号関数を用いるのに対し、レート 1 を保持したままブロック暗号の暗号化関数のみで全体を構成しようとする試みがある。Liting らの iFeed [55, 1] は CBC 暗号化に似た形式（より具体的には暗号化が CBC 復号に類似）を持ち, レート 1 であるが復号が並列処理できない。峯松の OTR [35, 1] では 2 ラウンドフェイステル置換の構造を取り入れることで、2 ブロック単位での並列化が暗号化と復号で可能となっている。いずれも下記に示す標準的なバースデーバウンドの安全性を有している。ブロック暗号の強擬似ランダム性は必要とせず、擬似ランダム性のみを必要とする点が OCB とは異なる。iFeed の安全性証明は Liting ら [56] に記載されている。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{iFeed}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{iFeed}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

また OTR に関しては [35, 34] に記載されている。

$$\mathtt{Adv}^{\mathtt{priv}}_{\mathrm{OTR}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{auth}}_{\mathrm{OTR}[E,\tau]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n} + \frac{q_v}{2^\tau}$$

2.6.1.9　On-line AE 方式

■McOE　設計者: Fleischmann, Forler, Lucks, Wenzel [11] により 2012 年に作られた。
構成：Bellare らによる On-line cipher [4, 5] のアイディアをベースとした構成である。Rogaway と Zhang による、Tweakable ブロック暗号 [27] を用いた On-line cipher の構成方法 TC3 [49] に、さらにメッセージ認証の機能を追加した構成ともとらえることができる。

McOE ではまず、n-bit ブロック暗号 E_K をベースに、n-bit tweak, n-bit ブロックを持つ Tweakable ブロック暗号 $\widetilde{E}_K$ を構成する。構成方法は二つあり、それぞれ McOE-G, McOE-X と呼ばれる。McOE-X で用いる $\widetilde{E}_K$ では、tweak と鍵の XOR により tweak を処理する。具体的には E_K の鍵長 $|K| = n$ であり、平文 M, Tweak T について暗号文は $C = \widetilde{E}_K(T, M) = E_{K \oplus T}(M)$ となる。McOE-G で用いる $\widetilde{E}_K$ では、GF(2^n) 上の要素 X と Y の乗算 $H_Y(X)$ を用いる。具体的には、$|K| = 2n$ であり、$K = (K_1, K_2)$, $|K_i| = n$ と分けたのち、平文 M, Tweak T について暗号文は $C = \widetilde{E}_K(T, M) = E_{K_1}(M \oplus H_{K_2}(T)) \oplus H_{K_2}(T)$ となる。このようにして構成された $\widetilde{E}_K$ を用いて、TC3 の暗号化である Tweak chaining を行う。これは i 番目の平文ブロック $M[i]$ について暗号文ブロック $C[i]$ を $\widetilde{E}_K(S[i], M[i])$ とするものである。$S[i]$ はチェインさせる tweak であり、$S[i+1] = M[i] \oplus C[i]$ として更新する。初期値 $S[0]$ は 0^n である。タグの生成には、最初にヘッダを Tweak chaining で暗号化した結果（の最終ブロック）を Z とし、平文の後ろに Z を連結したのち Tweak chaining で暗号化した結果得られる最終ブロックをタグ T とする。なお平文がブロックサイズの等倍におさまらない場合は、tag-splitting と呼ばれる処理をさらに導入する必要がある（CBC 暗号化における Ciphertext stealing と呼ばれる処理に近い）。タグの長さは常に n bit である。

安全性：[11] により安全性証明がなされている。ここでは簡単のため tag-splitting の不要な、平文が常にブロックサイズの等倍であるケースのバウンドを示す（実際には CCA3 という $\mathtt{Adv}^{\mathtt{opriv}}$ と $\mathtt{Adv}^{\mathtt{oauth}}$ を組み合わせた評価で示して

いるが、証明の内部にて下記のように分解がなされている)。McOE-X と McOE-G それぞれ、

$$\mathtt{Adv}^{\mathtt{opriv}}_{\text{McOE-G}[E,n]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n},$$
$$\mathtt{Adv}^{\mathtt{oauth}}_{\text{McOE-G}[E,n]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n},$$
$$\mathtt{Adv}^{\mathtt{opriv}}_{\text{McOE-X}[E,n]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{rk\text{-}sprp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n},$$
$$\mathtt{Adv}^{\mathtt{oauth}}_{\text{McOE-X}[E,n]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{rk\text{-}sprp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n}$$

ここで $\mathtt{Adv}^{\mathtt{rk\text{-}sprp}}_{E}(\mathcal{A}')$ は鍵差分 T を入力できるもとでの CCA 攻撃へのアドバンテージ（すなわち、T を tweak とした Tweakable ブロック暗号の CCA セキュリティ）を示す。

McOE-G では安全性がブロック暗号の強疑似ランダム性に帰着されるが、McOE-X では $C = \widetilde{E}_K(T, M) = E_{K \oplus T}(M)$ という暗号化が T ごとに独立と見なせる、という計算量的仮定、すなわち 2.6.1.5 節で述べた関連鍵安全性（Related-key Security）を要する。鍵に Tweak を加算する部分を利用した McOE-G へのアタックが [32] で提案されているが、基本的には計算量 $O(2^{n/2})$ であり、証明自体の決定的な誤りを指摘するものとはなっていない。ただし、このアタックは鍵回復を可能とするものであり、証明が考慮する識別攻撃・改ざん攻撃よりも強い。また [32] は証明における計算量的仮定の置き方に関する問題を示しており、同種の構成を考える際の参考とはなるであろう。

■COPA　設計者: Andreeva ら [2] により 2013 年に作られた。
構成：McOE と異なり、On-line cipher のアイディアを明示的には利用していない。Tweakable ブロック暗号である XEX [45] をベースに、ECB ライクなレイヤーを二つずらして重ねることで構成されている。暗号化のレートは 2 であり、暗号化にはブロック暗号暗号化関数を 2 回、復号にはブロック暗号復号関数を 2 回用いる。CPA-secure な On-line cipher である COPE と、COPE をベースとした On-line AE の COPA が提案されている。タグの長さは n-bit に固定されている。

安全性：[**?**] で示されている。

$$\mathtt{Adv}^{\mathtt{opriv}}_{\text{COPA}[E,n]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{oauth}}_{\text{COPA}[E,n]}(\mathcal{A}) \leq \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n}$$

2.6.1.10　Deterministic AE **方式**

■SIV　設計者: Rogaway と Shrimpton [48] により 2006 年に作られた。Deterministic AE(DAE) の最初の提案である。
構成：基本的には平文 M、およびヘッダ A が存在すれば A と M の連結、に対して MAC 関数を適用したのち、得られた出力 V をカウンターモードの初期値として用いて、平文を暗号化する。出力は V とカウンターモードの暗号化結果 C を連結した系列である。復号においては、V を用いて C を復号した後、復号結果 $\widetilde{M}$ を MAC 関数へ適用した結果が V と一致するかでメッセージ認証を行う。MAC 関数とカウンターモードの鍵は独立である。MAC 関数は並列実行可能で、vector-input (pseudorandom) function と呼ばれる形式を持つ、String-to-Vector (S2V) と呼ばれる関数である。これは、一つのバイナリ系列を vector として、vector の系列に対する PRF ととらえることができる[*3]。S2V

[*3] 原理上は Vector-input PRF は容易に単一の可変長入力 PRF と入力の符号化で構成可能であるが、S2V は vector に関するある種のインクリメンタル計算が可能という特徴を持つ。

は可変長（バイナリ系列）入力 PRF を部品として定義される。論文では CMAC を部品としている。レートは 2 であり、ブロック暗号の暗号化関数のみを利用する。タグ長は n bit 以下で設定可能である（が、安全性のバウンドは n の場合のみ扱っているとみられる；$\tau < n$ の場合は $q_v/2^\tau$ がバウンドに加算されると考えられる）。

安全性：[48] により示されている。なお、上述のように [48] では DAE security という単一の指標を中心に説明されているが、DPRIV と DAUTH の二つの指標で導出することが可能である（[48] の Proposition 9 を用いる）。

$$\mathtt{Adv}^{\mathtt{dae}}_{\mathrm{SIV}[E,n]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{dpriv}}_{\mathrm{SIV}[E,n]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_p^2}{2^n}$$
$$\mathtt{Adv}^{\mathtt{dauth}}_{\mathrm{SIV}[E,n]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n}$$

■HBS　設計者: 岩田と安田 [22] により 2009 年に作られた。

構成：SIV がブロック暗号の二つの鍵を用いて構成されているのに対し、HBS ではブロック暗号と Polynomial hashing とを組み合わせ、かつ一つのブロック暗号の鍵のみを用いる。Polynomial hashing の鍵の係数を調節して、ヘッダとメッセージを二つの vector とした vector-input 関数としている。大域的な構成は SIV と似ているが、復号でのタグの検証においてブロック暗号の復号関数を用いるため、全体の安全性はブロック暗号の強疑似ランダム性に帰着される。レートは 1 であり、追加としてヘッダ a ブロック、平文 m ブロックに対して $a+m+2$ 回の $\mathrm{GF}(2^{128})$ 乗算を要する。タグに対してブロック暗号復号関数を適用するため、タグ長は n bit に固定されている。

安全性：[22] により示されている。

$$\mathtt{Adv}^{\mathtt{dae}}_{\mathrm{HBS}[E,n]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{sprp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n}$$

■BTM　設計者: 岩田と安田 [21] により 2009 年に作られた。

構成：BTM は HBS におけるブロック暗号の復号関数の利用を無くすことを目的に開発された。Polynomial hash の利用などは HBS と同様である。結果として、全体の安全性はブロック暗号の疑似ランダム性に帰着される。レートは 1 であり、追加としてヘッダ a ブロック、平文 m ブロックに対して $a+m-1$ 回の $\mathrm{GF}(2^{128})$ 乗算を要する。タグに対してブロック暗号復号関数を適用しなくてよいため、タグ長は $\tau \le n$ bit に設定することが可能である。

安全性：[22] により示されている（タグ長 n bit のケースであるとみられる）。

$$\mathtt{Adv}^{\mathtt{dae}}_{\mathrm{BTM}[E,n]}(\mathcal{A}) \le \mathtt{Adv}^{\mathtt{prp}}_{E}(\mathcal{A}') + \frac{\sigma_a^2}{2^n}$$

2.6.1.11　その他

■軽量暗号技術との関連性　ブロック暗号を用いた認証暗号の軽量化に関しては、いくつかのアプローチがある。もっとも全体軽量化に貢献すると思われるのは軽量ブロック暗号を用いることであるが、その多くが 64-bit ブロックサイズであるために、ここで紹介した多くの方式が 32-bit データセキュリティ、すなわち、$2^{32} * 8$ byte $\approx$ 34 Gbyte よりも十分少ないデータ量を処理したところで、鍵の更新が必要となる。例えばカウンターモード（ないしカウンターモードを含んだ AE）を 64-bit ブロック暗号で運用した場合に、PRIV アドバンテージが $\sigma^2/2^{64}$ であるとして、これを 2^{-20} 以下におさえるにはおおよそ $2^{5.5}$ Mbyte のデータ処理の後に鍵更新が必要となる。これは帯域の制限されたセンサーネットなどでセッション鍵生成を頻繁に行う環境であれば実用的だが、一般的にはきわめて制約が強いと思われ

る。なお 128-bit ブロック暗号であれば PRIV を 2^{-20} におさえるのに鍵更新が必要となるデータ処理は $2^{28.5}$ GByte となり、一般的に十分と思われる。

一方、n-bit ブロック暗号で $n/2$-bit 以上のデータセキュリティを保証する AE としては岩田 [15, 14] の方式が知られるのみであり、またこれらの方式は比較的ブロック暗号の外側の処理としてオーバーヘッドが比較的大きい（例えば汎用の $GF(2^n)$ 乗算を有する点で）ため、軽量ブロック暗号のメリットを消してしまう懸念がある。

AE としての複雑さや処理のオーバーヘッドを下げる試みとしては前述の EAX-prime やその改良 [37] があげられる。これらは EAX と比べて、処理の前に必要なブロック暗号のコール回数や、処理中に保持すべきメモリ量を減らしている。またこれらの設計思想をさらに推し進めた CLOC, SILC もある。また、CCFB も安全性のバウンドに強い制約はあるものの、モードとしての処理のオーバーヘッドはかなり小さく、センサーネットでの実装に適することが知られている [24]。ただし、プラットフォームによっては用いるブロック暗号自体の影響が大きく、モードの選択は全体性能において大きな違いをもたらさない可能性もある。

また、一般にセンサーネットで重要とされる消費電力については、計算よりも通信部分の電力消費が大きい。Struik [52] により指摘されているように、組み込み環境で AE による保護を考えるときには、AE 適用による通信量の増分（IV とタグ）を考慮し、ここを小さくするように無駄のないプロトコルを設計することが重要であろう。この場合、IV なし、タグ無しなどの暗号化方式を適切なリスク分析のもと用いることも一つの手段である。

■想定する安全性モデルから逸脱した場合の影響　暗号化、およびメッセージ認証について、安全性のバウンドを超えたデータ量を処理した場合にどのような攻撃が起こりうるかはいくつかの論文で議論されている。例えば McGrew [29] は CTR, CFB, CBC の三つの基本的な暗号化のモードにおいて処理量がバースデーバウンドを超えた場合に、ほぼデータ量の対数に比例して線形に平文ビットが漏れることを示した。また、MAC の場合については Black と Cochran [8] が、一度偽造が成功した場合にその情報をもとにどのような偽造が可能となるかを様々な MAC について調査した。ここでの結果は、該当する MAC を用いた AE についても当てはまることが予想される。ただし AE についてこのような観点から網羅的に安全性評価を試みた研究は見つかっていない。このように安全性の保証を超えた使い方をした場合、いわゆるミスユースに対する安全性の議論は今後重要になるかもしれない。

バースデーバウンドとはやや異なるが、いわゆる弱鍵を利用した攻撃もいくつか提案されている。特に多項式ハッシュ（およびそれを用いている GCM）について数多く報告があり、Sarrinen による cycling attack [51], これを拡張した Procter と Cid [43] などの研究がある。鍵空間の部分集合 D について、D に鍵が入っているかを $|D|$ よりも少なくテストできるとき、D が弱鍵集合であるというのが従来の定義 [13] であったが、Procter, Cid はこの定義に従った場合、多項式ハッシュの鍵のほぼありとあらゆる部分集合が弱鍵集合とされてしまうことを示した。多項式ハッシュの脆弱性を指摘しているとも受け取れるが、証明可能安全性と矛盾するものではなく、ある意味では弱鍵集合の定義自体の意味を見直す必要があることも示唆している。

2.6.1.12　まとめ

認証暗号の安全性定義と、ブロック暗号に基づく具体的な方式とを調査した結果を報告した。5 章で述べたように、軽量な認証暗号を実現するために部品として軽量ブロック暗号を用いるだけでは解決できない課題がいくつかあり、またそれらの解決には認証暗号より上位のレイヤーでの解決が求められるケースもありそうである。また近年、ブロック暗号を用いず、ハッシュ関数やその部品をベースとする方式や、ブロック暗号のラウンド関数を部品として用いる方式などが提案されてきており、これらの動向にも注意が必要と思われる。

参考文献

[1] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. `http://competitions.cr.yp.to/caesar.html`.

[2] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Parallelizable and authenticated online ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8269 of *Lecture Notes in Computer Science*, pages 424–443. Springer, 2013.

[3] Kazumaro Aoki and Kan Yasuda. The Security of the OCB Mode of Operation without the SPRP Assumption. In Susilo and Reyhanitabar [53], pages 202–220.

[4] Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. Online Ciphers and the Hash-CBC Construction. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 292–309. Springer, 2001.

[5] Mihir Bellare, Alexandra Boldyreva, Lars R. Knudsen, and Chanathip Namprempre. On-line ciphers and the hash-cbc constructions. *J. Cryptology*, 25(4):640–679, 2012.

[6] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In Tatsuaki Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.

[7] Mihir Bellare, Phillip Rogaway, and David Wagner. The EAX Mode of Operation. In Roy and Meier [50], pages 389–407.

[8] John Black and Martin Cochran. MAC Reforgeability. In Dunkelman [10], pages 345–362.

[9] Anne Canteaut, editor. *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, volume 7549 of *Lecture Notes in Computer Science*. Springer, 2012.

[10] Orr Dunkelman, editor. *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *Lecture Notes in Computer Science*. Springer, 2009.

[11] Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In Canteaut [9], pages 196–215.

[12] Virgil D. Gligor and Pompiliu Donescu. Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes. In Mitsuru Matsui, editor, *FSE*, volume 2355 of *Lecture Notes in Computer Science*, pages 92–108. Springer, 2001.

[13] Helena Handschuh and Bart Preneel. Key-Recovery Attacks on Universal Hash Function Based MAC Algo-

rithms. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2008.

[14] Tetsu Iwata. New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In Matthew J. B. Robshaw, editor, *FSE*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2006.

[15] Tetsu Iwata. Authenticated Encryption Mode for Beyond the Birthday Bound Security. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2008.

[16] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, and Sumio Morioka. CLOC: Compact Low-Overhead CFB. `http://competitions.cr.yp.to/round1/clocv1.pdf`.

[17] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, and Sumio Morioka. CLOC: authenticated encryption for short input. *Proceedings of Fast Software Encryption 2014*, 2014:157, 2014.

[18] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi. SILC: SImple Lightweight CFB. `http://competitions.cr.yp.to/round1/silcv1.pdf/`.

[19] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and Eita Kobayashi. SILC: SImple Lightweight CFB. DIAC: Directions in Authenticated Ciphers, 2014. `http://2014.diac.cr.yp.to/`.

[20] Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and Repairing GCM Security Proofs. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 31–49. Springer, 2012.

[21] Tetsu Iwata and Kan Yasuda. BTM: A Single-Key, Inverse-Cipher-Free Mode for Deterministic Authenticated Encryption. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 313–330. Springer, 2009.

[22] Tetsu Iwata and Kan Yasuda. HBS: A Single-Key Mode of Operation for Deterministic Authenticated Encryption. In Dunkelman [10], pages 394–415.

[23] Jakob Jonsson. On the Security of CTR + CBC-MAC. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 76–93. Springer, 2002.

[24] Marcos A. Simplício Jr., Bruno Trevizan de Oliveira, Paulo S. L. M. Barreto, Cintia B. Margi, Tereza Cristina M. B. Carvalho, and Mats Näslund. Comparison of Authenticated-Encryption schemes in Wireless Sensor Networks. In Chun Tung Chou, Tom Pfeifer, and Anura P. Jayasumana, editors, *IEEE 36th Conference on Local Computer Networks, LCN 2011, Bonn, Germany, October 4-7, 2011*, pages 450–457. IEEE, 2011.

[25] Charanjit S. Jutla. Encryption Modes with Almost Free Message Integrity. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 529–544. Springer, 2001.

[26] Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In Antoine Joux, editor, *FSE*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2011.

[27] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *J. Cryptology*, 24(3):588–613, 2011.

[28] Stefan Lucks. Two-Pass Authenticated Encryption Faster Than Generic Composition. In Henri Gilbert and Helena Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 284–298. Springer, 2005.

[29] David A. McGrew. Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. *Pre-proceedings of Fast Software Encryption 2013*. Available from `http://eprint.iacr.org/2012/623`.

[30] David A. McGrew and John Viega. The Galois/Counter mode of operation (GCM). NIST Submission, 2004. Available from `http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html`.

[31] David A. McGrew and John Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In Anne Canteaut and Kapalee Viswanathan, editors, *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

[32] Florian Mendel, Bart Mennink, Vincent Rijmen, and Elmar Tischhauser. A Simple Key-Recovery Attack on McOE-X. In Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis, editors, *CANS*, volume 7712, pages 23–31. Springer, 2012.

[33] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.

[34] Kazuhiko Minematsu. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. *IACR Cryptology ePrint Archive*, 2013:628, 2013.

[35] Kazuhiko Minematsu. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. In Nguyen and Oswald [42], pages 275–292.

[36] Kazuhiko Minematsu, Stefan Lucks, and Tetsu Iwata. Improved Authenticity Bound of EAX, and Refinements. In Susilo and Reyhanitabar [53], pages 184–201.

[37] Kazuhiko Minematsu, Stefan Lucks, Hiraku Morita, and Tetsu Iwata. Attacks and security proofs of EAX-prime. In Moriai [40], pages 327–347.

[38] Chris J. Mitchell. Cryptanalysis of Two Variants of PCBC Mode When Used for Message Integrity. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP*, volume 3574 of *Lecture Notes in Computer Science*, pages 560–571. Springer, 2005.

[39] Chris J. Mitchell. Analysing the IOBC Authenticated Encryption Mode. In Colin Boyd and Leonie Simpson, editors, *ACISP*, volume 7959 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013.

[40] Shiho Moriai, editor. *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume 8424 of *Lecture Notes in Computer Science*. Springer, 2014.

[41] Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering Generic Composition. In Nguyen and Oswald [42], pages 257–274.

[42] Phong Q. Nguyen and Elisabeth Oswald, editors. *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*. Springer, 2014.

[43] Gordon Procter and Carlos Cid. On Weak Keys and Forgery Attacks Against Polynomial-Based MAC Schemes. In Moriai [40], pages 287–304.

[44] Francisco Recacha. Input and Output Chaining. NIST Submission, 2013. Available from `http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html`.

[45] Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.

[46] Phillip Rogaway. Nonce-Based Symmetric Encryption. In Roy and Meier [50], pages 348–359.

[47] Phillip Rogaway, Mihir Bellare, and John Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, 2003.

[48] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.

[49] Phillip Rogaway and Haibin Zhang. Online Ciphers from Tweakable Blockciphers. In Aggelos Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 237–249. Springer, 2011.

[50] Bimal K. Roy and Willi Meier, editors. *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*. Springer, 2004.

[51] Markku-Juhani Olavi Saarinen. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. In Canteaut [9], pages 216–225.

[52] Rene Struik. Revisiting design criteria for AEAD ciphers targeting highly constrained networks. DIAC: Directions in Authenticated Ciphers, 2013. `http://2013.diac.cr.yp.to/`.

[53] Willy Susilo and Reza Reyhanitabar, editors. *Provable Security - 7th International Conference, ProvSec 2013, Melaka, Malaysia, October 23-25, 2013. Proceedings*, volume 8209 of *Lecture Notes in Computer Science*. Springer, 2013.

[54] Douglas Whiting, Russ Housley, and Niels Ferguson. Counter with CBC-MAC (CCM). NIST Submission, 2002. Available from `http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html`.

[55] Liting Zhang, Sui Han, Wenling Wu, and Peng Wang. iFeed: the Input-Feed AE Modes. Rump Session of FSE 2013, 2013. slides from `http://fse.2013.rump.cr.yp.to/`.

[56] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. iFeed[AES] v1. `http://competitions.cr.yp.to/round1/ifeedaesv1.pdf`.

2.6.2 認証暗号の実装性能

本章では、軽量暗号技術の現状調査として、主要な認証暗号の実装性能 (ハードウェア、ソフトウェア) 調査結果をまとめる。

2.6.2.1 調査内容

■Grain-128a　Grain-128 は 2004 年に eSTREAM のハードウェア部門に提案されたアルゴリズムであり、eSTREAM の Winner の一つである。文献 [1] に示されるハードウェア性能を表 2.22 にまとめる。文献 [1] ではゲートカウントの見積もりのみを実施している。

表 2.22　Grain-128a のゲートカウント見積もり

機能	速度モード毎のゲートカウント [gate]					
	1×	2×	4×	8×	16×	32×
暗号化のみ	2145.5	2243	2438	2828	3608	5168
32bitMAC 付き暗号化	2769.5	2867	3174	3788	5016	7472

■ALE　ALE は FSE2013 で Rijmen らによって提案されたアルゴリズムである。AES-NI を積極的に利用することが可能な設計が採られている。デディケイトの設計ではあるが、モードの設計にも近く、性能比較も AES のモードとの比較をしている。文献 [2] に示される AES の Serial 実装（S-box 1 つを使いまわして暗号化演算を行う HW 実装）をベースにした 65nmCMOS スタンダードセルライブラリによる実装評価結果を表 2.23 に纏める。

表 2.23　ALE の回路性能

Design	Area[gate]	Clock cycles / block	Overhead cycles / message	Power [uW]
AES-ECB	2,435	226	-	87.84
AES-OCB2	4,612	226	452	171.23
AES-OCB2 e/d	5,916	226	452	211.01
ASC-1 A	4,793	370	904	169.11
ASC-1 A e/d	4,964	370	904	193.71
ASC-1 B	5,517	235	904	199.02
ASC-1 B e/d	5,632	235	904	207.13
AES-CCM	3,472	452	-	128.31
AES-CCM e/d	3,765	452	-	162.15
ALE	2,579	105	678	94.87
ALE e/d	2,700	105	678	102.32

ここで、ASC-1 は文献 [2] で示されるアルゴリズムであり、ALE の原型と呼べるアルゴリズムである。ALE は AES-OCB2 に対して半分の回路規模で 2 倍の処理速度が得られる。

図 2.2 に文献 [2] に記載される Sandy Bridge (AES-NI) 利用時のソフトウェア性能を示す。図から ALE は AES-OCB3 と同程度の処理性能を持つことがわかる。

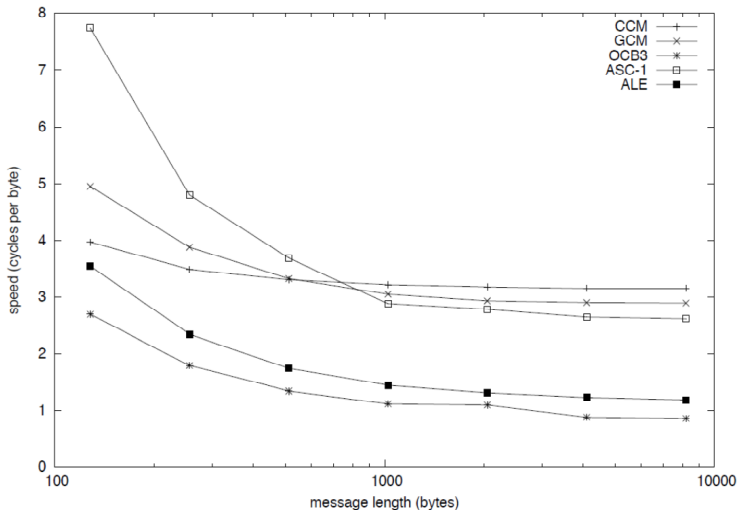


図 2.2　ALE のソフトウェア性能

■FIDES　FIDES は CHES2013 で提案された認証暗号であり、以下のような特徴を持つ。

・ 論理回路として 793 gate で実装可能
・ Sponge 構造で 5bit、 6bit S-box を持つ。
・ 鍵長、ステートが 80bit、160bit と 96bit、192bit の 2 種類ある。

表 2.24 に文献 [3] に記載されるハードウェア性能を示す。文献 [3] では 3 種類の CMOS プロセスを用いた評価結果を示している。表中、Threshold implementation とはハードウェアにおけるサイドチャネル対策のための一方式を指す。

■Phelix　Phelix は 2004 年に eSTREAM に提案されたＭＡＣ付きストリーム暗号である。Phase 2 で落選で落選している。表 2.25 に文献 [4] に記載されるソフトウェア性能を示す。文献 [4] では Pentium M CPU での速度性能が示されている。

■Mode of Operation　AES-NI 前提で CCM、GCM、OCB3 など認証暗号用のモードに対する速度性能評価が文献 [5, 6, 7, 8] でなどで実施されている。表 2.26 にそれぞれの評価結果をまとめる。

■CAESAR **プロジェクト提案暗号**　認証暗号アルゴリズムの公募プロジェクトである CAESAR プロジェクトへ提案されているアルゴリズムについて、提案者ら提示している実装性能を以下に示す。尚、既に鍵の全数探索よりも効率のよい攻撃方法が見つかっているアルゴリズムを含め、まとめている点に留意されたい。

表 2.27 に、FPGA の性能評価結果をまとめる。3 つのアルゴリズムで性能値が示されている。

表 2.28 に、ASIC の性能評価結果をまとめる。5 つのアルゴリズムで性能値が示されている。表 2.29 は、具体的な実装結果ではないが、ゲート規模の見積もりなどを実施しているアルゴリズムの性能値をまとめた結果である。5 つのアルゴリズムで性能値が示されている。

表 2.24　FIDES のハードウェア性能

Design	Security (bits)	Area (GE)	Frequency (kHz)	Latency	Throughput (kb/s)	Power (μW)
	Advanced NXP 90 nm CMOS process, typical PVT (25 °C, 1.2 V)					
FIDES-80-S	80	793	100	47	10.64	N/A
FIDES-80-4S	80	1178	100	23	21.74	N/A
FIDES-80-R	80	2922	100	1	500.00	N/A
FIDES-80-T	80	2876	100	47	10.64	N/A
FIDES-96-S	96	1001	100	47	12.77	N/A
FIDES-96-4S	96	1305	100	23	26.09	N/A
FIDES-96-R	96	6673	100	1	600.00	N/A
FIDES-96-T	96	4792	100	47	12.77	N/A
	NANGATE 45 nm CMOS process, typical PVT (25 °C, 1.1 V)					
FIDES-80-S	80	1244	100	47	10.64	N/A
FIDES-80-4S	80	1819	100	23	21.74	N/A
FIDES-80-R	80	4023	100	1	500.00	N/A
FIDES-80-T	80	4696	100	47	10.64	N/A
FIDES-96-S	96	1584	100	47	12.77	N/A
FIDES-96-4S	96	2023	100	23	26.09	N/A
FIDES-96-R	96	9180	100	1	600.00	N/A
FIDES-96-T	96	7541	100	47	12.77	N/A
	UMC 130 nm CMOS process, typical PVT (25 °C, 1.2 V)					
FIDES-80-S	80	1153	100	47	10.64	1.97
FIDES-80-4S	80	1682	100	23	21.74	2.82
FIDES-80-R	80	4175	100	1	500.00	7.90
FIDES-80-T	80	4267	100	47	10.64	7.47
FIDES-96-S	96	1453	100	47	12.77	2.49
FIDES-96-4S	96	1870	100	23	26.09	3.12
FIDES-96-R	96	8340	100	1	600.00	14.82
FIDES-96-T	96	6812	100	47	12.77	11.84

Fides-xy-S : Serial architecture (1 S-box).

Fides-xy-4S : Architecture with 4 S-boxes.

Fides-xy-R : Round-based architecture (32 S-boxes).

Fides-xy-T : Threshold implementation (1 S-box).

表 2.25　Phelix のソフトウェア性能

Operation	Version	Packet Size (N)			Approximate Equation (clks)
		64 bytes	256 bytes	1024 bytes	
Encrypt	C	41.6 cpb	20.3 cpb	15.0 cpb	$1810 + 13.2N$
Decrypt	C	42.3 cpb	21.1 cpb	15.8 cpb	$1610 + 14.0N$
Encrypt	ASM	18.5 cpb	9.8 cpb	7.4 cpb	$810 + 6.6N$
Decrypt	ASM	18.2 cpb	9.6 cpb	7.4 cpb	$750 + 6.7N$

cbp: clocks per byte

*1 1 round per cycle

*2 2 rounds per cycle

表 2.26　暗号利用モードのソフトウェア性能 (Sandy Bridge)

Mode	cbp	data	source
ECB	0.702	4KB	[6]
	0.853	8KB	OpenSSL 1.0.1c
CTR	0.691	4KB	[6]
	0.79	16KB	[RWC2013]
	0.916	8KB	OpenSSL 1.0.1c
OCB2	1.016	4KB	[6] (連続 2 倍)
	1.350	4KB	[6] (通常 2 倍)
OCB3	0.818	4KB	[6]
	0.87	4KB	[9]
GCM	2.47	16KB	[7]
	2.53	4KB	[7]
	2.564	4KB	[6]
	2.899	8KB	OpenSSL 1.0.1c

Algorithm	Platform	Area	Freq. (MHz)	Throughput (Mbps)	Source
ICEPOLE	Xilinx Virtex6	1501 (Slices/ALUT)	N/A	41,364	[21]
	Altera Stratix IV	4564 (Slices/ALUT)	N/A	38,779	[21]
KIASU-BC	Xilinx Virtex5	1989 (slices)	N/A	1,080	[24]
pi-Cipher	Xilinx Virtex6	41 (slices)	N/A	N/A	[32]

表 2.27　CAESAR 候補の FPGA 性能 (実装値)

表 2.30 に、Mode-of-operation の提案で、Ivy Bridge マイクロアーキテクチャを実装ターゲットとしたソフトウェアの性能評価結果をまとめる。3 つのアルゴリズムで性能値が示されている。同様に表 2.31 には Dedicated としての提案に対する性能値をまとめる。

Algorithm	Area	Freq. (MHz)	Throughput (Mbps)	Source
CLOC	17137.75 (GE)	100	685.71	[17]
Minalpher-P	2810 (GE)			
NORX	62000 (GE)	125	10240	[30]
SCREAM-10 (Enc-/Dec-only)*1	12,951 (μm^2)	751	4577	[36]
SCREAM-10 (Enc-/Dec-only)*2	17,292 (μm^2)	446	5190	[36]
SCREAM-10 (Enc+Dec)*1	17,292 (μm^2)	751	4577	[36]
SCREAM-10 (Enc+Dec)*2	25,974 (μm^2)	446	5190	[36]
iSCREAM-12 (Enc-/Dec-only)*1	13,375 (μm^2)	740	3789	[36]
iSCREAM-12 (Enc-/Dec-only)*2	17,024 (μm^2)	448	4411	[36]
iSCREAM-12 (Enc+Dec)*1	13,375 (μm^2)	740	3789	[36]
iSCREAM-12 (Enc+Dec)*2	17,024 (μm^2)	448	4411	[36]
SILC	15675.5 (GE)	100	764.12	[38]

表 2.28　CAESAR 候補の ASIC 性能 (実装値)

Algorithm	Area (GE)	Source
Deoxys-BC-128-128	3400	[18]
Deoxys-BC-256-128	4400	[18]
Deoxys-128-128	4600	[18]
Deoxys-128-128	5600	[18]
Joltik≠-64-64	2100	[19]
Joltik≠-80-48	2100	[19]
Joltik≠-96-96	2600	[19]
Joltik≠-128-64	2600	[19]
Joltik=-64-64	2600	[19]
Joltik=-80-48	2600	[19]
Joltik=-96-96	3100	[19]
Joltik=-128-64	3100	[19]
KIASU≠	4000	[23]
KIASU=	5000	[23]
LAC	1300	[25]
Sablier	1925	[35]

表 2.29　CAESAR 候補の ASIC 性能 (概算値)

Algorithm	Speed (cpb)	Message length (bytes)	Source
AES-CPFB(Enc)	2	1500	[13]
	1.47	32768	[13]
AES-CPFB(Dec)	7.5	1500+	[13]
AES-SILC	4.9	long	[38]
PRESENT-SILC	42	long	[38]
LED-SILC	40	long	[38]
Scream-10	7.1	long	[36]
iScream-12	9.1	long	[36]

表 2.30　CAESAR 候補 (Mode of operation) のソフトウェア性能 (Ivy Bridge)

Algorithm	Speed (cpb)	Message length (bytes)	Source
ICEPOLE (without special instruction sets)	9	N/A	[22]
Minalpher	23.1	31	[27]
	14.4	8192	[27]
	14.4	65536	[27]

表 2.31　CAESAR 候補 (Dedicated) のソフトウェア性能 (Ivy Bridge)

表 2.32 に、Mode-of-operation の提案で、Sandy Bridge マイクロアーキテクチャを実装ターゲットとしたソフトウェアの性能評価結果をまとめる。5 つのアルゴリズムで性能値が示されている。同様に表 2.33 には Dedicated としての提案に対する性能値をまとめる。

Algorithm	Speed (cpb)	Message length (bytes)	Source
AES-JAMBU	17.7	64	[14]
	14.54	128	[14]
	13.06	256	[14]
	12.27	512	[14]
	11.86	1024	[14]
	11.60	4096	[14]
AEGIS-128L(Enc/Dec)	3.68/3.81	64	[11]
	2.05/2.12	128	[11]
	1.23/1.27	256	[11]
	0.83/0.85	512	[11]
	0.63/0.63	1024	[11]
	0.48/0.48	4096	[11]
AEGIS-128(Enc/Dec)	3.37/3.78	64	[11]
	1.99/2.17	128	[11]
	1.30/1.36	256	[11]
	0.96/1.02	512	[11]
	0.80/0.84	1024	[11]
	0.66/0.67	4096	[11]
AEGIS-256(Enc/Dec)	3.51/4.00	64	[11]
	2.10/2.35	128	[11]
	1.34/1.51	256	[11]
	1.03/1.09	512	[11]
	0.86/0.90	1024	[11]
	0.70/0.74	4096	[11]
Deoxys$^{\neq}$-128-128	2.30	128	[18]
	1.73	256	[18]
	1.45	512	[18]
	1.36	1024	[18]
	1.15	2048	[18]
	1.13	4096	[18]
Deoxys$^{\neq}$-256-128	4.26	128	[18]
	2.53	256	[18]
	1.92	512	[18]
	1.57	1024	[18]

Algorithm	Speed (cpb)	Message length (bytes)	Source
	1.48	2048	[18]
	1.32	4096	[18]
Deoxys$^{=}$-128-128	4.50	128	[18]
	3.42	256	[18]
	2.84	512	[18]
	2.61	1024	[18]
	2.43	2048	[18]
	2.33	4096	[18]
Deoxys$^{=}$-256-128	7.89	128	[18]
	5.13	256	[18]
	3.55	512	[18]
	3.07	1024	[18]
	2.75	2048	[18]
	2.59	4096	[18]
KIASU$\neq$	1.02	4096	[23]
KIASU$=$	1.98	4096	[23]
Tiaoxin	2.49	128	[41]
	1.45	256	[41]
	0.91	512	[41]
	0.65	1024	[41]
	0.50	2048	[41]
	0.44	4096	[41]
	0.40	8192	[41]
	0.38	2^16	[41]

表 2.32: CAESAR 候補 (Mode of operation) のソフトウェア性能 (Sandy Bridge)

表 2.34 に、Mode-of-operation の提案で、Haswell マイクロアーキテクチャを実装ターゲットとしたソフトウェアの性能評価結果をまとめる。9 つのアルゴリズムで性能値が示されている。同様に表 2.35 には Dedicated としての提案に対する性能値をまとめる。

Algorithm	Speed (cpb)	Message length (bytes)	Source
AES-COPA	1.44	128(short)	[12]
	1.29	2048(long)	[12]
AEZ	0.38(検証失敗時)	1500	[15]
	0.89	1500	[15]

Algorithm	Speed (cpb)	Message length (bytes)	Source
	0.72	16384	[15]
AEGIS-128L(Enc/Dec)	3.44/3.45	64	[11]
	1.88/1.88	128	[11]
	1.11/1.09	256	[11]
	0.71/0.70	512	[11]
	0.51/0.50	1024	[11]
	0.37/0.35	4096	[11]
AEGIS-128(Enc/Dec)	3.29/2.98	64	[11]
	1.92/1.77	128	[11]
	1.24/1.16	256	[11]
	0.91/0.86	512	[11]
	0.73/0.81	1024	[11]
	0.61/0.60	4096	[11]
AEGIS-256(Enc/Dec)	3.98/3.88	64	[11]
	2.28/2.22	128	[11]
	1.42/1.39	256	[11]
	0.99/0.98	512	[11]
	0.78/0.77	1024	[11]
	0.62/0.62	4096	[11]
Deoxys$^{\neq}$-128-128	2.25	128	[18]
	1.84	256	[18]
	1.64	512	[18]
	1.55	1024	[18]
	1.49	2048	[18]
	1.46	4096	[18]
Deoxys$^{\neq}$-256-128	3.68	128	[18]
	2.66	256	[18]
	2.14	512	[18]
	1.88	1024	[18]
	1.76	2048	[18]
	1.69	4096	[18]
Deoxys$^{=}$-128-128	4.07	128	[18]
	3.43	256	[18]
	3.12	512	[18]
	2.97	1024	[18]
	2.89	2048	[18]
	2.85	4096	[18]
Deoxys$^{=}$-256-128	5.68	128	[18]

Algorithm	Speed (cpb)	Message length (bytes)	Source
	4.44	256	[18]
	3.82	512	[18]
	3.51	1024	[18]
	3.36	2048	[18]
	3.28	4096	[18]
HS1-SIV	0.8	N/A	[20]
KIASU≠	0.74	4096	[23]
KIASU=	1.39	4096	[23]
Marble	1.6	8192	[26]
Silver(Enc/Dec)(AES-NI)	10.8/9.6	44	[39]
	1/1.2	1536	[39]
	0.73/0.81	long	[39]
Silver(Enc/Dec)(non-AES-NI)	30.4/28.2	44	[39]
	11.85/13.59	1536	[39]
	11.45/12.9	long	[39]
Tiaoxin	0.31	8192	[41]
	0.28	long	[41]

表 2.34: CAESAR 候補 (Mode of operation) のソフトウェア性能 (Haswell)

最後に表 2.36 として、上記までででは分類に含まれないターゲットに対するソフトウェアの性能評価結果をまとめる。

Algorithm	Platform	ROM/RAM (bytes)	Speed (cpb)	Message length (bytes)	Source
HS1-SIV	MIPS32	N/A	16	N/A	[20]
	Cortex-A9	N/A	5	N/A	[20]
LAC	Core i7-3612QM	N/A	720	12	[25]
			589	16	[25]
			440	32	[25]
			256	64	[25]
			206	128	[25]
			174	256	[25]
			152	512	[25]
			144	1024	[25]

[*3] Ref: 移植可能な C レファレンス実装、AVX2: AVX2 利用の最適実装

			140	2048	[25]
			138	4096	[25]
Minalpher	RL78	1275/470	514	long	[27]
NORX32-6-1 (Ref/NEON)*4	Samsung Exynos 4412 Prime (Cortex-A9)	N/A	794.12/541.00	8	[30]
			128.66/77.78	64	[30]
			42.14/22.79	576	[30]
			35.45/18.36	1536	[30]
			32.35/16.70	4096	[30]
			31.56/15.66	long	[30]
NORX32-4-1 (Ref/NEON)*4	Samsung Exynos 4412 Prime (Cortex-A9)	N/A	663.75/434.88	8	[30]
			97.94/61.73	64	[30]
			30.50/16.40	576	[30]
			24.94/12.77	1536	[30]
			22.86/11.41	4096	[30]
			21.57/10.57	long	[30]
NORX64-6-1 (Ref/AVX)*5	Core i7-2630QM	N/A	304.00/198.00	8	[30]
			37.75/24.81	64	[30]
			11.54/7.52	576	[30]
			9.08/5.90	1536	[30]
			8.14/5.24	4096	[30]
			7.69/4.94	long	[30]
NORX64-4-1 (Ref/AVX)*5	Core i7-2630QM	N/A	208.00/133.50	8	[30]
			26.00/16.69	64	[30]
			7.94/5.03	576	[30]
			6.24/3.91	1536	[30]
			5.59/3.49	4096	[30]
			5.28/3.28	long	[30]
NORX64-6-1 (Ref/AVX)*5	Core i7-3667U	N/A	371.50/276.00	8	[30]
			34.87/25.44	64	[30]
			10.59/7.71	576	[30]
			8.32/6.03	1536	[30]
			7.46/5.37	4096	[30]
			7.04/5.04	lonog	[30]
NORX64-4-1 (Ref/AVX)*5	Core i7-3667U	N/A	310.00/218.00	8	[30]
			24.93/17.18	64	[30]
			7.43/5.16	576	[30]
			5.86/4.01	1536	[30]
			5.24/3.59	4096	[30]
			4.92/3.37	long	[30]

POET	Core i5-4300U	N/A	4.61	128	[34]
			4.24	256	[34]
			4.13	512	[34]
			4.02	1024	[34]
			3.92	2048	[34]
OMD-SHA256	Core i5-2415M	N/A	44.56	128	[31]
			28.77	4096	[31]
OMD-SHA512	Core i5-2415M	N/A	45.93	128	[31]
			23.28	4096	[31]
Scream-10[*4]	Cortex A15	N/A	21.8	long	[36]
	Atom Cedarview	N/A	55	long	[36]
	Core i7 Nehalem	N/A	9.3	long	[36]
	Atmel AVR	3221/80	7646(E)/7672(D)	N/A	[36]
	Atmel AVR	1723/80(Enc-only)	7646	N/A	[36]
	Atmel AVR	1751/80(Dec-only)	7672	N/A	[36]
iScream-12[*5]	Cortex A15	N/A	26.2	long	[36]
	Atom Cedarview	N/A	65	long	[36]
	Core i7 Nehalem	N/A	11.2	long	[36]
	Atmel AVR	1975/64	8724(E)/8724(D)	long	[36]
	Atmel AVR	1595/64(Enc-only)	8724	N/A	[36]
	Atmel AVR	1593/64(Dec-only)	8724	N/A	[36]
STRIBOB	Core i7 860	N/A	25.3	N/A	[40]

表 2.36: CAESAR 候補のソフトウェア性能 (Others)

2.6.2.2 **まとめ**

本節では、主要な認証暗号の実装性能 (ハードウェア、ソフトウェア) 調査結果をまとめた。本調査は CAESAR プロジェクトがスタートし、第二ラウンド進出アルゴリズムを選定している段階で実施しているため、数多くのアルゴリズムについて性能値を掲載している。しかしながら、これらはあくまで著者らの主張に基づいた提示であり、本資料記載のデータを用いてアルゴリズム間の比較を行う目的にはそぐわないことに注意されたい。

今後、安全性やサイドチャネル対策との関連性を含めプロジェクトでの絞り込みについて動向を注視していく必要があると考える。

[*4] only the tweakable block sipher is implemented

[*5] only the tweakable block sipher is implemented

[*4] Ref: 移植可能な C レファレンス実装、NEON: NEON 利用の最適実装

[*5] Ref: 移植可能な C レファレンス実装、AVX: AVX 利用の最適実装

Algorithm	Speed (cpb)	Message length (bytes)	Source
ACORN	72.1	64	[10]
	41.5	128	[10]
	26.3	256	[10]
	18.6	512	[10]
	14.7	1024	[10]
	12.8	2048	[10]
	11.9	4096	[10]

表 2.33　CAESAR 候補 (Dedicated) のソフトウェア性能 (Sandy Bridge)

Algorithm	Speed (cpb)	Message length (bytes)	Source
ICEPOLE(without special instruction sets)	8	N/A	[22]
Minalpher	5.76	8192	[28]
MORUS-640(Enc/Dec)	7.72/7.99	64	[29]
	1.18/1.23	4096	[29]
	1.11/1.16	long	[29]
MORUS-1280(Enc/Dec)	8.28/8.46	64	[29]
	0.78/0.80	4096	[29]
	0.69/0.69	long	[29]
NORX64-6-1(Ref/AVX2)*3	1248.00/748.24	8	[30]
	156.61/93.23	64	[30]
	9.85/5.71	576	[30]
	7.77/4.47	1536	[30]
	7.00/3.98	4096	[30]
	6.63/3.73	long	[30]
NORX64-4-1(Ref/AVX2)*3	863.12/509.51	8	[30]
	106.94/63.38	64	[30]
	6.71/3.83	576	[30]
	5.27/3.01	1536	[30]
	4.76/2.66	4096	[30]
	4.50/2.51	long	[30]

表 2.35　CAESAR 候補 (Dedicate) のソフトウェア性能 (Haswell)

参考文献

[1] M. Ågren, M. Hell, T. Johansson and W. Meier:
Grain-128a: a new version of Grain-128 with optional authentication.
IJWMC 5(1): 48-59, 2011.

[2] A. Bogdanov, F. Mendel, F. Regazzoni, E. Tischhauser, and V. Rijmen:
ALE: AES-Based Lightweight Authenticated Encryption. FSE2013.

[3] B. Bilgin, A. Bogdanov, M. Knezevic, F. Mendel and Q. Wang:
FIDES: Lightweight Authenticated Cipher with Side-Channel Resistance for Constrained Hardware. CHES2013.

[4] D. Whiting, B. Schneier, S. Lucks and F. Muller: Phelix Fast Encryption and Authentication in a Single Cryptographic Primitive.
https://www.schneier.com/paper-phelix.pdf

[5] K. Aoki, T. Iwata and K. Yasuda: How Fast Can a Two-Pass Mode Go? A Parallel Deterministic Authenticated Encryption Mode for AES-NI.
DIAC 2012

[6] K. Aoki: Optimization of mode implementations on Sandy Bridge.
SCIS 2013

[7] S. Gueron: AES-GCM for Efficient Authenticated Encryption - Ending the Reign of HMAC-SHA-1?
https://crypto.stanford.edu/RealWorldCrypto/slides/gueron.pdf

[8] S. Gueron: AES-GCM software performance on the current high end CPUs as a performance baseline for CAESAR competition?
http://2013.diac.cr.yp.to/slides/gueron.pdf

[9] P. Rogaway, M. Bellare, J. Black, T. Krovetz, and T. Shrimpton: The Evolution of Authenticated Encryption.
http://hyperelliptic.org/DIAC/slides/sweden-rogaway-ae-2012b.pdf

[10] Hongjun Wu, "ACORN: A Lightweight Authenticated Cipher (v1),"
http://competitions.cr.yp.to/round1/acornv1.pdf

[11] Hongjun Wu, Bart Preneel, "AEGIS: A Fast Authenticated Encryption Algorithm (v1),"
http://competitions.cr.yp.to/round1/aegisv1.pdf

[12] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, Kan Yasuda, "AES-COPA v.1,"
http://competitions.cr.yp.to/round1/aescopav1.pdf

[13] Miguel Montes, Daniel Penazzi, "AES-CPFB v1,"

http://competitions.cr.yp.to/round1/aescpfbv1.pdf

[14] Hongjun Wu, Tao Huang, "JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU (v1)," http://competitions.cr.yp.to/round1/aesjambuv1.pdf

[15] Viet Tung Hoang, Ted Krovetz, Phillip Rogaway, "AEZ v3: Authenticated Encryption by Enciphering," http://web.cs.ucdavis.edu/~rogaway/aez/aez.pdf

[16] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, "CLOC: Compact Low-Overhead CFB," http://competitions.cr.yp.to/round1/clocv1.pdf

[17] Iwata, "CAESAR candidate SILC," http://2014.diac.cr.yp.to/slides/iwata-silc.pdf

[18] Jérémy Jean, Ivica Nikolić, Thomas Peyrin, "Deoxys v1," http://competitions.cr.yp.to/round1/deoxysv1.pdf

[19] Jérémy Jean, Ivica Nikolić, Thomas Peyrin, "Joltik v1," http://competitions.cr.yp.to/round1/joltikv1.pdf

[20] Ted Krovetz, "HS1-SIV," http://2014.diac.cr.yp.to/slides/krovetz-hs1.pdf

[21] Pawel Morawiecki, Kris Gaj, Ekawat Homsirikamol, Krystian Matusiewicz, Josef Pieprzyk, Marcin Rogawski, Marian Srebrny, Marcin Wójcik, "ICEPOLE v1," http://competitions.cr.yp.to/round1/icepolev1.pdf

[22] Rogawski, "CAESAR candidate ICEPOLE", http://2014.diac.cr.yp.to/slides/rogawski-icepole.pdf

[23] Jérémy Jean, Ivica Nikolić, Thomas Peyrin, "KIASU v1," http://competitions.cr.yp.to/round1/kiasuv1.pdf

[24] Peyrin, "CAESAR candidate KIASU," http://competitions.cr.yp.to/round1/kiasuv1.pdf

[25] Lei Zhang, Wenling Wu, Yanfeng Wang, Shengbao Wu, Jian Zhang, "LAC: A Lightweight Authenticated Encryption Cipher," http://competitions.cr.yp.to/round1/lacv1.pdf

[26] Jian Guo, "Marble Specification Version 1.1," http://competitions.cr.yp.to/round1/marblev11.pdf

[27] Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, Shoichi Hirose, "Minalpher v1," http://competitions.cr.yp.to/round1/minalpherv1.pdf

[28] Kazumaro Aoki, "Observations on Prøst and Minalpher," https://www.cryptolux.org/mediawiki-esc2015/images/c/cb/Slide.pdf

[29] Hongjun Wu, Tao Huang, "The Authenticated Cipher MORUS (v1)," http://competitions.cr.yp.to/round1/morusv1.pdf

[30] Jean-Philippe Aumasson, Philipp Jovanovic, Samuel Neves, "NORX v1," http://competitions.cr.yp.to/round1/norxv1.pdf

[31] Simon Cogliani, Diana-Ştefania Maimuţ, David Naccache, Rodrigo Portella do Canto, Reza Reyhanitabar, Serge Vaudenay, Damian Vizár, "OMD A Compression Function Mode of Operation for Authenticated

Encryption,"
http://2014.diac.cr.yp.to/slides/reyhanitabar-omd.pdf

[32] Danilo Gligoroski, Hristina Mihajloska, Simona Samardjiska, Håkon Jacobsen, Mohamed El-Hadedy, Rune Erlend Jensen, "π-Cipher v1,"
http://competitions.cr.yp.to/round1/picipherv1.pdf

[33] Gligoroski, "CAESAR candidate PiCipher," http://2014.diac.cr.yp.to/slides/gligoroski-picipher.pdf

[34] Farzaneh Abed, Scott Fluhrer, John Foley, Christian Forler, Eik List, Stefan Lucks, David McGrew, Jakob Wenzel, "The POET Family of On-Line Authenticated Encryption Schemes,"
http://competitions.cr.yp.to/round1/poetv101.pdf

[35] Bin Zhang, Zhenqing Shi, Chao Xu, Yuan Yao, Zhenqi Li, "Sablier v1,"
http://competitions.cr.yp.to/round1/sablierv1.pdf

[36] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, François Durvaux, Lubos Gaspar, Stéphanie Kerckhof, "SCREAM & iSCREAM Side-Channel Resistant Authenticated Encryption with Masking,"
http://competitions.cr.yp.to/round1/screamv1.pdf

[37] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, Eita Kobayashi, "SILC: SImple Lightweight CFB,"
http://competitions.cr.yp.to/round1/silcv1.pdf

[38] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, Eita Kobayashi, "SILC: SImple Lightweight CFB,"
http://2014.diac.cr.yp.to/slides/iwata-silc.pdf

[39] Daniel Penazzi, Miguel Montes, "Silver and AESCPFB,"
http://2014.diac.cr.yp.to/slides/penazzi-silver-cpfb.pdf

[40] Markku-Juhani O. Saarinen, "The STRIBOBr1 Authenticated Encryption Algorithm,"
http://competitions.cr.yp.to/round1/stribobr1.pdf

[41] Ivica Nikolić, "Tiaoxin–346,"
http://competitions.cr.yp.to/round1/tiaoxinv1.pdf

第 3 章

軽量暗号に関する現状調査: 軽量暗号に関わる新しい技術動向

3 章の執筆担当者は下記の通りである.

第 3 章	軽量暗号に関わる新しい技術動向	
3.1 章	低レイテンシ暗号	崎山委員
3.2 章	サイドチャネル攻撃耐性	成吉委員
3.3 章	CAESAR プロジェクト	岩田委員
3.4 章	軽量暗号の活用事例および標準化動向	小川委員

3.1 低レイテンシ暗号

3.1.1 はじめに

Low-Latency Cryptography に関する論文のうち、特に欧州で研究が活発であるブロック暗号を用いた Low-Latency Encryption/Decryption について技術動向調査を行った。ハードウェア実装に関する論文 [1, 2, 3] を紹介し、今後の展望について述べる。

3.1.2 Low-Latency Cryptography 研究のモチベーション

暗号処理における低レイテンシ性は、暗号処理時の応答速度を重視するデータ通信アプリケーションに求められている。例えば、車の自動運転支援システム（Car2X communication)、セキュア・ストレージ及び CPU と外部ストレージ間のデータを暗号化するバス・エンクリプションである。半導体加工技術の高精度化（CMOS プロセスのの微細化）による集積回路の信号遅延時間短縮が大きく期待できない中、低レイテンシ暗号を実現するためには、暗号処理に要する計算量自体を大幅に削減する必要がある。これが、軽量暗号が新たに求められる理由のひとつと考える。現在広く使われている AES ブロック暗号では、回路規模、レイテンシともに上述のようなアプリケーションが求める性能要求を満たさない。例えば、1〜2 ns のレイテンシ性能を実現する AES 暗号ハードウェアは、現在の回路技術では実装が困難である。

3.1.3 ブロック暗号による Low-Latency Encryption/Decryption の性能評価

Knežević らによって CHES2012 で発表された論文 [1] では、暗号処理回路を 1 サイクルあるいは 2 サイクルでが完了するように実装し、数 10 MHz〜数 100 MHz のオーダーの最大動作周波数での処理時間をレイテンシとしている。つまり、レイテンシは数 ns〜数 10ns 程度となる。本報告では簡単のために、1 サイクルで処理が完了する場合についてのみ紹介する。90 nm CMOS テクノロジで合成した場合、 AES-128 のレイテンシは 14.8 ns、mCrypton-128 では 9.7 ns、PRESENT-128 では 14.3 ns と報告されている。Encryption/Decryption 両機能を搭載した場合、 AES-128 のレイテンシは約 17.8ns となり、性能の低下が見られるが、mCrypton-128 と PRESENT-128 ではそれぞれ 9.8 ns と 14.8 ns でとなる、ほとんど差異がないと評価されている。 3 つの暗号方式それぞれの回路規模は、AES-128, mCrypton-128, PRESENT-128 の順に、約 360 kGE、50 kGE、80 kGE（GE: Gate Equivalent の略、回路面積を表す単位）である。この結果から、mCrypton-128 が優れているように見えるが、安全性を犠牲にしている可能性がある。また、低レイテンシ暗号の場合には、回路規模はそれほど重要ではなく、むしろレイテンシに重きを置いた評価が好ましいと思われる。

Borghoff らによる ASIACRYPT2012 の発表論文 [2] で、低レイテンシのブロック暗号 PRINCE が提案された。4 ビット S-box による非線形演算と線形演算で構成されるデータ・パスは 64 ビット長で、鍵は 128 ビット長である。AES の鍵スケジュールと比べて、非常に単純な鍵スケジュール方式を採用している。回路規模は、約 8 kGE と報告されている。レイテンシは、45 nm CMOS テクノロジで 4.7 ns、90 nm CMOS テクノロジで 13.9 ns と報告されている。

SCIS2014 で、鈴木らは PRESENT と PRINCE の低レイテンシ実装を発表した [3]。PRESENT と PRINCE を 45 nm CMOS テクノロジで合成した結果、回路規模はそれぞれ 22 kGE と 8 kGE となり、レイテンシは 9.03 ns と 5.49 ns となった。ちなみに AES では 174 kGE で 12.25ns のレイテンシであった。ただし、以上の回路規模の数値は、暗号処理回路のみに基づくものであり、ARM プロセッサ向けの周辺モジュール用のバス・インターフェイス回路分（AMBA APB: 約 2 kGE）は含まない。この論文 [3] では、RFID タグへの実装に関する興味深い考察が与えられている。RFID タグ・チップのシリコン・ダイのサイズは、基板実装上の制限を受け、300μm 角程度が限界（下限）とされている。CMOS プロセスの微細化にともない、シリコンダイに実装できる回路規模が増大する。例えば 90 nm プロセスでは、300μm 角のシリコン・ダイに 30 kGE のロジック回路が搭載可能である。つまり、PRESENT と PRINCE は 90 nm（より微細な）CMOS テクノロジを用いることで、RFID タグに搭載できる。ただし、パッシブ RFID タグでは、低消費電力が重要となるため、この点は留意する必要がある。

3.1.4 まとめ

ここでは、低レイテンシを実現するいくつかの軽量ブロック暗号に関する技術動向調査を行った。ブロック暗号 PRINCE は、鍵拡張の単純化やデータ・パスの 64 ビット化により、回路規模の低減と低レイテンシ化の両方を同時に実現した。回路規模に対するレイテンシ性能は、アプリケーションによっては十分な性能と言える水準にあると考える。複数ラウンドを 1 サイクルで実装することは、サイドチャネル耐性の向上に繋がることが報告されている [4]。低レイテンシ実装においても同様の耐性向上が期待できるため、今後は、軽量暗号実装における耐タンパー性評価を併せて考える必要があると思われる。

参考文献

[1] Miroslav Knežević, Ventzislav Nikov, Peter Rombouts. Low-Latency Encryption - Is “Lightweight = Light + Wait”? In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems — CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 426-446,, Springer-Verlag, Berlin, Heidelberg, 2012.

[2] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knežević, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and-Tolga Yalçin. PRINCE — A Low-Latency Block Cipher for Pervasive Computing Applications — Extended Abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology — ASIACRYPT 2012 — 18th International Conference on the Theory and Application of Cryptology and Information Security*, volume 7658 of *Lecture Notes in Computer Science*, pages 208-225, Springer-Verlag, Berlin, Heidelberg, 2012

[3] 鈴木大輔, 菅原健, 佐伯稔. 軽量/低遅延暗号のハードウェア実装性能について. *2014* 年 暗号と情報セキュリティシンポジウム — *SCIS 2014*, 6 pages, 2014.

[4] Shivam Bhasin, Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, Unrolling Cryptographic Circuits, Unrolling Cryptographic Circuits: A Simple Countermeasure Against Side-Channel Attacks. In Josef Pieprzyk, editor, *Topics in Cryptology — CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010*, volume 5985 of *Lecture Notes in Computer Science*, pp.195-207, Springer-Verlag, Berlin, Heidelberg, 2010.

3.2 サイドチャネル攻撃耐性

本章では、軽量暗号技術に関する現状調査のうち、サイドチャネル攻撃耐性に関する文献調査結果を記載する。

3.2.1 調査対象

サイドチャネル攻撃耐性に関して、2012 年度まで CRYPTREC 暗号実装委員会にて活動していたサイドチャネルワーキンググループ活動の報告書 [1]、ならびにセキュリティ認証に関する規格 ISO/IEC15408 のコモンクライテリア承認アレンジメント (Common Criteria Recognition Arrangement) の web サイトに登録されている攻撃手法 [2] を参考に、以下 (i)(ii) を調査対象とし、調査結果を以降に記す。

(i) サイドチャネル攻撃 (リーク解析、電流解析。電磁波解析も含む)
(ii) 故障利用攻撃

物理攻撃、例えば文献 [2] には IC へのアクセスもしくは加工をおこなう物理解析などの記載はあるが、暗号アルゴリズムによる対策等の記載がなかったため、各軽量暗号アルゴリズムにおける物理攻撃耐性の評価は調査対象から除外した。

CRYPTREC Report 2012 暗号実装委員会報告ならびに文献 [2] に記載されていない攻撃手法、例えば文献 [3] は AES を攻撃対象として鍵抽出を試みているが、これらも調査対象外とした。調査対象とする暗号技術は、ブロック暗号のうち CRYPTREC 電子政府推奨暗号の AES、 TDES、 Camellia、ISO/IEC 29192-2 記載の PRESENT[4]、CLEFIA、ならびに LED[5]、 Piccolo[6]、 TWINE[7]、 PRINCE[8] とした。

3.2.2 軽量暗号アルゴリズムにおけるサイドチャネル攻撃 (リーク解析) の耐性調査

現状調査として、リーク解析 [9] における各種解析手法に関する文献、リーク解析に関する測定手法に関する文献、軽量暗号に関するリーク解析ならびに耐性を題目とした文献、リーク解析対策の最新手法、リーク耐性全般に関する文献について順に報告する。

3.2.2.1 リーク解析における各種解析手法の現状調査

ブロック暗号を対象としたリーク解析において、差分電力解析 (DPA) ならびにその派生の解析方法 [10] を示す。

1. DPA。演算途中の特定の 1 ビット (選択関数) に着目。入力を変化させたときの各候補鍵で計算した値とリークとの相関を算出し、鍵を推測する手法 [11]
2. CPA。演算途中の特定の値 (複数ビット) に着目。入力を変化させたときの各候補鍵で計算したハミングウエイト、ハミングディスタンス等とリークとの相関を算出し、鍵を推測する手法 [12]
3. High-order 型。演算途中に着目する箇所を複数個所とし、1.、 2. と同様に鍵を推測する手法 [13]
4. 相互情報量を用いた手法 [14]
5. template を用いた攻撃 [10][15]。鍵と入力を変化させて事前に各鍵の電力 (電磁波) のプロファイルを作成し、攻撃時には固定した鍵に対して入力を変化させ、プロファイルとリークとの比較により鍵を推測。
6. シミュレーション結果をリーク結果との相関の入力に与える手法 [16]。攻撃対象となる選択関数の値と、例えば論理シミュレーションでトグル回数を入手しておき、各候補鍵において相関を求めて鍵を推測する手法。

文献 [16] が顕著な例だが、ファンクションから回路を起こした時に発生した AES SBOX での過渡遷移を含んだ消費電力において、SBOX への DPA では 90 万サンプルなのに対して、シミュレーションの結果を相関関数の入力に適用した場合では 13 万サンプルと大幅に減ったとしている。以上から、実装時における過渡遷移の程度など消費電力モデルの精度にばらつきが発生し、プロセスならびに論理合成のコンフィグファイル等で過渡遷移の発生頻度も変わることが文献 [16] から容易に想定できるため、実装結果に関する論文間での厳密な比較は困難である。無対策の軽量暗号アルゴリズムにおいてデータに依存したリークの発生源となり得る演算回路の規模以外の観点からリーク耐性の優劣をつけるのは困難と推測する[*1] 。

3.2.2.2 リーク解析における測定手法の現状調査

リーク解析において電磁波観測を利用した手法 [17] が提案されてから久しいが、2013 年の国際ワークショップ CHES では下記の文献で 2NAND セルに関するリークの違いが報告されており、ゲートレベルですら無対策のものは攻撃されつつあることから、無対策の暗号アルゴリズムでは方式に依存せずに攻撃できるものと類推する。

・On Measurable Side-Channel Leaks inside ASIC Design Primitives[18]

本文献では電磁波リークを観測することでチップ動作を識別する研究がされており、以下の識別が可能とのこと。

- 2NAND セルに対して、入力 (1,1) の状態から入力 (0,0) への変化と入力 (0,1) の変化の区別が可能である (各 1 万波形取得後の平均での比較において)
- メモリのカラム線のアクセスの違いも識別可能

文献 [18] での環境にて鍵抽出評価を実施した場合は、既存の研究結果よりも大幅にサンプル数を減らすことが期待される。

評価環境については評価ボード SASEBO[19] あるいは ZUIHO をキャリブレーションとして使用することで一定の能力の担保はできているものと思われるが、電磁波解析などはコイルから測定場所の選定までパラメータが多く、評価結果に関する論文間の比較は困難である。

3.2.2.3 軽量暗号に関するリーク解析を題目とした文献調査

軽量暗号に関するリーク解析を題目とした発表を文献 [20] にて確認したため、その内容を報告する。本件は特定アルゴリズムに呼応した対策ではない。

文献 [20] においては Adiabatic logics(断熱的回路) を用いた手法でのサイドチャネル攻撃対策がメインである。面積のオーバーヘッドは存在するが、いわゆるグリッチタイプの瞬間的な消費電力の抑制によりサイドチャネル攻撃の耐性が急激に上昇しており、RFID などの低消費電力用途での対策において既存の MDPL[22] や RSL[23] [24] の同じセルレベルでのリーク対策方式と比較して向いているとしている。実チップ評価なし。

その消費電力抑制の効果から軽量暗号のことが触れられている。軽量暗号モジュールは消費電力が比較的小さいことから S/N 比が小さいことが強みである一方で、省電力技術がサイドチャネル攻撃への抵抗を弱めており、まとめとしてサイドチャネル攻撃の成功の可能性を大きくあげており、その実例としてブロック暗号である Keeloq を用いたアプリケーションへの攻撃 [25] を挙げている。

[*1] モジュール内において暗号演算と無関係な回路の動作が多いほど S/N 比が下がるので、そのような暗号アルゴリズムはリーク解析には有利に働く可能性はあると考える。

3.2.2.4 リーク解析対策の最新手法

リーク解析の対策においては秘密情報と秘密情報に依存した消費電力との相関をなくすというのが一般的な手法だが、リーク解析していることを検知することで秘密情報の流出を防ぐ新しいタイプの対策が最近提案されている。

文献 [21] によると、リーク解析のひとつである電磁波解析攻撃の対抗策として EM attack sensor と命名したセンサを暗号モジュールを搭載したチップに実装、実チップによる評価を実施している。EM attack sensor はコイルの形状をしている配線を有しており、その配線に一定の周波数の信号を流しておく。電磁波解析攻撃のため観測用のプローブを近づけると相互インダクタンスが発生し、上記信号の周波数がシフト。この周波数のシフトを観測することで攻撃を受けているかどうかを判別することでリークを防ぐ手法である。

3.2.2.5 リーク解析耐性の文献調査ならびにまとめ

厳密にリーク対策を実施しようとするとセルレベルでの対策、例えば MDPL[22] や RSL[23] [24] などといった手法の採用が必要と考える。それ故、対策の対象となる SBOX などの暗号演算処理部、具体的には NAND、 NOR セル使用部が小さいほど低面積、低消費電力の耐リークモジュールを実現できると考える。文献 [2] ではリーク解析を実装した各種暗号方式の電力解析の評価結果が記載されており、対策効果を確認したと結論づけている。ただし、対策セルを使用して実装した場合においても文献 [18] までを想定すると、論理的には同じでも実装した際の配線などの容量に依存してマスクの値が区別できると指摘している文献 [26] もあることから、レイアウトにおける対策も必要となることが想定される。これは面積だけではなく、設計工数にも大きく影響することを意味している。対策箇所が少ないほど設計工数の面からも優秀であり、これらは一般的に AES よりも軽量暗号のほうが優位に働くものと思われる。

最新リーク対策手法である EM attack sensor について暗号モジュールを搭載したチップに適用させた場合、電磁波攻撃をするためにはセンサを回避しなければならず、十分な起電力が得られない状況に陥ると推測される。この条件下において SBOX 単体への電磁波解析による鍵抽出を考えた場合、テーブルルックアップ方式で実装された 8 ビットの AES SBOX と多くの軽量暗号で採用されている 4 ビットの SBOX では、SBOX の回路規模に起因する消費電力の少なさから軽量暗号への攻撃のほうが困難になることが推測される。SBOX が小型化になることで、SBOX 以外の回路から発生されるノイズの比率が高くなる以外に、電磁波攻撃するためのコイルの最適なポジジョンの選定も SBOX の消費電力の少なさから見つけにくくなることが想定される。

3.2.2.6 リーク解析の現状調査に関する今後の課題

リーク解析への耐性の優劣に関する暗号アルゴリズム間の比較は文献調査だけでは限界があると考える。対策回路を実装した各種暗号方式に対し、文献 [18] 相当のリーク解析の実施が今後の課題である。

3.2.3 軽量暗号アルゴリズムにおける故障利用攻撃の耐性調査

3.2.3.1 故障利用攻撃の調査概要

文献 [27] をはじめとした、故障注入による鍵の抽出攻撃 DFA(Differential Fault Analysis) の容易性は暗号方式に依存する。DFA の攻撃に関する論文の多くが効率的な攻撃手法をシミュレーションなどを用い理論的に研究しているものであり、例えば実際にレーザを注入して特定段の一つ、あるいは複数の SBOX 等を攻撃して Differential Fault Analysis が可能かどうか評価した論文は皆無である。但し、レーザ装置とステージ装置の自動スキャンにより AES 暗号などを対象として DFA ができるツールは市販されており [28]、特に 1 か所への攻撃を想定しているものについて故障対策なく実装された場合は再現可能と考える。本ツールは対象暗号方式以外の他の暗号方式への応用も可能なものと

思われる。以降、各ブロック暗号に関して理論的な DFA 攻撃の研究事例を挙げる。

攻撃を受けることで想定される故障の種類として、演算器の出力などが一時的に誤り、その値を取り込んでしまうことで故障が発生するテンポラリなものと、中間値を格納するフリップフロップが反転するなど恒久的に値が変わってしまうパーマネントなものが考えられるが、ここでは両方とも実チップにおいて攻撃可能と判断する。前者は演算器の入力となる格納された値には故障が含まれていないことになる。

3.2.3.2 AES への故障利用攻撃の文献調査

鍵長 128 ビット使用時の AES への DFA について文献 [29] によると 8 段目の拡大鍵がストアされた領域への 1 ブロックに対してのフォルト注入攻撃において、1 ペアの結果で 2^8 の空間まで絞り込みが可能とのこと。鍵長 192 ビット使用時ならびに 256 ビット使用時の AES への DFA については文献 [30] によるとそれぞれ 3 ペア、4 ペアの結果で 2^{32} の空間まで絞り込むことが可能とのこと。

3.2.3.3 CLEFIA への故障利用攻撃の文献調査

鍵長 128 ビット使用時の CLEFIA への DFA について文献 [31] によると 2 か所への攻撃、2 ペアで平均 $2^{19.02}$ の探索空間まで絞り込むことが可能としている。文献 [32] によると、CLEFIA への DFA について、鍵長 128 ビット使用時は 2 ペアの攻撃結果のみ、鍵長 192 ビットならびに鍵長 256 ビット使用時では 2 ペアの攻撃結果で平均 $2^{10.78}$ の探索空間まで絞り込むことが可能としている。

鍵長 192 ビットならびに鍵長 256 ビット使用時の CLEFIA への DFA について文献 [33] によると、いずれも 8 ペアの攻撃結果で鍵が判明するとしている。

3.2.3.4 TDES への故障利用攻撃の文献調査

TDES ではないが、Single DES への DFA について文献 [34] によると、特定された single ビットへの攻撃を 12 段目で実施していき 7 ペアを入手すると、ランダムな場所への single ビットの故障注入の場合は 9 ペアを入手すると、それぞれ 99% 以上の確率で 16 段目の鍵が回復できるとしている。

3.2.3.5 PRESENT への故障利用攻撃の文献調査

PRESENT-80/128 への DFA に関して文献 [35] によると、2 バイトのランダムフォルトを 28 段目に注入することで、PRESENT-80 であれば 2 ペア、PRESENT-128 であれば 3 ペアで鍵を回復できるとしている。

3.2.3.6 LED への故障利用攻撃の文献調査ならびに対策に関する特記事項

64 ビットブロック暗号、64 ビット鍵である LED-64 への DFA に関して文献 [36] によると、29 段目に対して故障を注入することで、1 ペアで鍵探索空間を平均で $2^{4.03}$ まで絞り込むことができるとしている (鍵探索空間の調査においてはランダムに生成した誤り暗号文ペア 50 組に対して、実際に攻撃を適用後の鍵候補数から算出)。

LED-64 は拡大鍵として使用する 64 ビットの鍵を全て同じ鍵としており、LED-128 は 64 ビット長 2 組の拡大鍵を交互に使用するため、演算中での拡大鍵の演算は不要である。故に、故障攻撃可能な範囲が狭くなる、対策回路を実装したときの負担低減などのメリットが考えられる (但し、文献 [36] は鍵スケジュール部ではなく、暗号処理中の中間値への攻撃)。

3.2.3.7 Piccolo、 TWINE への故障利用攻撃

研究が開始されたところである。暗号演算部分の 1 ビットあるいは 1 ニブルのレーザ攻撃ではないが、ソフトウェアによる暗号実装において命令への故障攻撃を想定したものとしては、64 ビットブロック暗号で 80 ビット鍵の Piccolo-80、 同じく 64 ビットブロック暗号で 80 ビット鍵の TWINE-80 に対して、正しい暗号文と故障注入により誤った二つの暗号文の組で鍵を抽出できるとしており、128 ビット鍵の CLEFIA-128 より容易という報告は出ている [37]。

3.2.3.8 PRINCE への故障利用攻撃ならびに対策に関する特記事項

64 ビットブロック暗号、128 ビット鍵である PRINCE の 10 段目に対して 1 ニブルの攻撃を実施。1000 例による PC での探索空間調査の結果、4 回の故障注入で 2^{18} 未満の探索空間まで絞り込むことができるとしている [38]。

PRINCE は拡大鍵として使用する 64 ビットの鍵を全て同じ鍵としており、拡大鍵の演算は実装不要である。故に、故障攻撃可能な範囲が狭くなる、対策回路を実装したときの負担低減などのメリットが考えられる (但し、文献 [38] は鍵スケジュール部ではなく、暗号処理中の中間値への攻撃)。

3.2.3.9 複数の暗号方式を対象とした故障利用攻撃の文献調査

多数の暗号方式への故障利用攻撃の最近の調査として文献 [39] が挙げられる。本文献では一般型 Feistel 構造への故障利用攻撃を比較しており、対象は DES(single)、 TWINE、CLEFIA 等。解析のしかたはオーソドックスで、ラウンドの前後でフォルトが伝搬するブロック関係を行列で表記。各段において single ビットの故障を与えたとき、Subkey ブロックのうちアタックされた個数、故障利用攻撃の際に中間値を推測した候補の数をまとめており、過去に発表された論文、例えば文献 [31] との比較を行いながら、本解析手法における故障利用攻撃の最適な攻撃の段数をまとめている。

3.2.3.10 故障利用攻撃耐性のまとめ

鍵長 128 ビット使用時の AES、 LED-64 が 1 ペアで鍵探索空間を 2^8 以下まで絞りこみ可能となっており、耐性が比較的低い。一方、TDES は鍵を 56 ビット毎 3 回に分けて使用するため、故障利用攻撃の耐性が比較的高いと考える。

また、今回調査した軽量暗号への故障利用攻撃の多くが 2012 年から 2014 年に発表されたものであるため、今後の研究により更なる故障注入回数の低減の可能性があると考える。

実チップへの攻撃については、拡大鍵演算部がないなど演算回路規模の小さいもののほうが攻撃範囲が狭いなどの可能性がある。更に、上記にも記載した文献 [29] の AES の攻撃に関しては拡大鍵の演算結果のブロックの一つにパーマネントのエラーを注入することで、中間値だけではなく拡大鍵計算時に故障が伝搬することも利用しており、特にオンザフライによる実装の脆弱性を確認しているが、上記軽量暗号の中には拡大鍵演算実行不要のものが提案されており、拡大鍵の故障伝搬による攻撃が利用できないという点で従来より故障耐性が高いと考えることが出来る。

次に、二回演算ならびに逆算による対策、冗長回路の実装、各種センサの実装による 3 つの主な対策手法において、それぞれ軽量暗号に適用した際の AES と比較しての優劣を記述する。

■二回演算ならびに逆算による対策 対策方法のひとつに文献 [40] に記載されている二回計算 (Doubling)、逆算などが考えられる。文献 [40] では DES を例にとっているが、他の共通鍵暗号方式においても適用可能と考える。但し、文献 [40] ではレーザによる故障利用攻撃において同じ場所に複数回照射する攻撃例や、複数の箇所にレーザを照射させる攻撃でこれらの対策を無効にすることが出来るとしており、攻撃者の能力や攻撃費用を想定して必要相当の対策を講

じることを DES と同様、他の暗号を実装した場合にも求められる。Doubling 対策を実装した実チップへのレーザ攻撃成功例については文献 [41] が挙げられる。複数回演算による対策を施す場合、一般的に暗号処理時間が高速である軽量暗号のほうが AES と比較して追加対策によるレイテンシ増加を抑えることが期待できる。

■冗長回路の実装による対策　冗長化、二重化など追加回路の実装による故障対策も考えられる。例えば文献 [42] で、冗長の程度と検出率を比較している。上記テンポラリーエラーが発生することで故障が注入された場合、単に中間値が格納されているフリップフロップ等に冗長ビットを持たせただけでは検出できない可能性がある。以上から冗長化等による対策の場合、演算器等を含めた暗号実装本体の面積に比例するものと思われ、一般的に軽量暗号が AES などと比較して面積コストの観点から優位に働くものと想定される。なお、上記二回演算等と同様に、冗長回路あるいは多重化された回路と元の回路の双方に攻撃される可能性についての脅威分析は必要であり、分析結果に応じて両方の回路が攻撃された場合の追加の対策が必要となるが、分析の必要性、対策実施の有無は暗号方式には依存しないものと思われる。

■各種センサによる対策　各種故障攻撃を各センサで対処する方法も考えられる。例えば、レーザなどの光源をチップ表面、あるいは裏面から局所的に照射することで故障利用攻撃を試みる手法に対して、光センサをチップ内にちりばめるように実装することで故障を防ぐ方法も提案されている [43]。本方式による対策の場合、光センサ実装による面積増は暗号実装本体の面積に比例するものと想定できることから、一般的に軽量暗号が AES などと比較して面積コストの観点から優位に働くものと想定される。電磁波注入によるチップへの局所攻撃 [44] も出てきているが、暗号アルゴリズムへの故障利用攻撃に使用された場合のセンサ複数配置による対策についても光センサと同様、面積コストの観点から一般的に軽量暗号が優位と考える。電源グリッチ [28] による故障利用解析をセンサ等で防御する場合は、チップ全体の電源回りの設計に大きく依存することになるため、暗号方式による面積コストの優位不利は少ないものと思われる。

3.2.3.11　故障利用攻撃手法の応用

故障利用攻撃の応用として、AES 演算における鍵長 128 ビット使用時の攻撃において Differential ではなく、攻撃により誤った暗号文のみを集めて鍵を復元する試みも文献 [45] でおこなわれている。故障注入の成功率が 50% から 100% それぞれにおいて、ラウンド 7 への攻撃において 4 から 10 の誤ったメッセージで 2^0 から $2^{39.7}$ の鍵候補の絞り込みが 62 から 100% の確率で出来ると調査されている。

3.2.3.12　故障利用攻撃の現状調査に関する今後の課題

故障利用攻撃に関しても文献調査のみならず、厳密には実チップによる各暗号アルゴリズムでの比較対象が望ましい。実チップによる故障利用攻撃、耐性評価は今後の課題である。

参考文献

[1] CRYPTREC Report 2012 暗号実装委員会報告

[2] CCRA. Application of Attack Potential to Smartcards CCDB-2013-05-002, `http://www.commoncriteriaportal.org/cc/`

[3] Pascal Manet, and Bruno Robisson. Differential Behavioral Analysis. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 413–426. Springer-Verlag, Berlin, Heidelberg, New York, 2007.

[4] Andrey Bogdanov, Lars Ramkilde Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and Charlotte H. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer-Verlag, Berlin, Heidelberg, New York, 2007.

[5] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer-Verlag, Berlin, Heidelberg, New York, 2011.

[6] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 342–357. Springer-Verlag, Berlin, Heidelberg, New York, 2011.

[7] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight, versatile block cipher. In Gregor Leander and François-Xavier Standaert, editors, *ECRYPT Workshop on Lightweight Cryptography*, pages 146–169. ECRYPT II, 2011.

[8] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, Tolga Yalçn, " PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications ", In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[9] Paul Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Neal Koblitz , editor, *Advances in Cryptology - CRYPTO 1996 - 16th Annual International Cryptology Conference Santa Barbara, California, USA, August 18-22, 1996. Proceedings*, volume 1109 of *Lecture Notes*

in Computer Science, pages 104–113. Springer-Verlag, Berlin, Heidelberg, New York, 1996. http://www.cryptography.com/public/pdf/TimingAttacks.pdf

[10] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. Power Analysis Attacks Revealing the Secrets of Smart Cards. 2007 Springer.

[11] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. http://www.cryptography.com/public/pdf/DPA.pdf

[12] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. In Marc Joye, and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems — CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer-Verlag, Berlin, Heidelberg, New York, 2004.

[13] Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer-Verlag, Berlin, Heidelberg, New York, 2000.

[14] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems — CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer-Verlag, Berlin, Heidelberg, New York, 2008.

[15] S. Chari, J.R. Rao, and P. Rohatgi. Template Attacks. In Burton S. Kaliski, Çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[16] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 157–171. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[17] D. Agrawal, B. Archambeault, J.R. Rao, and P. Rohatgi. The EM Side-channel(s). In Burton S. Kaliski, Çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45. Springer-Verlag, Berlin, Heidelberg, New York, 2002

[18] Takeshi Sugawara, Daisuke Suzuki, Minoru Saeki, Mitsuru Shiozaki, and Takeshi Fujino. On Measurable Side-Channel Leaks inside ASIC Design Primitives. In Guido Bertoni, and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems — CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 159–178. Springer-Verlag, Berlin, Heidelberg, New York, 2013

[19] http://www.risec.aist.go.jp/project/sasebo/

[20] Amir Moradi and Axel Poschmann. Lightweight Cryptography and DPA Countermeasures: A Survey. http://emsec.rub.de/media/crypto/veroeffentlichungen/2010/09/05/wlc.pdf

[21] Naofumi Homma, Yu-ichi Hayashi, Noriyuki Miura, Daisuke Fujimoto, Daichi Tanaka, Makoto Nagata, and Takafumi Aoki. EM Attack Is Non-invasive? - Design Methodology and Validity Verification of EM Attack Sensor. In Lejla Batina, and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems — CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, Berlin,

Heidelberg, New York, 2014

[22] T. Popp and S. Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance without Routing Constraints. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[23] D. Suzuki, M. Saeki, and T. Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive, Report 2004/346, 2004. `http://eprint.iacr.org/`

[24] D. Suzuki, M. Saeki, and T. Ichikawa. Random Switching Logic: A New Countermeasure against DPA and Second-Order DPA at the Logic Level. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E90-A(1): pages 160–168. IEICE, 2007.

[25] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoqCode Hopping Scheme. In Wagner David, editors *Advances in Cryptology - CRYPTO 2008 - 28th Annual International Cryptology Conference Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 203–220. Springer-Verlag, Berlin, Heidelberg, New York, 2008.

[26] Patrick Schaumont and Kris Tiri. Masking and Dual-Rail Logic Don't Add Up. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 95–106. Springer-Verlag, Berlin, Heidelberg, New York, 2007.

[27] D. Boneh, R. A. DeMillo, and R. J. Lipton. A New Breed of Crypto Attack on "Tamperproof" Tokens Cracks Even the Strongest RSA Code. 1996.

[28] RISCURE 社. `https://www.riscure.com/`

[29] Sk Subidh Ali and Debdeep Mukhopadhyay. A Differential Fault Analysis on AES Key Schedule using Single Fault. In Luca Breveglieri, Sylvain Guilley, Israel Koren, David Naccache, and Junko Takahashi, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on Date 28-28 Sept. 2011 IEEE computer society*, pages 54–64, IEEE, 2011.

[30] 高橋 順子、福永 利徳Differential Fault Analysis on AES with 192 and 256-bit keys. *2010年 暗号と情報セキュリティシンポジウム — SCIS 2010*. 2010.

[31] Junko Takahashi, and Toshinori Fukunaga. Improved Differential Fault Analysis on CLEFIA. In Luca Breveglieri, Shay Gueron, Israel Koren, David Naccache, and Jean-Pieere Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2008 Workshop on Date 10-10 Aug. 2008 IEEE computer society*, pages 25–34. IEEE, 2008.

[32] Junko Takahashi, Toshinori Fukunaga. Differential Fault Analysis on CLEFIA with 128, 192, and 256-Bit Keys. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E93-A No.1 pages 136–143. IEICE, 2010

[33] S Ali, and D Mukhopadhyay. Improved Differential Fault Analysis of CLEFIA. In Wieland Fischer, and Jorn-Marc Schmidt, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on Date 20-20 Aug. 2013 IEEE computer society*, pages 60–72. IEEE, 2013.

[34] Matthieu Rivain, Emmanuel Prouff, Julien Doget. Differential Fault Analysis on DES Middle Rounds. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems — CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 457–470. Springer-Verlag, Berlin, Heidelberg, New

York, 2009.

[35] Kitae Jeonga, Yuseop Leea, Jaechul Sungb, and Seokhie Honga. Improved differential fault analysis on PRESENT-80/128. *International Journal of Computer Mathematics, Volume 90, Issue 12*, pages 2553-2563. 2013.

[36] 上野 嶺、 本間 尚文、 青木 孝文. LED 暗号への単一の故障注入を用いた差分故障解析とその評価. *2014* 年 暗号と情報セキュリティシンポジウム — *SCIS 2014*. 2014.

[37] Hideki YOSHIKAWA, Masahiro KAMINAGA, Arimitsu SHIKODA, and Toshinori SUZUKI. Round Addition DFA on 80-bit Piccolo and TWINE. *IEICE Transactions on Information and Systems*, Vol.E96-D No.9 pages 2031–2035. IEICE, 2013.

[38] Ling Song, Lei Hu. Differential Fault Attack on the PRINCE Block Cipher. `http://eprint.iacr.org/2013/043.pdf`

[39] Helene Le Bouder, Gael Thomas, Yanis Linge and Assia Tria. On Fault Injections in Generalized Feistel Networks. *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on Date 23-23 Sept. 2014 IEEE computer society*, pages 83–93. IEEE, 2014.

[40] Rob Bekkers and Hans König“ Fault Injection, a Fast Moving Target in Evaluations ”, FDTC2011, IEEE computer society, p.65, IEEE. `http://conferenze.dei.polimi.it/FDTC11/shared/FDTC-2011-keynote-2.pdf`

[41] 大野 仁、土屋 遊、 中田 量子、松本 勉. IC カードへのレーザー照射フォールト攻撃は単純な冗長実装では防げない. *2014* 年 暗号と情報セキュリティシンポジウム — *SCIS 2014*. 2014.

[42] Tal G. Malkin, François-Xavier Standaert, Moti Yung. A Comparative Cost/Security Analysis of Fault Attack Countermeasures. In Luca Breveglieri, Israel Koren, David Naccache, and Jean-Pierrei Seifert, editors, *Fault Diagnosis and Tolerance in Cryptography: Third International Workshop, FDTC 2006*, volume 4236 of *Lecture Notes in Computer Science*, pages 159–172. Springer-Verlag, Berlin, Heidelberg, New York, 2006.

[43] Odile Derouet. Secure Smartcard Design against Laser Fault Injection Attacks (invited), FDTC2007, `http://conferenze.dei.polimi.it/FDTC07/Derouet_remaster.pdf`

[44] F. Poucheret, K. Tobich, M. Lisarty, L. Chusseauz, B. Robissonx, and P. Maurine. Local and Direct EM Injection of Power Into CMOS Integrated Circuits. In Luca Breveglieri, Sylvain Guilley, Israel Koren, David Naccache, and Junko Takahashi, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2011 Workshop on Date 28-28 Sept. 2011 IEEE computer society*, pages 100–104, IEEE, 2011.

[45] Thomas Fuhr, Eliane Jaulmes, Victor Lomné and Adrian Thillard. Fault Attacks on AES with Faulty Ciphertexts Only. In Wieland Fischer, and Jorn-Marc Schmidt, editors, *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on Date 20-20 Aug. 2013 IEEE computer society*, pages 108–118. IEEE, 2013.

3.3 CAESAR プロジェクト

本章では、暗号技術調査 WG (軽量暗号 WG) の外部動向調査として、CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) プロジェクトについてまとめる。本プロジェクトのウェブサイトは `http://competitions.cr.yp.to/caesar.html` である。

3.3.1 CAESAR プロジェクト

■プロジェクト発足の背景 認証暗号は、データの暗号化と認証を同時に行うための共通鍵暗号技術である。AES-CCM や AES-GCM など、すでに標準化され実用化されている認証暗号では、オンラインではない (あらかじめ入力データ長を決めないと処理を開始できない)、計算効率を改善させる余地がある、証明可能安全性に不備がある、弱鍵が存在する、といった様々な問題点が指摘されている。

また、OpenSSH や TLS、802.11 ネットワークにおける WEP などで安全性の問題点が指摘されており、認証暗号はこれらの問題の解決策として期待されている。一方、これらにおいて認証暗号の普及は遅れており、現状の認証暗号の計算効率が、たとえば RC4 などより劣る点にその原因の一つがあると考えられる。

■プロジェクトの目標 本プロジェクトの目標は、(1) AES-GCM より (安全性、計算効率、実装効率、あるいはその他何らかの点において) 優れていて、なおかつ (2) 広範に実用化されることに適した認証暗号のポートフォリオを選定することにある。

■プロジェクトの概要 本プロジェクトでは認証暗号アルゴリズムの公募を行う。応募締め切りは 2014 年 3 月であり、誰でも応募が可能である。公募されたアルゴリズムは第一ラウンドアルゴリズムであり、おおよそ 1 年の評価期間を経て 2015 年 1 月に第二ラウンド進出アルゴリズムを決定する。その後さらに 1 年の評価期間を経て 2015 年 12 月に第三ラウンド進出アルゴリズムを決定し、さらに 1 年の評価期間を経て 2016 年 12 月に最終候補アルゴリズムを決定する。ポートフォリオのアナウンスは 2017 年 12 月を予定している。

本プロジェクトは研究者主体で進められるものであり、ポートフォリオは標準を意味するものではない (ただし、本プロジェクトは NIST によるスポンサーシップを受けている)。また、各ラウンドに進出するアルゴリズムの決定では、選定委員による投票が行われる予定である。

本プロジェクトでは各提案者の設計指針に応じて安全性、機能、実装性能、計算効率など様々な評価要素が考えられ、「軽量」性についても評価要素に入ることが予想される。

■スケジュール詳細 下記スケジュールを予定している[*2]。

- M-20, 2012.07.05–06: DIAC: Directions in Authenticated Ciphers. Stockholm.
- M-14, 2013.01.15: Competition announced at the Early Symmetric Crypto workshop in Mondorf-les-Bains; also announced online.
- M-7, 2013.08.11–13: DIAC 2013: Directions in Authenticated Ciphers 2013. Chicago.
- M0, 2014.03.15: Deadline for first-round submissions.
- M1, 2014.05.15: Deadline for first-round software.

[*2] 2015 年 2 月 20 日現在。頻繁に更新されており、最新情報はプロジェクトのウェブサイト `http://competitions.cr.yp.to/caesar.html` より確認できる。

- M5 2014.08.23–24: DIAC 2014: Directions in Authenticated Ciphers 2014. Santa Barbara.
- M12 (tentative), 2015.03.15: Announcement of second-round candidates.
- M13 (tentative), 2015.04.15: Deadline for second-round tweaks.
- M14 (tentative), 2015.05.15: Deadline for second-round software.
- M15 (tentative), 2015.06.15: Deadline for second-round Verilog/VHDL.
- 2015 summer (tentative): DIAC 2015.
- M21 (tentative), 2015.12.15: Announcement of third-round candidates.
- M22 (tentative), 2016.01.15: Deadline for third-round tweaks.
- M23 (tentative), 2016.02.15: Deadline for third-round software.
- M24 (tentative), 2016.03.15: Deadline for third-round Verilog/VHDL.
- 2016 summer (tentative): DIAC 2016.
- M33 (tentative), 2016.12.15: Announcement of finalists.
- M34 (tentative), 2017.01.15: Deadline for finalist tweaks.
- M35 (tentative), 2017.02.15: Deadline for finalist software.
- M36 (tentative), 2017.03.15: Deadline for finalist Verilog/VHDL.
- 2017 summer (tentative): DIAC 2017.
- M45 (tentative), 2017.12.15: Announcement of final portfolio.

■公募要領　2014 年 1 月 27 日に公募要領の最終版が公表された。2014 年 3 月の応募時点では下記情報を含めたドキュメントを提出する。

- 方式の名称、設計者、応募者、連絡用メールアドレス
- 仕様
- 安全性のゴール
- 安全性解析
- 特筆すべき事項、特徴
- 設計の合理性
- 知的財産に関する事項
- 応募に際して合意する事項

その後 2014 年 5 月 15 日までにソフトウェアでのレファレンスコードを提出する。また、第二ラウンド進出アルゴリズムについては、応募者は 2015 年 4 月までにハードウェアでのレファレンス実装を提出する。

■選定委員　選定委員は下記 22 名のメンバーからなる。

1. Steve Babbage (Vodafone Group, UK)
2. Daniel J. Bernstein (University of Illinois at Chicago, USA, and Technische Universiteit Eindhoven, Netherlands); secretary, non-voting
3. Alex Biryukov (University of Luxembourg, Luxembourg)
4. Anne Canteaut (Inria Paris-Rocquencourt, France)
5. Carlos Cid (Royal Holloway, University of London, UK)
6. Joan Daemen (STMicroelectronics, Belgium)

7. Christophe De Cannière (Google, Switzerland)
8. Orr Dunkelman (University of Haifa, Israel)
9. Henri Gilbert (ANSSI, France)
10. Tetsu Iwata (Nagoya University, Japan)
11. Lars R. Knudsen (Technical University of Denmark, Denmark)
12. Stefan Lucks (Bauhaus-Universität Weimar, Germany)
13. David McGrew (Cisco Systems, USA)
14. Willi Meier (FHNW, Switzerland)
15. Kaisa Nyberg (Aalto University School of Science, Finland)
16. Bart Preneel (COSIC, KU Leuven, Belgium)
17. Vincent Rijmen (KU Leuven, Belgium)
18. Matt Robshaw (Impinj, USA)
19. Phillip Rogaway (University of California at Davis, USA)
20. Greg Rose (Qualcomm Technologies Inc., USA)
21. Serge Vaudenay (EPFL, Switzerland)
22. Hongjun Wu (Nanyang Technological University, Singapore)

■応募方式一覧　下記の 57 方式が提案された。方式の名称と設計者を記載している。冒頭の (L) は、軽量性を特徴として挙げている方式を示している[*3]。

1. (L) ACORN: v1 (Hongjun Wu)
2. (L) ++AE: v1.0 (Francisco Recacha)
3. AEGIS: v1 (Hongjun Wu, Bart Preneel)
4. AES-CMCC: v1, v1.1 (Jonathan Trostle)
5. AES-COBRA: v1, withdrawn, (Elena Andreeva, Andrey Bogdanov, Martin M. Lauridsen, Atul Luykx, Bart Mennink, Elmar Tischhauser, Kan Yasuda)
6. AES-COPA: v1 (Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, Kan Yasuda)
7. AES-CPFB: v1 (Miguel Montes, Daniel Penazzi)
8. (L) AES-JAMBU: v1 (Hongjun Wu, Tao Huang)
9. AES-OTR: v1 (Kazuhiko Minematsu)
10. AEZ: v1 (Viet Tung Hoang, Ted Krovetz, Phillip Rogaway)
11. Artemia: v1 (Javad Alizadeh, Mohammad Reza Aref, Nasour Bagheri)
12. (L) Ascon: v1 (Christoph Dobraunig, Maria Eichlseder, Florian Mendel, Martin Schläffer)
13. AVALANCHE: v1 (Basel Alomair)
14. (L) Calico: v8, withdrawn, (Christopher Taylor)
15. CBA: v1 v1-1 (Hossein Hosseini, Shahram Khazaei)
16. (L) CBEAM: r1, withdrawn, (Markku-Juhani O. Saarinen)

[*3] “lightweight” をキーワードとして応募ドキュメントを検索し、軽量性を方式の特徴として挙げているか、あるいは使用している演算や構成要素を軽量性を考慮して選定している方式をピックアップした。

17. CLOC: v1 (Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka)
18. (L) Deoxys: v1 (Jérémy Jean, Ivica Nikolić, Thomas Peyrin)
19. (L) ELmD: v1 (Nilanjan Datta, Mridul Nandi)
20. Enchilada: v1 v1.1 (Sandy Harris)
21. (L) FASER: v1, withdrawn, (Faith Chaza, Cameron McDonald, Roberto Avanzi)
22. HKC: v1, withdrawn, (Matt Henricksen, Shinsaku Kiyomoto, Jiqiang Lu)
23. HS1-SIV: v1 (Ted Krovetz)
24. ICEPOLE: v1 (PawełMorawiecki, Kris Gaj, Ekawat Homsirikamol, Krystian Matusiewicz, Josef Pieprzyk, Marcin Rogawski, Marian Srebrny, Marcin Wojcik)
25. iFeed[AES]: v1 (Liting Zhang, Wenling Wu, Han Sui, Peng Wang)
26. (L) Joltik: v1 (Jérémy Jean, Ivica Nikolić, Thomas Peyrin)
27. Julius: v1.0 (Lear Bahack)
28. (L) Ketje: v1 (Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche, Ronny Van Keer)
29. Keyak: v1 (Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche, Ronny Van Keer)
30. (L) KIASU: v1 (Jérémy Jean, Ivica Nikolić, Thomas Peyrin)
31. (L) LAC: v1 (Lei Zhang, Wenling Wu, Yanfeng Wang, Shengbao Wu, Jian Zhang)
32. Marble: v1.0 (Jian Guo)
33. McMambo: v1, withdrawn, (Watson Ladd)
34. (L) Minalpher: v1 (Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, Shoichi Hirose)
35. MORUS: v1 (Hongjun Wu, Tao Huang)
36. NORX: v1 (Jean-Philippe Aumasson, Philipp Jovanovic, Samuel Neves)
37. OCB: v1 (Ted Krovetz, Phillip Rogaway)
38. OMD: v1.0 (Simon Cogliani, Diana-Ştefania Maimuţ, David Naccache, Rodrigo Portella do Canto, Reza Reyhanitabar, Serge Vaudenay, Damian Vizár)
39. PAEQ: v1 (Alex Biryukov, Dmitry Khovratovich)
40. PAES: v1, withdrawn, (Dingfeng Ye, Peng Wang, Lei Hu, Liping Wang, Yonghong Xie, Siwei Sun, Ping Wang)
41. PANDA: v1, withdrawn, Dingfeng Ye, Peng Wang, Lei Hu, Liping Wang, Yonghong Xie, Siwei Sun, Ping Wang)
42. (L) π-Cipher: v1 (Danilo Gligoroski, Hristina Mihajloska, Simona Samardjiska, Håkon Jacobsen, Mohamed El-Hadedy, Rune Erlend Jensen)
43. POET: v1 (Farzaneh Abed, Scott Fluhrer, John Foley, Christian Forler, Eik List, Stefan Lucks, David McGrew, Jakob Wenzel)
44. POLAWIS: v1 (Arkadiusz Wysokinski, Ireneusz Sikora)
45. (L) PRIMATEs: v1 (Elena Andreeva, Begül Bilgin, Andrey Bogdanov, Atul Luykx, Florian Mendel, Bart Mennink, Nicky Mouha, Qingju Wang, Kan Yasuda)
46. (L) Prøst: v1 (Elif Bilge Kavun, Martin M. Lauridsen, Gregor Leander, Christian Rechberger, Peter Schwabe, Tolga Yalçın)
47. Raviyoyla: v1 (Rade Vuckovac)

48. (L) Sablier: v1 (Bin Zhang, Zhenqing Shi, Chao Xu, Yuan Yao, Zhenqi Li)
49. (L) SCREAM: v1 (Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, Kerem Varici, François Durvaux, Lubos Gaspar, Stéphanie Kerckhof)
50. SHELL: v1 (Lei Wang)
51. (L) SILC: v1 (Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, Eita Kobayashi)
52. Silver: v1 (Daniel Penazzi, Miguel Montes)
53. STRIBOB: v1 (Markku-Juhani O. Saarinen)
54. Tiaoxin: v1.0 (Ivica Nikolić)
55. TriviA-ck: v1 (Avik Chakraborti, Mridul Nandi)
56. Wheesht: v1 (Peter Maxwell)
57. YAES: v1 v2 (Antoon Bosselaers, Fre Vercauteren)

3.3.2 まとめ

本章では、暗号技術調査 WG (軽量暗号 WG) の外部動向調査として、CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) プロジェクトについてまとめた。AES コンペティション、NESSIE プロジェクト、eSTREAM プロジェクト、SHA-3 プロジェクトに続く国際的なコンペティションであり、継続的に注視していくことが求められる。

3.4 軽量暗号の活用事例および標準化動向調査

3.4.1 調査目的

今後の暗号の開発において、活用事例・標準化動向から軽量暗号に関する要求条件を導き出し、研究開発、標準化の指針を得る。

3.4.2 活用事例調査

3.4.2.1 調査方法

以下に示す軽量暗号が活用されると期待されている分野について公開されている情報を調査する。

- RFID
- センサーネットワーク（環境測定等）
- 医療センサ
- ITS、自動車
- 記録メディア（HDD、SSD 等）
- 携帯端末（携帯電話、タブレット端末、ポータブルゲーム機等）
- その他

3.4.2.2 調査報告

3.4.2.1 章に挙げたそれぞれの項目について調査を行った。現段階で実際に軽量暗号が使用されているという公開の情報はない。そこで、これらの項目について、暗号についてどのように利用されているか、その中で軽量暗号がどのような導入方法が考えられるかを考察する。

■RFID、センサーネットワーク（環境測定等）、携帯端末（携帯電話、タブレット端末、ポータブルゲーム機等） これらについては、一般論となる情報のみが公開されていた [1, 2, 3]。無線ネットワーク接続機能を持つ RFID がインターネットのようなオープンなネットワークに接続する際に暗号を利用する。ほとんどの RFID、センサー、携帯端末デバイスの CPU は低スペックであり、信号処理を行う能力に乏しいこと、またメモリサイズも小さい。さらに、低消費電力での実装をしなければならず、軽量暗号に対する期待は極めて大きい。

■医療センサ Texas Instrument[4] では、血圧、体温、心拍数、ブトウ糖等の測定を行い、Bluetooth で通信を行い、低消費電力のポータブル機器用のデバイス、MSP430FR59xx ファミリーを提供している。このデバイスの記事では、“ 政府標準である ” 256-bit AES を用いた医療用センサと記載されている。MSP430FR59xx ファミリーのスペックでは、MSP430microcotroller（16bitRISC CPU）を使用、メモリサイズは 32KB-256KB、消費電力は不明である。また、Position Paper[5] での報告として遠隔健康ケアシステムでは、埋め込みデバイスが使われており、これらとのコミュニケーションをとる際にセキュリティ技術が必要。そして、これらを Ultra-low-power で行いたいとしている。この論文では、待機電力を減らす、置換を減らすなどによるアルゴリズムの簡素化が主体で低消費電力化を図っている。

■ITS、自動車 ITS（Intelligent Transport Systems:高速道路交通システム）は、人と道路と自動車の間で情報の共有を行い、交通の最適化を図るシステムとして作られたシステムである [6]。そして、” 安全 ” が強調されたシステム造

りが目指されている。そして、その基本構成である自動車搭載機器について、装置機器間のデータの秘匿、認証のために暗号が利用されることが謳われている [7, 8, 9, 10]。これらについても、リソースが限られているとはわかっているものの、軽量暗号を使用する段階には至っておらず、軽量暗号を使うことを提案している段階 [11] である。

■記録メディア（HDD、SSD 等） SandForce[12] ではセキュリティ機能を持つ SSD を提供している。Windows8 の PC やタブレットにおいても低消費電力の記録デバイス（SSD）が必要であった。このため、従来品が 20mA を利用していたのに対し、0.05mA で動作するようにしている。モバイル端末でも利用可能であるとのこと。

■その他 ICT 社会において、ETC システムにおいても暗号化が使われている。このシステムではプライバシ保護のため、認証、暗号化などが必要となる [13]。 その他、軽量暗号の一般的な利用可能性については、軽量暗号関係の多くの論文 [14, 15] で書かれている。

3.4.2.3 ヒアリング

メーカー数社にヒアリングを行い、軽量暗号に対する考え方を調査した。その結果を以下に示す。最近の測定器や家電はほぼ CPU が積まれ、ネットワーク接続が可能となっているが、使用目的によって要求条件が異なっている。大きくわけて以下の 2 つのケースがある。

1. 一つのハードウェアもしくはソフトウェアに入れる機能が確定している場合
2. 一つのハードウェアもしくはソフトウェアに入れる機能が確定しておらず、いろいろな機能を入れる場合

前者は医療センサのように使用用途が厳密に限られる場合、後者は PC やタブレットのように汎用の機器であり、使用目的が厳密に制限されていない場合である。

■ケース 1 について

- セキュリティが必要で AES を入れたければ、外部モジュールとして AES チップを使う、もしくは、ソフトウェアとして AES を入れられるスペックの CPU やメモリを搭載する。
- 軽量暗号をあえて導入する必要を感じていない。チップサイズ（消費電力を含む）が小さくなればよいという一般論があるが、どれほど小さいチップが必要であるかの指定はない。

■ケース 2 について

- CPU、メモリなどのリソースをそれぞれの機能でシェアして使用する。欠点として、機能が多くなりリソースを取り合うことが生じる。
- ハードウェア、ソフトウェアの構築段階でどの機能を必須として使うかを決める。これにより、リソースを分割する度合い（量、順位）が決まる。
- AES が使用困難であり、軽量暗号であれば使用可能、というアプリケーションは少ない。
- リソースを削られたとしても高速な動作が保障されるような暗号方式として軽量暗号が求められていることはある。

3.4.2.4 活用事例のまとめ

軽量暗号に対する要求はあるものの、具体的なスペックまで落とし込んだ要求条件は出ていない。ただし、医療センサの項で紹介した Texas Instrument のように、標準として認められることで使用する業者があるということも事実で

ある。従って、CRYPTREC などの機関で調査・評価することは、軽量暗号の利用促進に供する情報を提供できると考えられる。

3.4.3 標準化動向調査

3.4.3.1 調査方法

軽量暗号の標準化について、ISO/IEC JTC 1/SC 27/WG 2 で進められてきた標準化内容を調査する。また、IETF Light-Weight Implementation Guidance(lwig) で行われている軽量暗号の実装に関するガイダンスについてまとめる。

3.4.3.2 ISO/IEC JTC 1/SC 27/WG 2 の活動

ISO/IEC 29192 は、チップサイズ、ハードウェアの消費電力、ソフトウェアのコードサイズ、RAM サイズ、伝送容量、実行時間などの制限がある場合の、データ秘匿、認証、本人識別、否認防止、鍵交換などを目的に適した軽量暗号の仕様を標準化している。ISO/IEC JTC 1/SC 27/WG 2 では技術カテゴリに対応し以下の 4 つのパートに分かれてこの標準化作業が行われた。これらの標準化作業はすべて 2013 年までに終了し、各パートの内容が ISO/IEC 29192-1，-2，-3，-4 として標準化されている。

■パート 1：総論　安全性要件、ハードウェア／ソフトウェア実装要件が規定された。

■安全性要件　80 ビットセキュリティ以上

■ハードウェア実装要件　ハードウェアチップ面積、実行サイクル数、１サイクル当たりの処理ビット数、消費電力、消費電力量、1 ビット当たりの消費電力量、実装に用いられたルールが方式比較のための参考情報とされた。但し、これらの具体的な数値はアプリケーション依存であるため、規定しない。

■ソフトウェア実装要件　プログラムコードサイズ、RAM サイズ、実行速度が方式比較のための参考情報とされた。但し、これらの具体的な数値はアプリケーション依存であるため、規定しない。

■他の特性　　軽量暗号は、短い平文、暗号文に対する処理も重要な要素となる。可能であれば、その特性が示されるべきである。また、実装に伴う遅延についても重要な要素となる。

■パート２：ブロック暗号　２つのブロック暗号、PRESENT と CLEFIA が標準化された。

- PRESENT　ブロックサイズ 64 ビット、鍵サイズ　80、128 ビット
- CLEFIA　　ブロックサイズ 128 ビット、鍵サイズ　128、192、256 ビット

■パート３：ストリーム暗号　２つのストリーム暗号、Enocoro と Trivium が標準化された。

- Enocoro　鍵サイズ 80、128 ビット
- Trivium　鍵サイズ 80 ビット

■パート４：公開鍵暗号（非対称暗号）技術を用いたメカニズム　公開鍵暗号技術を用いた、楕円上の離散対数問題をベースにした認証方式（cryptGPS）と、公開鍵暗号をベースにしたセッション鍵生成・鍵交換方式 (ALIKE) と、Identity ベース署名方式の３つが標準化された。

3.4.3.3 IETF における軽量暗号の実装に関するガイダンス

建物、車、電化製品などで使われている多くのデバイスでコミュニケーションができるようになってきた。但し、これらのデバイスの能力は様々であり、能力の小さいデバイスもある。IETF Light-Weight Implementation Guidance (lwig) では、このような小さい能力のデバイスに焦点をあて、非常に制限された環境下で、最小限の IP 接続を可能とする軽量暗号の実装方法、について標準化することを目的とする [16]。現在、lwig で議論が開始された段階であり、インターネット上での鍵交換関連 [17]、モーバイルネットワークでの低消費電力デバイス関連 [18]、TLS のカスタマイズ関連 [19] などの寄与文書が WG に提出されてきているが、まだ RFC 化されたものはない。

参考文献

[1] PRWeb, “ IEC and ISO adopt lower power encryption standard Enocoro stream cipher, ”http://www.prweb.com/releases/2012/11/prweb10132688.htm

[2] M. B. Abdelhalim, M.El-Mahallawy, and A. Elhennawy, “ Design & Implementation of an Encryption Algorithm for use in RFID System, ” International Journal of RFID Security and Cryptography, Vol.1, Issues1-4, Mar-Dec. 2012.

[3] TechRepublic, “ Is wireless RFID sensor authentication/ encryption possible? Maybe. ” http://www.techrepublic.com/blog/it-security/is-wireless-rfid-sensor-authentication-encryption-possible-maybe/

[4] TEXAS INSTRUMENT, “ Ultra-low Power Microcontrollers for Portable Medical Device Designs, ” http://www.engineering.com/ElectronicsDesign/ElectronicsDesignArticles/ArticleID/6222/Ultra-low-Power-Microcontrollers-for-Portable-Medical-Device-Designs.aspx

[5] F. H. Qi Hao, and M. Lukowiak, “ Implantable Medical Device Communication Security: Pattern vs. Signal Encryption (Position Paper), ” https://www.usenix.org/legacy/evnet/healthsec11/tech/final_files/hu-healthsec11.pdf

[6] ITS Japan,「ITS とは」, http://www.its-jp.org/about/

[7] IPA,「2012 年度自動車の情報セキュリティ動向に関する調査」, http://www.ipa.go.jp/files/000027274.pdf

[8] EVITA,“E-safety vehicle intrusion protected applications,” http://www.evita-project.org/

[9] PRESERVE,“PRESERVE preparing secure v2x communication systems,” http://www.evita-project.org/

[10] SAE,“Vehicle Electrical System Security Committee,” http://www.sae.org/works/committeeHome.do?comtID=TEVEES18

[11] 野島，盛合，「『シェア暗号』を自動車に」, http://techon.nikkeibp.co.jp/article/COLUMN/20140401/343501/?rt=nocnt

[12] The TECH REPORT, “ SandForce Improves SSD encryption, power management, ” http://techreport.com/news/24894/sandforce-improves-ssd-encryption-power-management

[13] 松井,「情報セキュリティ基盤技術暗号技術の最新動向- Cryptography:Technology and Applications -」, http://hiroshi1.hongo.wide.ad.jp/hiroshi/files/toku1/material/Matsui_Mitsubishi.pdf

[14] Axel Poschmann, “ Lightweight Cryptography, ” http://mathsci.ucd.ie/~gmg/ECC2007Talks/poschmann_LWC.pdf

[15] 鈴木，菅原，佐伯,「軽量／低遅延暗号のハードウェア実装性能について」, SCIS2014, 2A2-2

[16] IETF, “ Light-Weight Implementation Guidance (lwig), ” https://ietf.org/wg/lwig/charter/

[17] T. Kivinen, “ Minimal IKEv2 draft-ietf-iwig-ikev2-minimal-01 ”, https://ietf.org/doc/draft-ietf-lwig-ikev2-minimal/

[18] J. Arkko, A. Eriksson, and A. Keranen, “ Minimal IKEv2 draft-ietf-iwig-ikev2-minimal-01, ”https://ietf.org/doc/draft-ietf-lwig-cellular/

[19] S. S. Kumar, S. Keoh, and H. Tschofenig, “ Minimal IKEv2 draft-ietf-iwig-ikev2-minimal-01, ” https://ietf.org/doc/draft-ietf-lwig-tls-minimal/

第 4 章

軽量暗号のアプリケーションに関するヒアリング

2013 年度第 2 回軽量暗号 WG にて、エンドユーザーからのヒアリングとして、下記の 2 名の方から自動車および社会インフラへの軽量暗号技術の応用について意見を伺った。

- 「自動車における IT セキュリティ」 (トヨタ IT 開発センター 小熊 寿氏)
- 「制御システム向け暗号の要件の考察」 (日立製作所 大和田 徹氏)

小熊氏からは、自動車における IT セキュリティでは、例えば、車載ネットワーク CAN のデータ長が 8 バイトであることから、軽量暗号は、MAC を生成するアルゴリズムとして処理性能や MAC サイズの点で AES よりも有利と思われるとのコメントがあった。

また、大和田氏からは、課題からみた制御システム向け暗号の要件が抽出され、高速処理、低処理負荷、柔軟な暗号化対象長、低リソースでの鍵管理・更新機能等の要件で軽量暗号が役立つ可能性があるとコメントがあった。

2013 年度第 2 回軽量暗号 WG での発表資料を、参考資料として本報告書の A.1 章に掲載している。

第 5 章

軽量ブロック暗号の実装詳細評価

第 2 章で行った現状調査にも軽量暗号の実装評価は含まれるが、既存文献の調査であることから、文献により評価環境や実装者が異なるため、暗号アルゴリズム間の比較が困難であった。そこで、(独) 情報通信研究機構にて、軽量ブロック暗号 (AES, Camellia, CLEFIA, PRESENT, LED, Piccolo, TWINE, PRINCE) について、同一プラットフォーム上で、同一の実装者または統一的な実装ポリシーによりハードウェア実装およびソフトウェア実装の評価を行い、統一的な評価環境で比較調査を実施した。この評価結果が 2013 年度第 3 回軽量暗号 WG にて報告された。

実装環境および測定指標は下記の通りである。

■ハードウェア実装評価

- 標準的な CMOS セルライブラリ：NANGATE Open Cell Library (45nm CMOS)
- unrolled 実装, round 実装, serial 実装の 3 通りのアーキテクチャ
- 測定指標：最大動作周波数、処理速度、ゲートカウント、サイクルカウント、消費電力、ピーク電流

■ソフトウェア実装評価

- プロセッサ：ルネサスエレクトロニクス RL78 (16bit 組み込みマイコン)
- 測定指標：処理速度, RAM サイズ, ROM サイズ

ROM, RAM サイズに関して下記 4 通りの組み合わせで、それぞれの範囲内で処理速度を最大化する実装を行った。

ROM	512 バイト	1024 バイト
RAM	64 バイト	128 バイト

■評価結果概要　このハードウェア実装評価では、軽量暗号は AES と比較して 1-2kgate 回路規模が小さく、この違いはマチュアなプロセス (180nm-350nm) において実装の可否に影響する場合があり、アドバンテージとなること、リアルタイムのメモリ暗号化やμ秒クラスのリアルタイム通信などのアプリケーションにおいて優位となる可能性があることが報告された。また、小さい、速いという一つの指標だけだと AES との差分が少ないが、小さく、速く、サイドチャネル対策が容易という複数の軸で比較したときに AES に対する優位性がより明確になると報告された。

ソフトウェア（組み込みマイコン）実装においては、コードサイズの小さい暗号への要求が高い。メモリが十分あれば（例えば、アルゴリズム単体で暗号復号込みで ROM 1K バイトあれば）AES で十分である。よって組み込みマイコンにおいて AES より価値ある軽量ブロック暗号は、暗号・復号込みで ROM 200 バイト以下、RAM 32 バイト以下でそれなりの速度が達成できるアルゴリズムと考えられるという報告があった。

2013 年度第 3 回軽量暗号 WG での発表資料を、参考資料として本報告書の A.2 章に掲載している。

付録 A

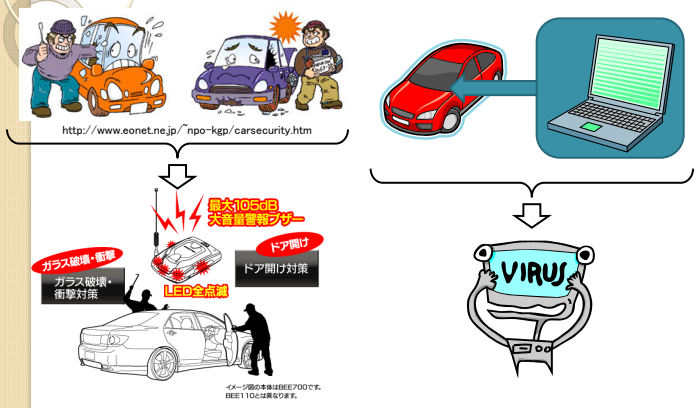
参考資料

A.1　軽量暗号のアプリケーションに関するヒアリング

A.1.1　自動車における IT セキュリティ

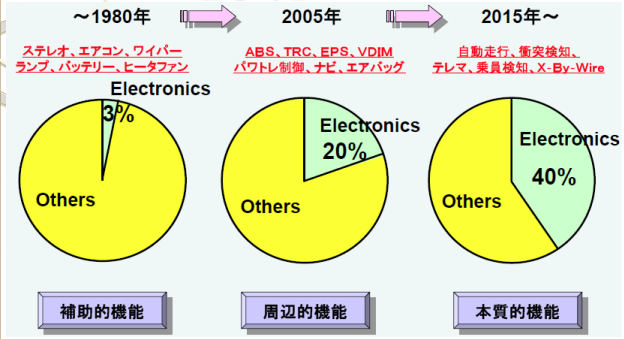
2013 年度第 2 回軽量暗号 WG(2013 年 12 月 26 日) でのトヨタ IT 開発センター 小熊 寿氏による発表資料を示す。

「クルマ」×「セキュリティ」



http://www.kato-denki.com/products/bee/bee110/img/standard_sensor.png

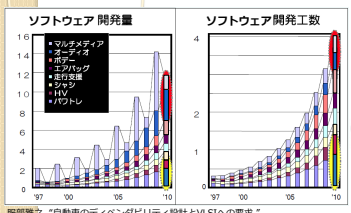
クルマの電子化

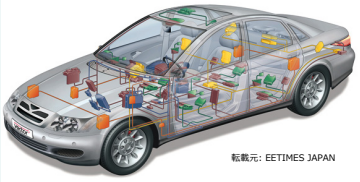


服部雅之, "自動車のディペンダビリティ設計とVLSIへの要求,"
JST CREST「ディペンダブルVLSIシステムの基盤技術」研究領域 平成19年度ワークショップ

- 1970年代の排ガス規制
  - エンジン状態を監視し、燃料噴射量と点火タイミングを制御
  - 電子部品導入のきっかけ

現在のクルマ





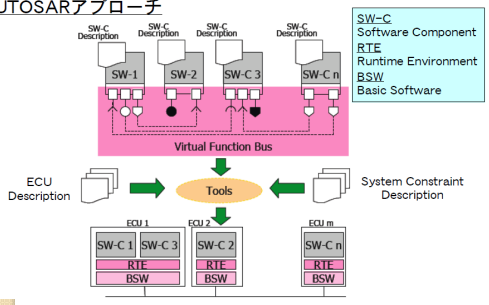
転載元: EETIMES JAPAN

服部雅之, "自動車のディペンダビリティ設計とVLSIへの要求,"
JST CREST「ディペンダブルVLSIシステムの基盤技術」研究領域 平成19年度ワークショップ

- 高級車の事例: LEXUS LS460
  - 100個以上の車載制御マイコン (ECU: Electronic Control Unit)
  - SWの総ライン数はカーナビなどを含めると17M行: 雑誌報道による数値
    - 2015年には100M行という予測も...
- ECUが連携してサービスを提供
  - e.g. 車体姿勢制御、プリクラッシュセーフティ
  - SWの大規模化と複雑化↑

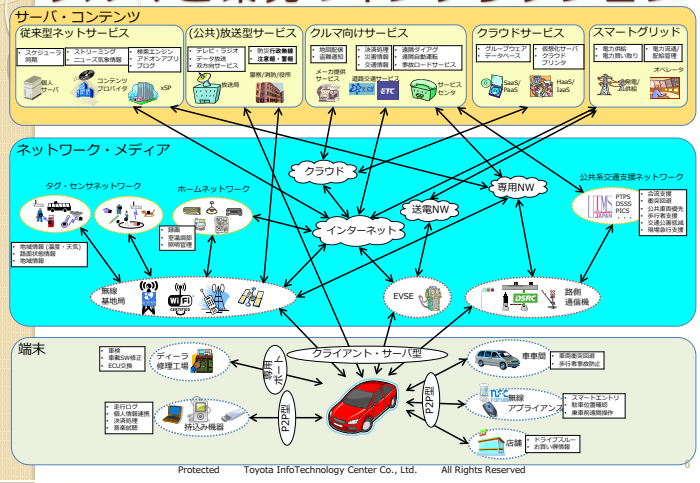
プラットフォームベース開発

AUTOSARアプローチ



- 基盤機能共通化による開発工数およびコストの削減
  - プラットフォームベースの開発へ移行
- AUTOSAR (Automobive Open System Architecture): 2003～
  - SW基盤の業界標準を作成
- JasPar (Japan Automotive Software Platform): 2004～
  - 国内メーカの要望を集約し、AUTOSARへのインプットなどを行う

クルマと環境のインタラクション



標準化とコンピュータ

- 攻撃者がクラック出来る環境
  - 仕様書が閲覧可能であり、セキュリティホールも確認出来る
- 全く同じ事象が起きる環境
  - 1台さえ解析できれば、他も同じアーキテクチャであるため同じ手法が適用可能
- 情報伝搬速度が早い
  - ほとんどのPCがネットワーク接続しているため、瞬時に広範囲に影響する

- アーキテクチャの変化：「機種毎」⇒「みんな同じ」
- 仕様書の公開

- 外部からの侵入口
- 情報発信

1980年代後半
機能の標準化

1990年代前半
インターネット接続

メインフレーム

PC/AT互換機

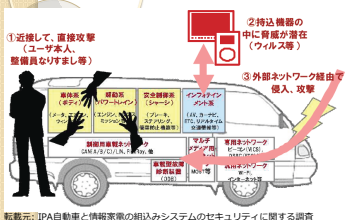
ワークステーション

- 「標準化」と「NW接続」する将来のクルマ
  - 同じサイクルにはまる可能性大

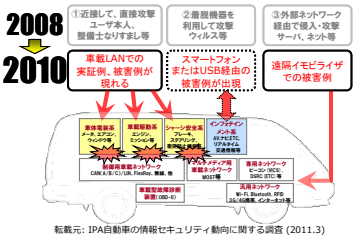
クルマの要求条件

- ハードリアルタイムとFail-Safe
  - 生命に直結するため時間制約が厳しい (即時応答性)
  - 不具合が起きたときに安全側に倒れる事
- 10年以上の耐用年数
  - 製造から廃車までの時間が長く、中古車市場にも転用
- 不具合発生を事前に防止
  - PC: ウイルス感染などの被害が現れてからの対応が多い (セキュリティSWメーカによる事前調査もある)
  - クルマ: 事故など具体的な被害が出る前に対応する必要あり
- 切断時動作
  - NW接続はモバイル機器と同様に無線: 生命に直結するサービスを常時接続前提で考えてはいけない
- 劣悪な環境での動作、信頼性
  - 電圧変動±50% 動作環境温度−40～140℃

2010年に潮目が変化



転載元: IPA自動車と情報家電の組込みシステムのセキュリティに関する調査 (2009.3)



転載元: IPA自動車の情報セキュリティ動向に関する調査 (2011.3)

- 想定される脅威の列挙
  - 「机上の空論」: ～のでは?
  - 想定ベースの議論であり、説得性に欠ける: Evidence不足
- 攻撃結果の公開
  - 予想した脅威が現実化
  - 事例として対外発表が行われる
- 具体的な事故は未報告
  - 想定外の事象が起こる可能性は否定できない

周辺動向

- 海外
  - 国プロ関連
    - 欧州によるFramework Program: 製品化を見据えた研究・標準化活動
      - SeVeCom (FP6): 路車間・車車間通信のセキュリティ
      - EVITA (FP7): 車載システムのセキュリティ
  - 学会
    - Escar: Embedded Security in Cars Conference
  - 産業界
    - SAE Vehicle Electrical System Security Committee (2011～): NHTSAによるCybersecurity Researchと連携
- 国内
  - 情報処理推進機構による研究会
    - 2006年、2008年～2011年: クルマ向け情報セキュリティ動向と想定される脅威を調査
  - 産業界
    - 自動車技術会による情報セキュリティ小委員会 (2010～)

escar: Embedded Security in Cars Conference

- 2003年から毎年ドイツ国内で開催
  - 2003年は20名程度、2008年からはCFP

	2009年	2010年	2011年	2012年	2013年
参加者	54	72	74	112	110
うち日本人	1	3	5	7	16

*参加者リストを参照

- 2013年からはUS、2014年からはアジアでも実施
  - 次回escar USAは7月、CFPの予定
  - 1st escar ASIAは4月

Trusted Assurance Levels

- C2C-CCにて検討中

TAL	Security Evaluation Requirements			Resulting Security Implications		
	Minimum Target of Evaluation (TOE)	Minimum Evaluation Assurance Level (EAL)	Minimum (Hardware) Security Functionality	Prevented (Internal) Attacker acc. to CC	Potential Security Implications	V2X Use Case Examples
0	None	None	None	None	Not reliable against security attacks	Some limited (e.g., using trusted V2I infrastructures)
1	+ V2X software	EAL 3	Only software security mechanisms	Basic	Not reliable against simple hardware attacks (e.g., offline flash manipulation)	Non-safety, but most privacy relevant use cases
2	+ V2X hardware	EAL 4	+ dedicated hardware security (i.e., secure memory & processing) + tamper evidence	Enhanced Basic	Not reliable against more sophisticated hardware attacks (e.g., side-channel attacks)	V2X day one use cases (e.g., passive warnings and helpers)
3	+ Private ECU & private network	EAL 4+ (AVA_VAN.4 vulnerability resistance)	+ basic tamper resistance	Moderate	V2X box secure as stand alone device, but w/o trustworthy in-vehicle inputs	Safety relevant relying not only on V2X inputs
4	+ Relevant in-vehicle sensors and ECUs	EAL 4+ (AVA_VAN.5 vulnerability resistance)	+ moderate – high tamper resistance	Moderate – High	V2X box is trustworthy also regarding all relevant in-vehicle inputs	All

- Marko Wolf, – Hardware Security Modules for Protecting Automotive IT Systems – The EVITA project and beyond, escar USA 2013

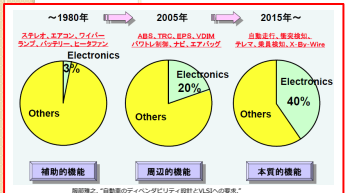
Vehicle Electrical System Security Committee

- SAEにて2011年から活動開始
  - 車載システムへの攻撃に関する研究発表などがトリガ
- 2つのタスクフォース
  - Automotive Security Guidelines and Risk Development
    - プロセスベースで車載システムのセキュリティレベルを策定
    - リスクを軽減するためのガイドラインおよび推奨デザインを作成
    - 2014年夏に初版リリースを目指す
  - Vehicle Electrical Hardware Security
    - ハードウェアベースのセキュリティ技術を利用
    - 「車載システムのセキュリティ」担保のための推奨デザインを作成
    - 2014年1月に作業完了予定

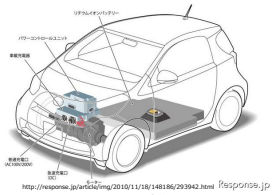
Battelle CyberAuto Challenge

- 自動車を中心とした交通システムに対する脅威と防御について学習
  - 高校/大学生向けサマーキャンプ: 1週間程度
  - USビッグ3やUS政府関係者などによる実地サポート
  - 実際に自動車をクラック
- 今年が第2回目

今後のクルマ



高度サービス・コスト削減のための電子化


電子部品比率が増えるEV

- 「計算機科学的アプローチ」によるセキュリティ技術が重要かつ必須
- 「EVならでは」も存在
  - EVは高トルク: エンジンチューンナップによる想定外の動き
  - 安価なサードパーティ製バッテリ: ただでさえEVは走行距離が短く、販売価格の30%を占めるバッテリ

まとめ

- クルマへの情報セキュリティ技術の必要性
  - 要因: 電子化比率の増加、標準化、EV化
- 気をつけること
  - ヒトの命を預かる耐久消費品
- 国内技術者の絶対的な不足

A.1.2　制御システム向け暗号の要件の考察

2013 年度第 2 回軽量暗号 WG(2013 年 12 月 26 日) での日立製作所 大和田 徹氏による発表資料を示す。

HITACHI
Inspire the Next

CRYPTREC 軽量暗号WG

制御システム向け暗号の要件の考察

2013/12/26

(株)日立製作所 横浜研究所
大和田 徹

1　IT分野における情報セキュリティ

HITACHI
Inspire the Next

情報システムに対する様々なセキュリティ上の脅威が存在

■ 情報システムにおける主な脅威

- 情報漏えい、盗聴、なりすまし、不正アクセス
- データ/プログラムの改ざん、ウィルス感染
- DoS攻撃によるネットワーク/サーバのダウン、データ/プログラムの削除

■ 情報セキュリティ:
情報資産(保護資産)の定義 + 当該資産に対する3要素の維持

要素	定義	代表的な対策技術
機密性	認可されていない個人、エンティティまたはプロセスに対して、情報を使用不可または非公開にする特性	暗号、認証、アクセス制御
完全性	資産の正確さ、および完全さを保護する特性	改ざん検知、ログ管理、バックアップ
可用性	許可されたエンティティが要求したときに、アクセスおよび使用が可能である特性	ファイアウォール、リソースの多重化

**情報システムでは各種対策技術を
組合せて情報資産のセキュリティを確保**

2 制御システム分野における情報セキュリティ優先事項

HITACHI Inspire the Next

制御システムにおける重要な保護資産とは

■ 制御システムと情報システムの比較(*)

比較項目	制御システム	情報システム
データ処理制約	リアルタイムかつ周期的な制御処理 ⇒遅延により制御不全に陥る可能性	処理集中による遅延は、 ある程度許容
システム更新頻度	10-20年	3-5年
稼動時間	24時間365日連続	一般には通常業務時間内
想定被害	システム不具合による人命影響の 可能性	金銭的損失、 プライバシー被害
優先保護資産	システムの連続稼動(可用性)	情報資産の漏洩防止(機密性)

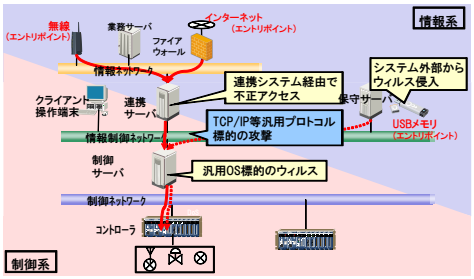
セキュリティ上の脅威に晒されても
制御システムが連続稼動すること(可用性)が最重要

＊IPA「重要インフラの制御システムセキュリティとITサービス継続に関する調査」を元に作成
http://www.ipa.go.jp/security/fy20/reports/ics-sec/

3 制御システムの一般的構成と脅威例

HITACHI Inspire the Next

昨今の制御システム～IT + CTのハイブリッド構成



物理アクセス性はシステム
によりまちまち
発電所、プラント～ビル
～スマートメータ

情報系部分は既存手法によるサイバー攻撃の対象となり得る
制御系部分への侵入で制御不全に

4 攻撃の顕在化と、それに対抗する動き

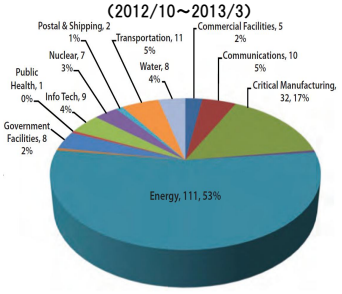
HITACHI Inspire the Next

制御システムへのサイバー攻撃が顕在化(12/下 200以上の報告),
制御システムに対するセキュリティ強化の要求が高まる

- WIB等, 業界レベルのセキュリティ規格策定段階から
  EDSA認証(CSSC)・CSMS認証(JIPDEC)等の
  国際的なセキュリティ認証制度整備段階へ進展
- ICS-CERTによる脆弱性公開が加速

コンポーネント/システムの
セキュリティ設計, セキュリティ運用
の重要性増大

重要インフラにおけるセキュリティ事故報告
(2012/10～2013/3)

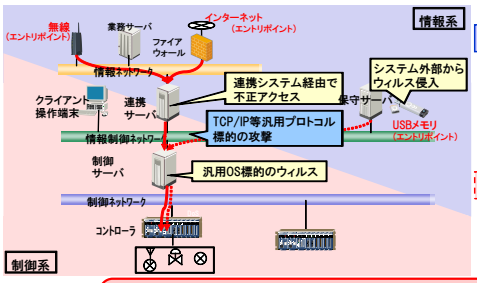


出典：米ICS-CERT,
http://ics-cert.us-cert.gov/sites/default/files/ICS-CERT_Monitor_April-June2013.pdf

5 制御システムにおけるセキュリティ対策

HITACHI Inspire the Next

昨今の制御システム～IT + CTのハイブリッド構成
ITベース攻撃手法の対象となり得る



対抗策
3要素を組合せた最新の情報系セキュリティ対策技術

（大） （汎用・標準）
外部接続 採用技術
（小） （専用・独自）

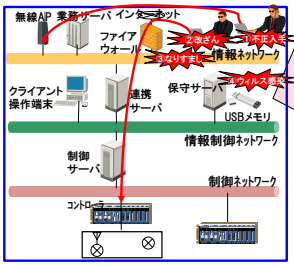
対抗策
制御系の特徴を考慮したセキュリティ対策技術

制御系の特徴を考慮した対策技術の確立が急務

6 制御システムの可用性確保に向けた対策技術

HITACHI Inspire the Next

制御システムの可用性を損なう主要な脅威の洗い出し



可用性を損なう主要な脅威とその対策方針

#	脅威	対策方針
①	第三者がシステム内を流れるデータを不正に入手し、誤作動を引き起こすために悪用する(不正パケットの送信など)	パケットの内容が理解できないようにする
②	第三者がコントローラ宛のデータを改ざんし、コントローラが誤った制御指令をセンサに送信してしまう(誤動作、機能低下)	不正パケットを検知する
③	第三者が保守サーバになりすまして制御サーバを乗っ取り、コンローラの誤動作を引き起こす	通信相手が正しいことを確認する
④	保守員が保守サーバにアクセスする際、保守サーバに接続する可搬型記憶媒体(USBなど)がウィルスに感染していた為、ウィルスがシステム内に侵入し、プログラムやデータの改ざん/破壊、DoS攻撃を行う(システムを停止するため)	ウィルスを検知する

暗号技術が可用性確保に繋がる分野が存在

7 暗号適用時の課題

HITACHI Inspire the Next

制御系への暗号適用影響を分析し、課題を抽出

#	制御系の制約	暗号適用時に想定される影響	暗号適用時の課題	
1	周期的/リアルタイム処理	負荷増加により、周期的/リアルタイム処理の困難化	レイテンシ制約	性能
2	組込み機器のリソース制約(CPU性能、メモリ容量等)	高処理負荷暗号の実行によるシステム通常動作の困難化	暗号による処理負荷大	性能
3	全てがクリティカルな制御情報とは限らない	制御データ全てを暗号化対象とすると処理負荷大	不必要な暗号処理による不要な処理負荷の発生	性能/安全性
4	連続稼動	鍵管理/鍵更新処理がシステムの連続稼動に影響	連続稼動に影響しない鍵管理/鍵更新方式が確立されていない	運用
5	長期運用(10-20年)	暗号危殆化の可能性	暗号危殆化による安全性低下	運用

上記課題を解決する暗号機能の実現が求められている

＊ JPCERT/CC「重要社会インフラのためのプロセス制御システムのセキュリティ強化ガイド」を元に作成
www.jpcert.or.jp/research/2009/PCSSecGuide_20091120.pdf

8 制御システム向け暗号の要件

HITACHI
Inspire the Next

課題から制御システム向け暗号の要件を抽出

#	【再掲】課題/課題の区分		制御システム向け暗号の要件
1	レイテンシ制約	性能	高速処理可能
2	暗号による処理負荷大	性能	低処理負荷
3	不必要な暗号処理による不要な処理負荷の発生	性能/安全性	柔軟な暗号化対象長
4	連続稼動に影響しない鍵管理/鍵更新方式が確立されていない	運用	低リソースな鍵管理/鍵更新機能
5	暗号危殆化による安全性低下	運用	複数鍵長/アルゴリズムの切替容易

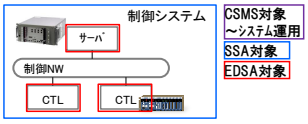
制御システム向けに上記要件を満たす「軽量暗号」が役立つ可能性はある

9 制御コンポーネントのセキュリティ認証

HITACHI
Inspire the Next

動向

- IEC62443に則ったｾｷｭﾘﾃｨ評価認証ｽｷｰﾑ整備中
  - 事業・運用者向けCSMS認証(→IEC62443-2)
  - システムベンダ向けCSS認証(→IEC62443-3)
  - **装置ベンダ向けEDSA認証(→IEC62443-4)**
    - 将来的な調達要件化が想定
    - 総構造認証～システム認証は認証機器が前提



- CSSCがEDSA国際相互認証ｽｷｰﾑを準備中
  EDSA認証取得機器：2社4製品(‘13/12時点)

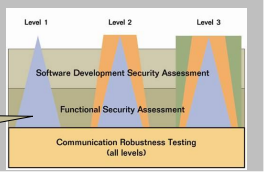
レベル1でのFSA要求は主にユーザ認証
アクセスコントロール、データ完全性、
機密性要求はレベル2以上で
→暗号化以前の課題が山積

EDSA認証

ISAセキュリティ適合性協会(ISCI)が運営する制御機器のセキュリティ保証に関する認証制度

評価項目としては
1. 機能セキュリティアセスメント(FSA)
2. ソフトウェア開発セキュリティアセスメント(SDSA)
3. 通信ロバストネステスト(CRT)

の3項目が存在。



出典：米ISA Secure
http://www.isasecure.org/ISASecure-Program.aspx

10 まとめ～制御システム向け暗号への期待

HITACHI
Inspire the Next

**実現したいシステムセキュリティからすると
暗号は数多くの要件のうちの一つ
～「グローバルで使い易い」暗号を期待**

- 暗号アルゴリズムだけ、では使いこなせない
  - 鍵管理手法等も含めたシステムパッケージを期待
  - 制御向けの暗号使い方ガイドを期待
- 制御コンポーネントは汎用技術活用の方向
  - 専用HWを必要としない組込みCPUに適した暗号を期待
- 評価認証スキームの確立と国際調達要件化
  - 国際標準でない暗号の採用困難化の方向→使える暗号の国際標準化推進を期待

A.2　軽量ブロック暗号の実装詳細評価

A.2.1　ハードウェア性能評価

2013 年度第 3 回軽量暗号 WG(2014 年 2 月 20 日) での三菱電機　鈴木 大輔氏による発表資料を示す。

MITSUBISHI
Changes for the Better

2014/02/11

資料2－1

軽量暗号の
ハードウェア実装性能

軽量暗号WG報告

＊三菱電機株式会社　情報技術総合研究所

三菱電機株式会社

MITSUBISHI
Changes for the Better

以下は、
実際に各種アルゴリズムを実装評価
した結果について述べる

- 同じプラットフォームで評価する（ライブラリで数Kgateはの差がでる）
- 同じ合成条件で比較する（制約で数Kgateの差がでる）
- 一般的な設計基準でRTLレベルで構成する
  （リセットを入れる、スキャンセルをつかわないなど）
- 目的はAESに対する性能比較　（軽量暗号間の性能差は議論しない）

MITSUBISHI Changes for the Better

機能概略

F1. 鍵長は規定される最小のモードを想定する.

F2. 暗号化のみの実装とする.
(一部暗号化・復号も実装したので報告する)

F3. CPU のコプロセッサとしての利用を想定し，コンパクトで低電力とされるAPB バス接続が可能な設計とする.

MITSUBISHI Changes for the Better

設計方針概略

P1. 各アルゴリズムに対して3 種類の実装を行う:

(i) 典型的なround ベースの実装
(ii) 1 サイクルで処理が完了するunrolled 実装
(iii) データパスをS-box のサイズとするserial 実装

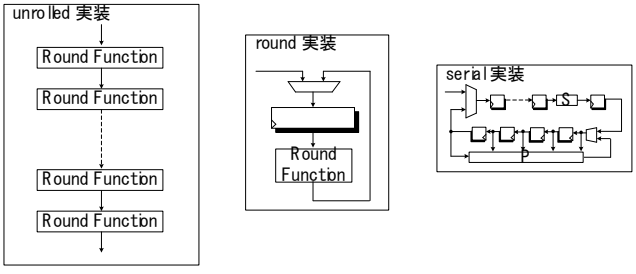
P2. 鍵スケジュールはon-the-yで実装する.

P3. CMOS セルライブラリを直接インスタンスするような最適化は行わず，ライブラリ非依存で合成可能な記述とする.

MITSUBISHI Changes for the Better

アーキテクチャ

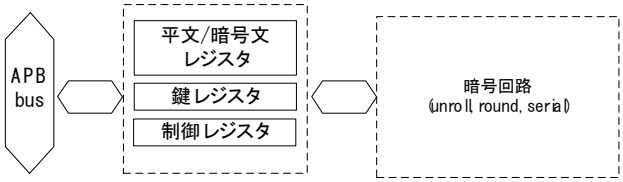
■実装方式



MITSUBISHI Changes for the Better

インターフェース

■インターフェース



MITSUBISHI Changes for the Better

評価環境

オープンセルライブラリと標準的なツールを使用.

論理合成ツール	Design Compiler (G-2012.06-SP5)
ライブラリ	NANGATE Open Cell Library(45nm CMOS)
合成制約	面積最小
遅延条件	NangateOpenCellLibrary slow (最悪条件の仮想遅延)
論理シミュレータ	NC-Verilog 10.20-s040
使用言語	Verilog-HDL

MITSUBISHI Changes for the Better

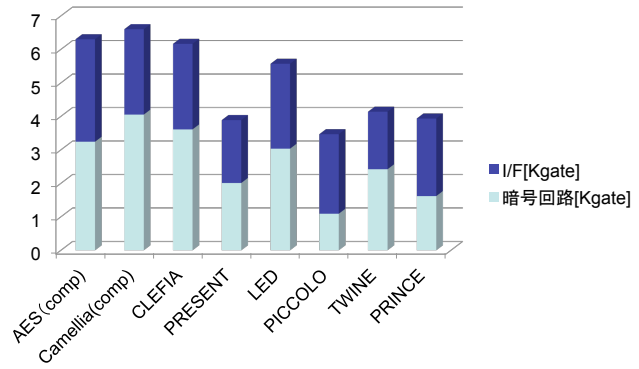
暗号化のみ

MITSUBISHI Changes for the Better

Serial

ゲートサイズ

AESに対して他のアルゴリズムは-2Kgateから+1Kgateの範囲

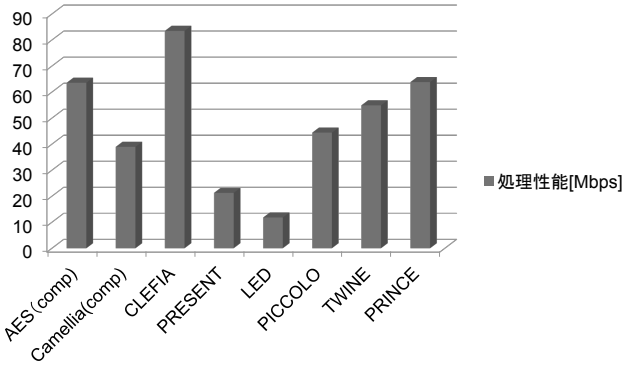


MITSUBISHI Changes for the Better

Serial

スループット

軽量暗号はSerial実装で高速化しにくい

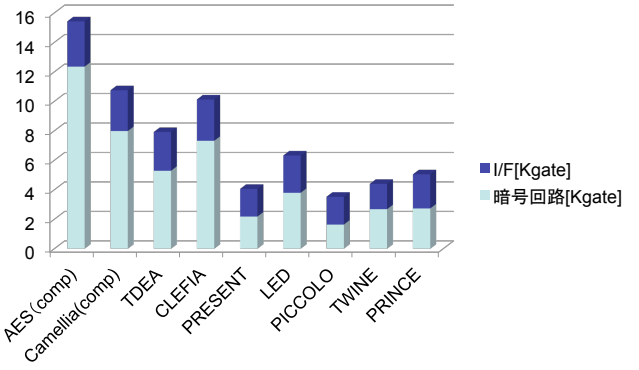


MITSUBISHI Changes for the Better

Round

ゲートサイズ

AESに対して軽量暗号はRound実装とSerial実装の回路規模に差がほとんどなく、4Kgate以下で実装できる

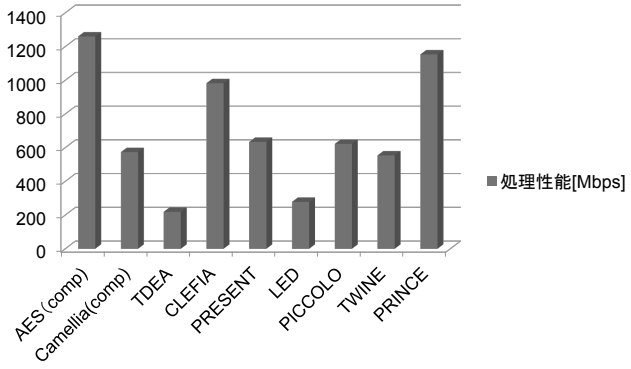


MITSUBISHI Changes for the Better

Round

スループット

軽量暗号はRound実装の軽量暗号はSerial実装のAESに対して
速度性能約10倍

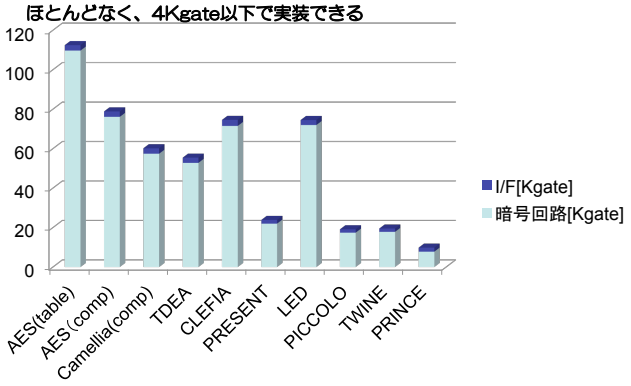


MITSUBISHI Changes for the Better

Unrolled

ゲートサイズ

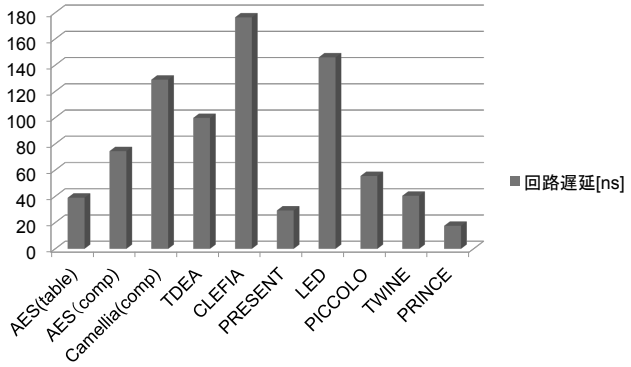
AESに対して軽量暗号はRound実装とSerial実装の回路規模に差が
ほとんどなく、4Kgate以下で実装できる



MITSUBISHI Changes for the Better

Unrolled

回路遅延



MITSUBISHI Changes for the Better

差分のまとめ

- 「論理回路性能の視点から」　軽量暗号とAESの違い

① 回路規模は１～2Kgate軽量暗号の方が小さい

② 約3Kgate以内でつくるなら軽量暗号の方がサイクル数が1/10

③ ②と同じサイクル数を達成するためにはAESは約10Kgate必要

④ 「1サイクル暗号化」に必要な回路規模が2ケタ違う。

⑤ 1サイクルとしてとれる周波数が2～3倍低遅延暗号が高速

MITSUBISHI Changes for the Better

RFIDをメインアプリと想定した場合

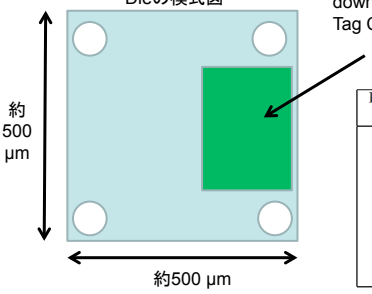
軽量暗号のハードウエア実装のアプリケーションと言えば「RFID」

この領域のアプリケーションでは処理時間は通信時間の方が支配的であることが多い

（②、③よりも）①の視点が産業上の有用性を見出せるか？

MITSUBISHI Changes for the Better

RFIDにおける実装制約



http://www.impinj.com/support/downloadable_documents.aspx#Monza 5 Tag Chips

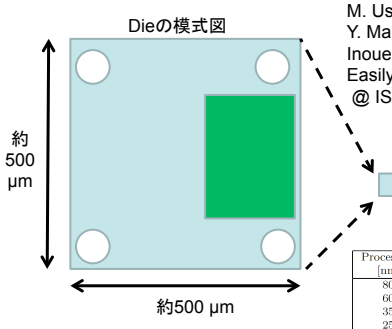
ロジック部は全体の3割程度

Process [nm]	Gate dencity [gate/mm^2]	500um 角の 30% 領域利用時 [gate]
800	1.2K	90
600	3K	225
350	18K	1.4K
250	35K	2.6K
180	80K	6K
130	180K	13.5K
90	400K	30K
65	800K	60K
40	1600K	120K

１～2Kgateの違いによって、実装の可否が変わるのは0.18 umくらいまで

MITSUBISHI Changes for the Better

RFIDにおける実装制約



M. Usami, H. Tanabe, A. Sato, I. Sakama, Y. Maki, T. Iwamatsu,T.Ipposhi and Y. Inoue:A 0.05 × 0.05 mm² RFID Chip with Easily Scaled-Down ID-Memory. @ ISSCC 2007

μ-chipくらいのRFIDであれば･･･

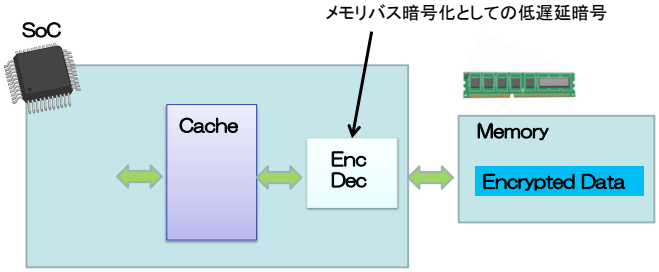
Process [nm]	Gate dencity [gate/mm²]	500um 角の 30% 領域利用時 [gate]	50um 角領域 利用時 [gate]
800	1.2K	90	3
600	3K	225	7.5
350	18K	1.4K	45
250	35K	2.6K	76
180	80K	6K	200
130	180K	13.5K	450
90	400K	30K	1K
65	800K	60K	2K
40	1600K	120K	4K

40nmでも1～2Kgateの差によって、実装の可否が変わる可能性あり

MITSUBISHI Changes for the Better

XOM, AEGIS

- 耐タンパプロセッサ
- 主記憶をOSや他のプロセス、あるいはプロービング攻撃などから秘匿することを目的として暗号化
- 読み出し速度がクリティカル



MITSUBISHI Changes for the Better

リアルタイム性能

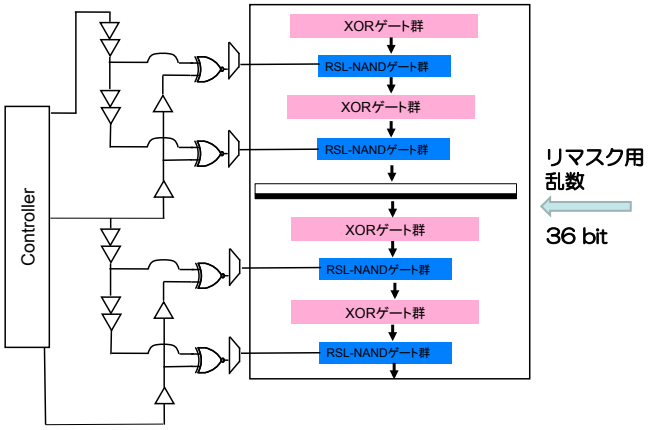
App.	Time region
Man-machine interface	数秒
Motor control	数 ms～数十 ms
I/O device control	数 μs
フラッシュ, EEPROM 読み出し	数 10～100ns
低電力 SRAM	数 10ns
高速 SRAM	～10ns

MITSUBISHI
Changes for the Better

＋　サイドチャネル対策

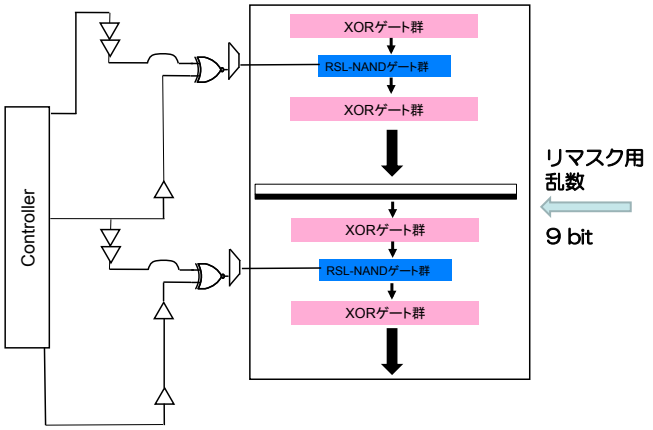
MITSUBISHI
Changes for the Better

8bit S-boxでのRSL



MITSUBISHI
Changes for the Better

4bit S-boxでのRSL



MITSUBISHI Changes for the Better

考察

■「軽量暗号」たる特徴は
- ✓ブロック長が64bit
- ✓鍵スケジュールが軽い（ラウンド定数のみ、レジスタ不要）
- ✓4bit S-box

これらがAESより1～2Kgate小さくなる主要因
(逆にいえば、これでほぼ特徴付けられる)

■ 改良は

暗号化のみ(PRESENT)
→復号もほぼ同じサイズでできる(Piccolo,TWINE)
→そもそも速い(低遅延 PRINCE)
→(認証暗号（FIDES）) という流れ？

MITSUBISHI Changes for the Better

まとめ

■ハードウェア実装からの視点としては軽量暗号は、
- マチュアなプロセスでの回路規模
- (リアルタイム)メモリ暗号化
- μ秒クラスのリアルタイム通信

などのアプリにおいて、AES に対してアドバンテージがある。

■小さい、速いという一つの指標だけだとAESとの差分が少ない。小さく、速く（低遅延)、サイドチャネル対策しやすい、のが良い軽量暗号、という考え方は？

■ファームウェア実装を考慮すれば、また違った視点
軽量暗号のアドバンテージが考えられる。

MITSUBISHI Changes for the Better

以下付録

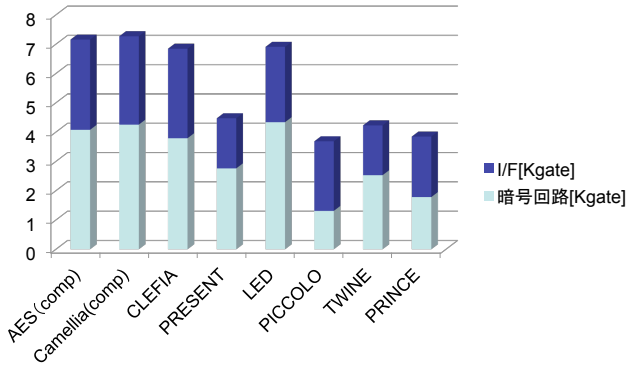
MITSUBISHI
Changes for the Better

暗号化・復号

MITSUBISHI
Changes for the Better

Serial

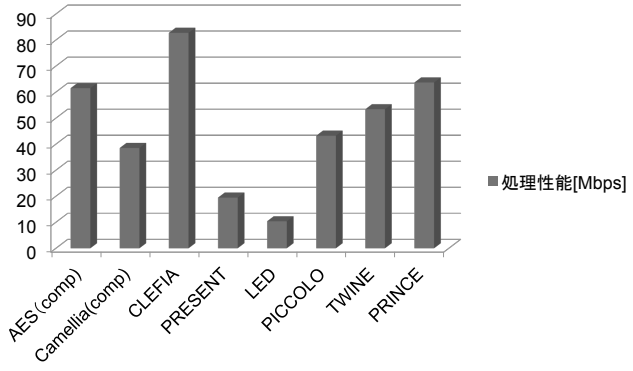
ゲートサイズ



MITSUBISHI
Changes for the Better

Serial

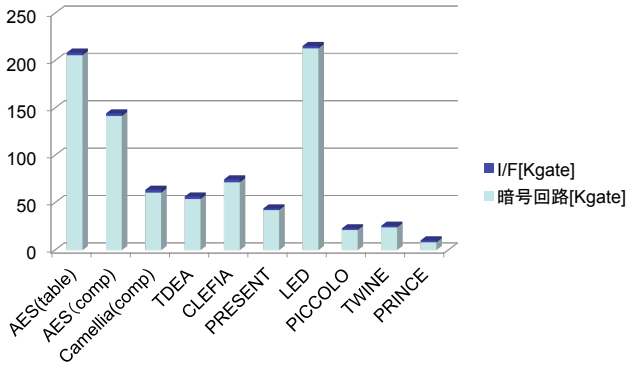
スループット



MITSUBISHI
Changes for the Better

Unrolled

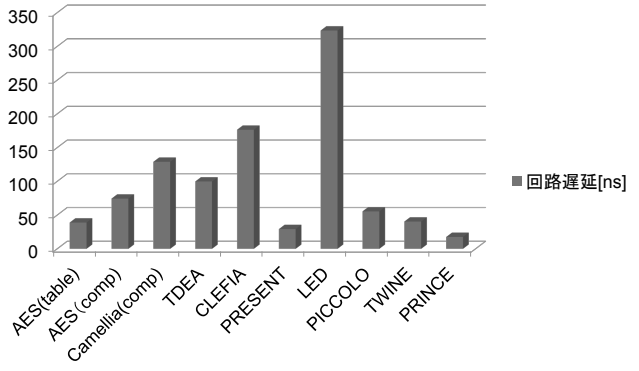
ゲートサイズ



MITSUBISHI
Changes for the Better

Unrolled

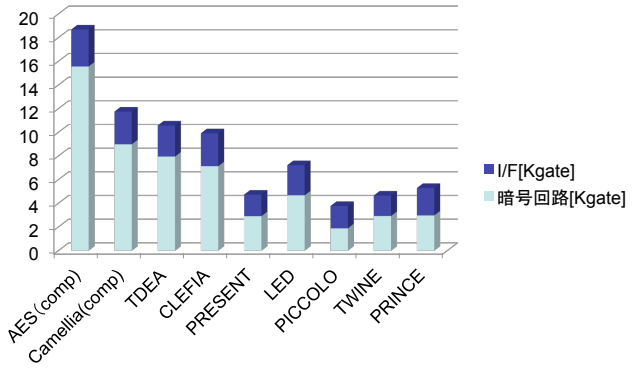
回路遅延



MITSUBISHI
Changes for the Better

Round

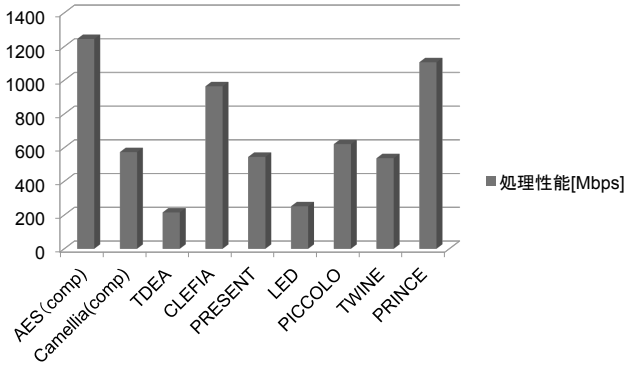
ゲートサイズ



MITSUBISHI Changes for the Better

Round

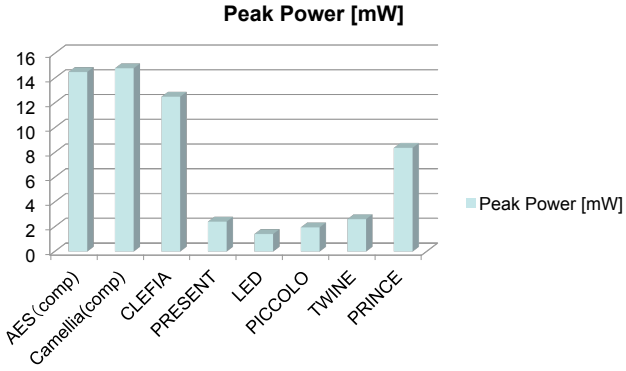
スループット



MITSUBISHI Changes for the Better

Serial

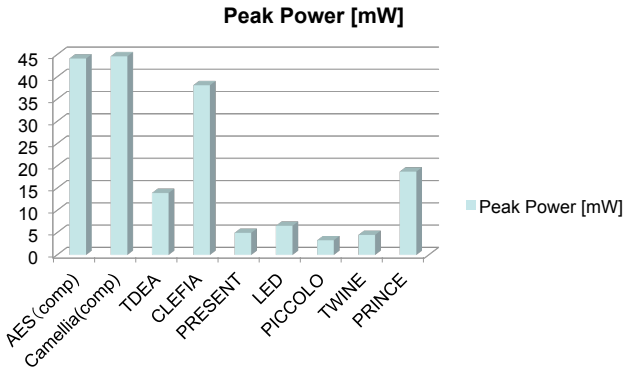
ピーク電流



MITSUBISHI Changes for the Better

Round

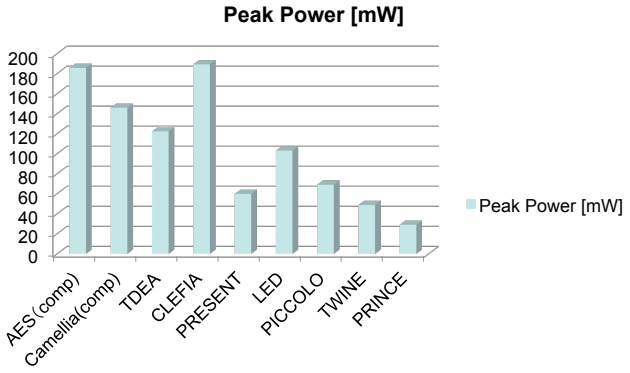
ピーク電流



MITSUBISHI Changes for the Better

Unrolled

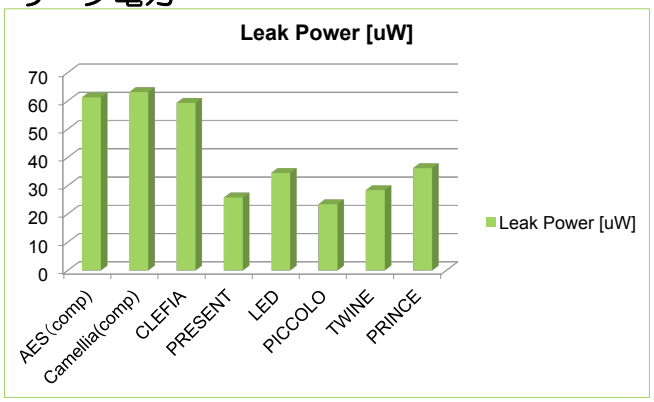
ピーク電流



MITSUBISHI Changes for the Better

Serial

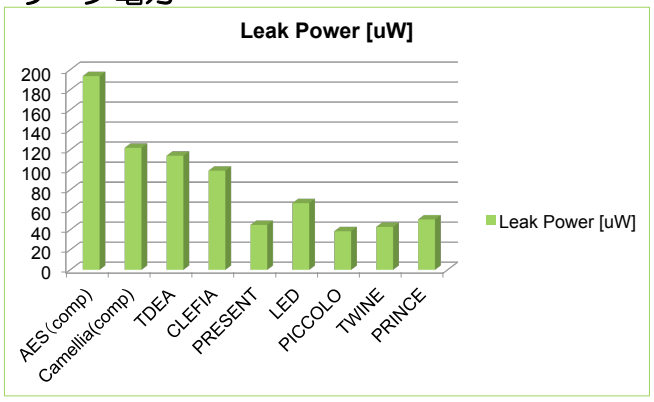
リーク電力



MITSUBISHI Changes for the Better

Round

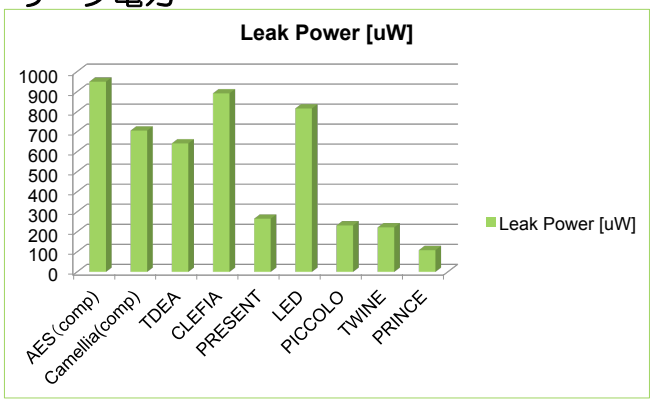
リーク電力



MITSUBISHI Changes for the Better

Unrolled

リーク電力



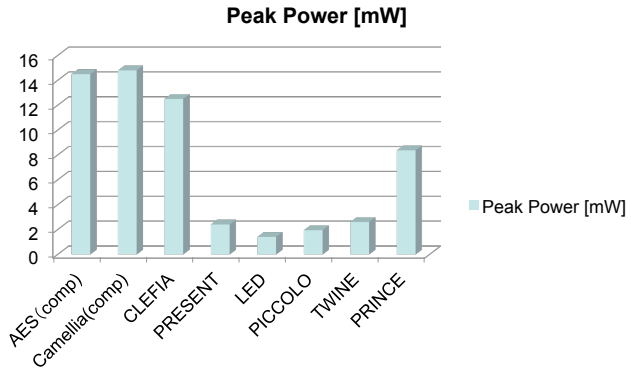
MITSUBISHI Changes for the Better

暗号化のみ(残りのデータ)

MITSUBISHI Changes for the Better

Serial

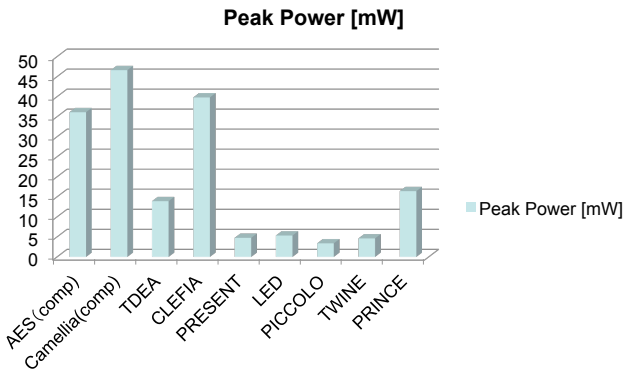
ピーク電流



MITSUBISHI
Changes for the Better

Round

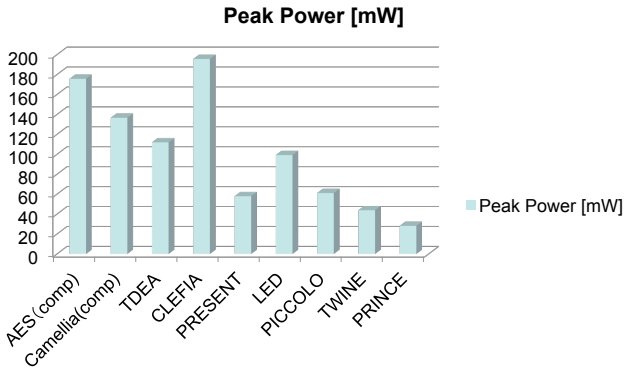
ピーク電流



MITSUBISHI
Changes for the Better

Unrolled

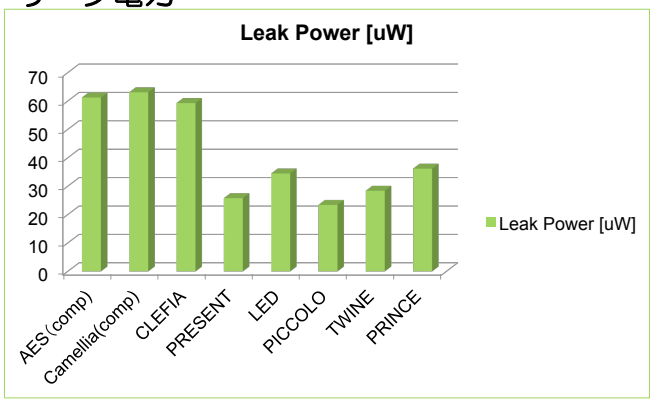
ピーク電流



MITSUBISHI
Changes for the Better

Serial

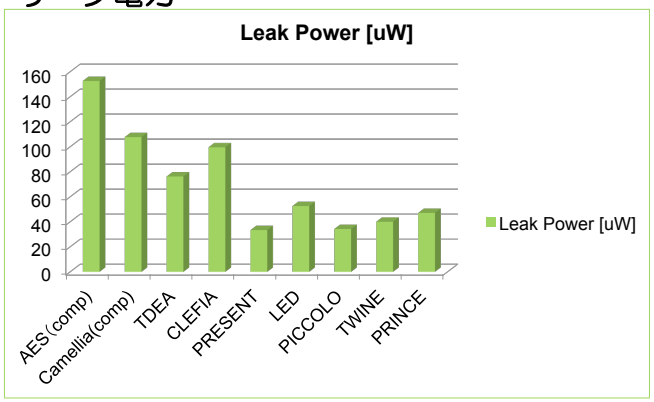
リーク電力



MITSUBISHI
Changes for the Better

Round

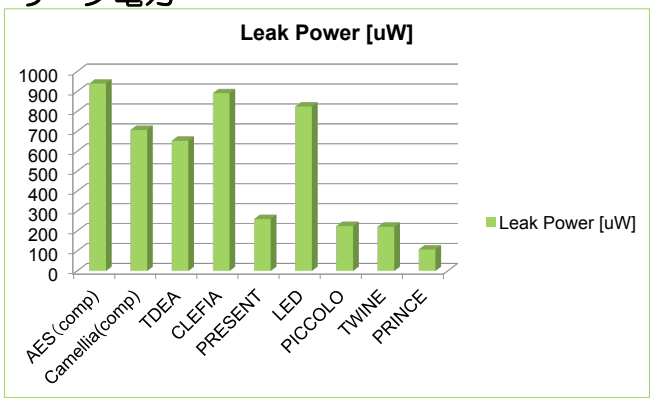
リーク電力



MITSUBISHI
Changes for the Better

Unrolled

リーク電力



A.2.2 ソフトウェア性能評価

2013 年度第 3 回軽量暗号 WG(2014 年 2 月 20 日) での三菱電機　松井 充氏による発表資料を示す。

資料2－2

軽量暗号のソフトウエア性能評価

(CRYPTREC軽量暗号WG資料)

2014年2月20日
三菱電機情報技術総合研究所　松井充

1

評価の目的

- Lightweight と呼ばれるブロック暗号がマイコン上のソフトウエアでどの程度 Lightweight であるかを調べる
  - Lightweight はハードウエアで語られることが多い
- マイコン上で小型を目指した暗号実装評価結果は数少ない
  - ソフトウエアでの暗号評価はほとんどの場合高速化が目標
  - しかも評価方法のコンセンサスがない
- 評価方法を提案
  - 実用的な観点からのインターフェースと評価方法を定義
  - ROM, RAM サイズを指定して、その範囲内で実装し速度を計測
  - 速度は無視してROMサイズがどこまで小さくなるかもみる

2

どの程度小さければLightweightか

- Renesas社マイコンRL78の場合
  - 汎用品（G1xシリーズ）： ROM 1KB, RAM 128B から
  - 車載品（F1xシリーズ）： ROM 8KB, RAM 512B から
- Atmel社マイコンAVRの場合
  - ATtiny: ROM 0.5KB, RAM 32B から ROM 16KB, RAM 1KB まで
  - ATtiny24/44/84 Automotive: ROM 2/4/8KB, RAM 128/256/512B
- 暗号機能はアプリケーションの一部
  - 暗号アルゴリズムが占有できるメモリ量は、通常全体のごく一部
  - 小さければ小さいほど暗号を使える品種が増える
- Lightweight というからには…
  - ROM 512B, RAM 64B 程度はめざしたいところ
  - この範囲の ROM, RAM サイズが議論されることは少ない

3

既存の評価事例 AES-128

独自インターフェース。C言語から呼び出し可能にするためは()内に示す追加メモリが必要

Algorithm	Processor	ROM	RAM static	RAM stack	Enc Speed	Dec Speed
AES (ED)	ATtiny	1659(+72)	33	0(+24)	$4557n$	$7015n$

http://perso.uclouvain.be/fstandae/lightweight_ciphers/ から作成

C言語から呼び出し可能。但し()内に示す平文・鍵領域やスタックがカウントされていない

Algorithm	Processor	ROM	RAM static	RAM stack	Enc Speed	Dec Speed
AES (ED)	ATmega	2070	176(+32)	0(+22)	$2039+2555n$	$2039+6764n$
AES (ED)	ATmega	2580	176(+32)	0(+22)	$2039+2555n$	$2039+3193n$

http://www.das-labor.org/wiki/AVR-Crypto-Lib/en から作成

C言語から呼び出し可能。RAMサイズには平文、鍵、スタックすべてを含む

Algorithm	Processor	ROM	RAM	Enc Speed	Dec Speed
AES (E)	RL78	486	78	$7288n$	-
AES (E)	RL78	1021	60	$3855n$	-
AES (ED)	RL78	970	84	$7743n$	$1821+10862n$
AES (ED)	RL78	1989	64	$3917n$	$893+5911n$

Matsui, Murakami: FSE2013

(E) Enc only
(ED) Enc+Dec
n: blocks
Size: bytes
Speed: cycles

4

既存の評価事例 Present-80

独自インターフェース。C言語から呼び出し可能にするためは()内に示す追加メモリが必要

Algorithm	Processor	ROM	RAM static	RAM stack	Enc Speed	Dec Speed
Present (ED)	ATtiny	1000(+72)	18	0(+24)	$11342n$	$13599n$

http://perso.uclouvain.be/fstandae/lightweight_ciphers/ から作成

上と同様の独自インターフェース。さらにカウントされていないスタックを加算

Algorithm	Processor	ROM	RAM static	RAM stack	Enc Speed	Dec Speed
Present (E)	ATtiny	204(+72)	18	0(+28)	$190048n$	-
Present (ED)	ATtiny	272(+72)	18	0(+30)	$190048n$	$253384n$
Present (E)	ATtiny	210(+72)	18	0(+28)	$55784n$	-
Present (ED)	ATtiny	278(+72)	18	0(+30)	$55784n$	$77304n$

http://rfidsec2013.iaik.tugraz.at/res/slides/Session4_Talk2_Verstegen.pdf から作成

C言語から呼び出し可能。RAMサイズには平文、鍵、スタックすべてを含む

Algorithm	Processor	ROM	RAM	Enc Speed	Dec Speed
Present (E)	RL78	210	54	$144879n$	-
Present (E)	RL78	897	42	$9007n$	-
Present (ED)	RL78	512	62	$61634n$	$44068+60834n$
Present (ED)	RL78	1855	48	$9007n$	$1903+8920n$

Matsui, Murakami: FSE2013

(E) Enc only
(ED) Enc+Dec
n: blocks
Size: bytes
Speed: cycles

5

評価方法

- インターフェースの統一が必要
  - 小型実装では、インターフェースの違いによるサイズ差は無視できない
  - 暗号を利用することによるすべてのオーバーヘッドを数値化すべき
- 実用性の観点から
  - 評価対象は高級言語から呼び出し可能なサブルーチンとして記述する
  - RAM サイズには平文や鍵の領域、スタックをすべて含める
  - アプリケーションプログラムの範囲をこえる特殊なことはしない
- 評価対象のソフトウエア仕様
  - 1ブロックを暗号化／復号する機能をもつ
  - 平文領域と暗号文領域は共通化する
  - 鍵領域は終了時に元の状態を復帰（一時的に変更してもよい）

6

評価対象と評価項目

- 評価対象

	AES	Camellia	Clefia	TDES	LED	Prince	Present	Piccolo	Twine
ブロックサイズ	128	128	128	64	64	64	64	64	64
鍵サイズ	128	128	128	168	128	128	80	80	80

- 評価環境
  - ルネサスマイコンRL78　CISCプロセッサで小型化に向いている
- 評価項目
  1. ROM 512B/1024B, RAM 64B/128B の4通り制約条件のもとで、暗号化のみの実装と、暗号化＋復号の実装をおこなう
  2. 暗号化のみで、ROM サイズを最小化する実装をおこなう
  3. ROM 2KB程度で、暗号化がどこまで高速になるかを調べる

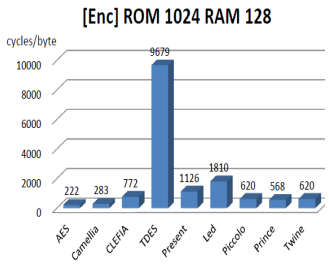
7

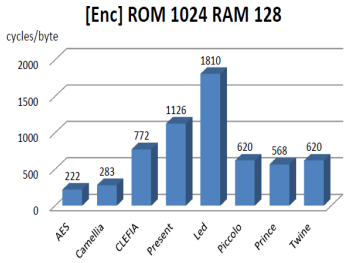
RL78 v.s. ATtiny

		RL78	ATtiny
ハードウエアレジスタ	レジスタ長	8, 16	8
	レジスタ数	8	32
アドレッシングモード	Read-Modify	Yes	No
	Post-Increment	No	Yes
命令長（bytes）	xor reg, [mem]	1-3	4
	call	3	2
	push / pop	1	2
実行時間（cycles）	read from RAM/ROM	1/4	2/3
	xor reg, [mem]	1	2
	taken/not-taken jump	4/2	2/1
	call + return	9	7

8

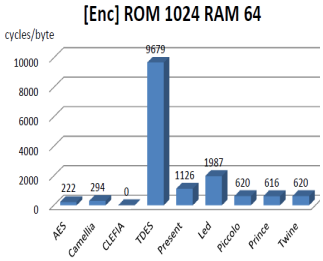
評価結果1：(E) ROM 1024B, RAM 128B

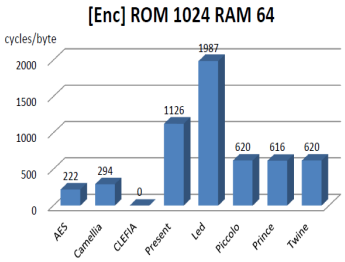




9

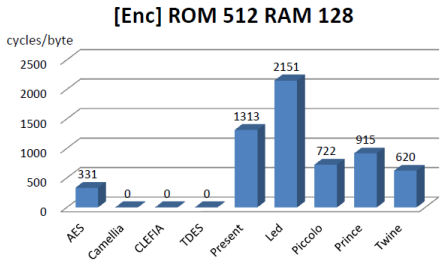
評価結果2：(E) ROM 1024B, RAM 64B





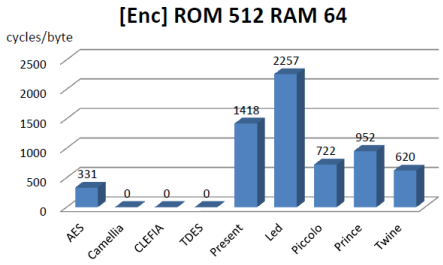
10

評価結果3：(E) ROM 512B, RAM 128B



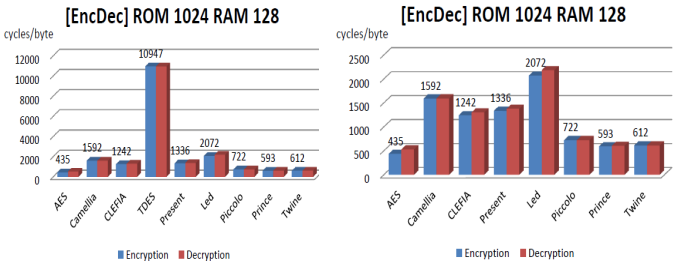
11

評価結果4：(E) ROM 512B, RAM 64B



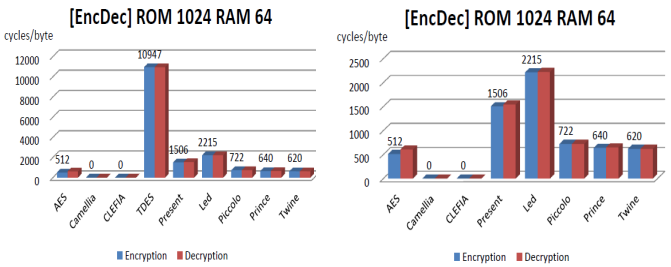
12

評価結果5：(ED) ROM 1024B, RAM 128B



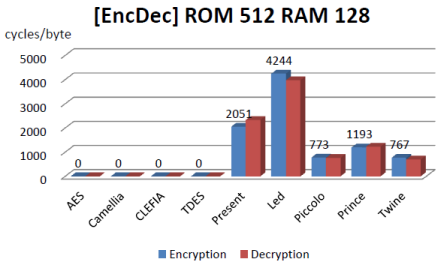
13

評価結果6：(ED) ROM 1024B, RAM 64B



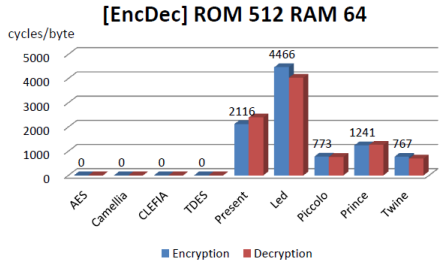
14

評価結果7: (ED) ROM 512B, RAM 128B

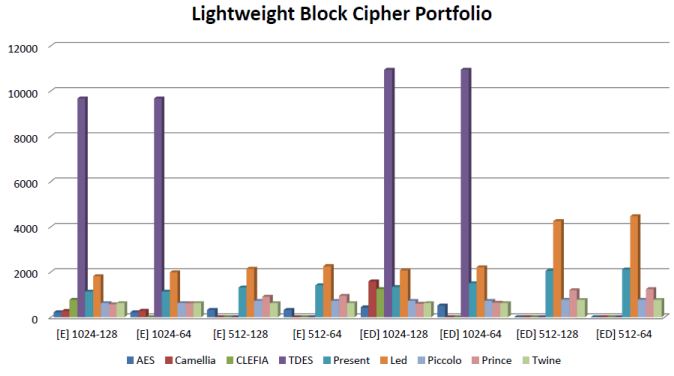


15

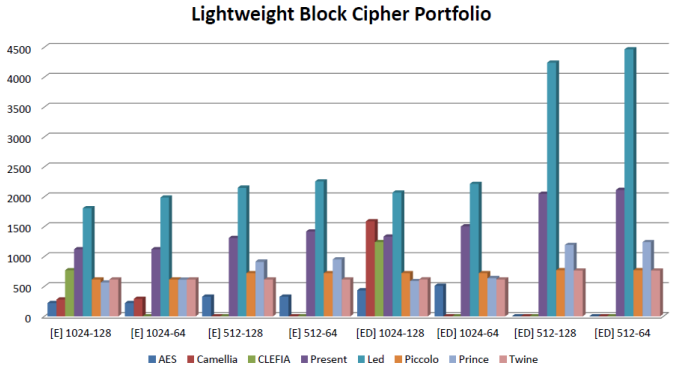
評価結果8: (ED) ROM 512B, RAM 64B



16

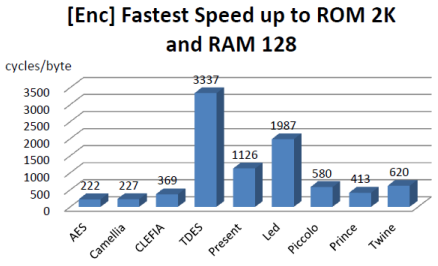


17



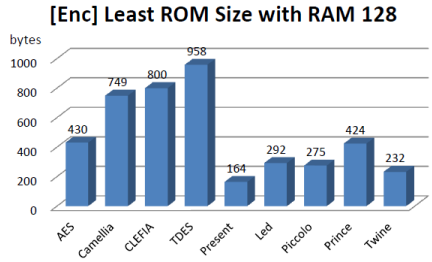
18

評価結果9：(E) Fastest Speed

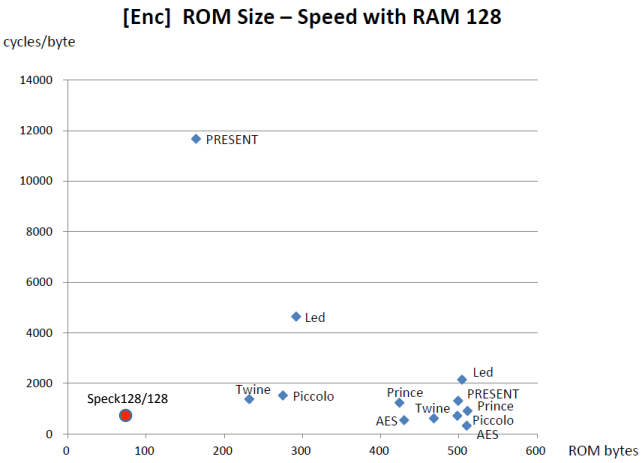


19

評価結果10：(E) Smallest Size



20



21

Lightweight v.s. AES

- アルゴリズム単体で考えるなら、暗号化のみならROM 512B、暗号復号両方こみならROM 1KBあれば、AESで十分
- 実際にはこれに加えモードを含む入出力データ処理が必要、またプロセッサのメモリ全てを暗号が使えるわけではない
- ROM 4KB-8KB, RAM 256B-512Bが、AESをソフトウエアで使えるプロセッサの下限と思われる。
- AESより価値あるソフトウエアLightweightブロック暗号とは…
  - メモリがたくさんあればAESなみの速度がでる
  - 暗号・復号こみでROM 200B以下、RAM 32B以下でそれなりの速度
  - 現時点ではNSAのSimon, Speckが有力候補（安全性は不明）

22

Software Lightweight Design

- 小型化実装は高速化実装と感覚がずいぶん違う
  - 無駄なコードを付け加えることが最終的に小型化に貢献することがある
  - 10バイト減らすと10倍遅くなることがある
- ほんの少しのことがコードサイズに大きく影響する
  - データの単なる移動や定数もオーバーヘッド
  - 数少ない単純な繰り返し構造だけでアルゴリズムを作る必要がある
- 鍵スケジュールがsoftware lightweightでない方式が多い
  - On-the-fly key schedulingを前提に設計すべき
- 回転シフト命令の効率はプロセッサに大きく依存
  - シフト命令もできるだけ避けよ
- Endian Neutralなアルゴリズムが望ましい
  - 今ではほとんどのプロセッサがlittle endianメモリアクセスなのに、多くのアルゴリズムがbig endianを前提に設計されている

23

その他私見

- 今回評価対象としたのはすべてS-box型ブロック暗号
- Lightweight ブロック暗号のトレンドは4ビットS-box
  - Present, LED, Prince, Piccolo, Twine
  - S-box型が安全性の評価がしやすい
- これは Lightweight として正しい方向か？
  - Simon, Speck が問うているもの
  - このタイプのブロック暗号TEAは昔からあった
- 暗号理論的にどこまで完全な安全性をめざすべきか？
  - 現実には side-channel attacks の方が脅威
  - そもそも64ビットブロック暗号がリバイバルしている
  - 安全性の条件を再定義する方向もあるのではないか

24

資料 2

2014 年度 暗号技術活用委員会活動報告

１． 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（旧名；現「SSL/TLS 暗号設定ガイドライン」）」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 委員構成

暗号技術活用委員会の委員は、表 1 の通り。

表 1　2014 年度暗号技術活用委員会　委員名簿

	委員氏名	所属
委員長	松本　勉	横浜国立大学　大学院環境情報研究院 教授
委員	上原　哲太郎	立命館大学　情報理工学部情報システム学科 教授
委員	遠藤　直樹	東芝ソリューション株式会社　技術統括部 技監
委員	川村　亨	日本電信電話株式会社　研究企画部門 プロデュース担当（セキュリティ） 担当部長／チーフプロデューサ
委員	菊池　浩明	明治大学　総合数理学部　先端メディアサイエンス学科 教授
委員	鈴木　雅貴	日本銀行　金融研究所　情報技術研究センター 主査 （※2015 年 3 月 15 日退任 or 所属変更）
委員	高木　繁	株式会社三菱東京 UFJ 銀行　システム部システム企画室 次長
委員	角尾　幸保	日本電気株式会社　パブリックビジネスユニット 宇宙・防衛事業推進本部 主席技術主幹
委員	手塚　悟	東京工科大学　コンピュータサイエンス学部 教授
委員	松井　充	三菱電機株式会社　情報技術総合研究所 技師長
委員	満塩　尚史	内閣官房　IT 総合戦略室 政府 CIO 補佐官
委員	八束　啓文	EMC ジャパン株式会社　RSA 事業本部　技術統括部 システムズ・エンジニアリング部　部長
委員	山口　利恵	東京大学　ソーシャル ICT 研究センター 特任准教授
委員	山田　勉	株式会社日立製作所　日立研究所 エネルギーマネジメント研究部　ES3 ユニット ユニットリーダ主任研究員
委員	山本　隆一	東京大学　大学院医学系研究科　医療経営政策学講座 特任准教授

1.3 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 2 のとおり。

表 2 2014 年度暗号技術活用委員会 開催概要

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

1.4 成果概要

1.4.1 暗号技術活用委員会の成果概要

≪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 年間の調査結果及び検討結果を、以下の通り、別添 1 に取りまとめた。

- 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

≪RC4 の注釈についての検討≫

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。」

1.4.2 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ガイドラインとして取りまとめた。

「SSL/TLS 暗号設定ガイドライン」本文を別添 2 に、「SSL/TLS 暗号設定ガイドラインチェックリスト」を別添 3 に添付する。

本ガイドラインの対象読者は、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。また、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も示している。

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。

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。

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 3, 表 4 参照）を提示している。

また、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。

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。

表 3　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。

表 4　要求事項の整理

<table>
<tr><td colspan="2">要件</td><td>高セキュリティ型</td><td>推奨セキュリティ型</td><td>セキュリティ例外型</td></tr>
<tr><td colspan="2">想定対象</td><td>G2G</td><td>一般</td><td>レガシー携帯電話を含む</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="4">暗号アルゴリズム</td><td>鍵交換</td><td>DHE 2048 bit
ECDHE 256 bit</td><td>DHE 1024 bit 以上
ECDHE 256 bit
RSA 2048 bit
ECDH 256 bit</td><td>DHE 1024 bit 以上
ECDHE 256 bit
RSA 2048 bit
ECDH 256 bit</td></tr>
<tr><td>暗号化</td><td>AES 256, 128
CAMELLIA 256, 128</td><td>AES 256, 128
CAMELLIA 256, 128</td><td>AES 256, 128
CAMELLIA 256, 128
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>SHA384, SHA256</td><td colspan="2">SHA384, SHA256, SHA1</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, SSL 3.0</td></tr>
<tr><td colspan="2">証明書での鍵長</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">SHA256</td><td>SHA256, SHA1</td></tr>
</table>

【委員構成】

運用ガイドライン WG の委員は表 5 の通り。

【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 26 日

表 5　2014 年度運用ガイドライン WG　委員名簿

	委員氏名	所属
主査	菊池 浩明	明治大学
委員	阿部 貴	株式会社シマンテック
委員	漆嶌 賢二	富士ゼロックス株式会社
委員	及川 卓也	グーグル株式会社
委員	加藤 誠	一般社団法人 Mozilla Japan
委員	佐藤 直之	株式会社イノベーションプラス
委員	島岡 政基	セコム株式会社IS研究所
委員	須賀 祐治	株式会社インターネットイニシアティブ
委員	高木 浩光	独立行政法人産業技術総合研究所
委員	村木 由梨香	日本マイクロソフト株式会社
委員	山口 利恵	東京大学

1.4.3　標準化推進 WG 概要報告

標準化推進 WG は、2013 年度の成果を踏まえて、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取りまとめた。

「暗号技術参照関係の俯瞰図」を別添 4 に、「標準化提案における交渉ノウハウ・課題及び参考情報」を別添 5 に添付する。

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。

【委員構成】

標準化推進 WG の委員は表 6 の通り。

表 6　2014 年度標準化推進 WG　委員名簿

	委員氏名	所属	担当領域
主査	渡辺　創	独立行政法人産業技術総合研究所	ISO/IEC JTC1/SC27
委員	江原　正規	東京工科大学	ISO/IEC JTC1/SC31
委員	河野　誠一	レノボ・ジャパン株式会社	TCG
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	IETF
委員	坂根　昌一	シスコシステムズ合同会社	M2M/IoT
委員	佐藤　雅史	セコム株式会社	長期署名（ETSI）
委員	武部　達明	横河電機株式会社	制御機器・制御システム
委員	廣川　勝久	ISO/IEC JTC1/SC17 国内委員会	ISO/IEC JTC1/SC17
委員	真島　恵吾	日本放送協会	放送
委員	真野　浩	コーデンテクノインフォ株式会社	IEEE802.11
委員	茗原　秀幸	三菱電機株式会社	医療

【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

資料 2
別添 1

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

1　ヒアリング調査結果

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	C 社
	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

1.2　ヒアリング調査結果概要

【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「暗号利用環境調査」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。

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 文献調査結果

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、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。

⑪ 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

⑫ 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 米国「IT Security」

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。

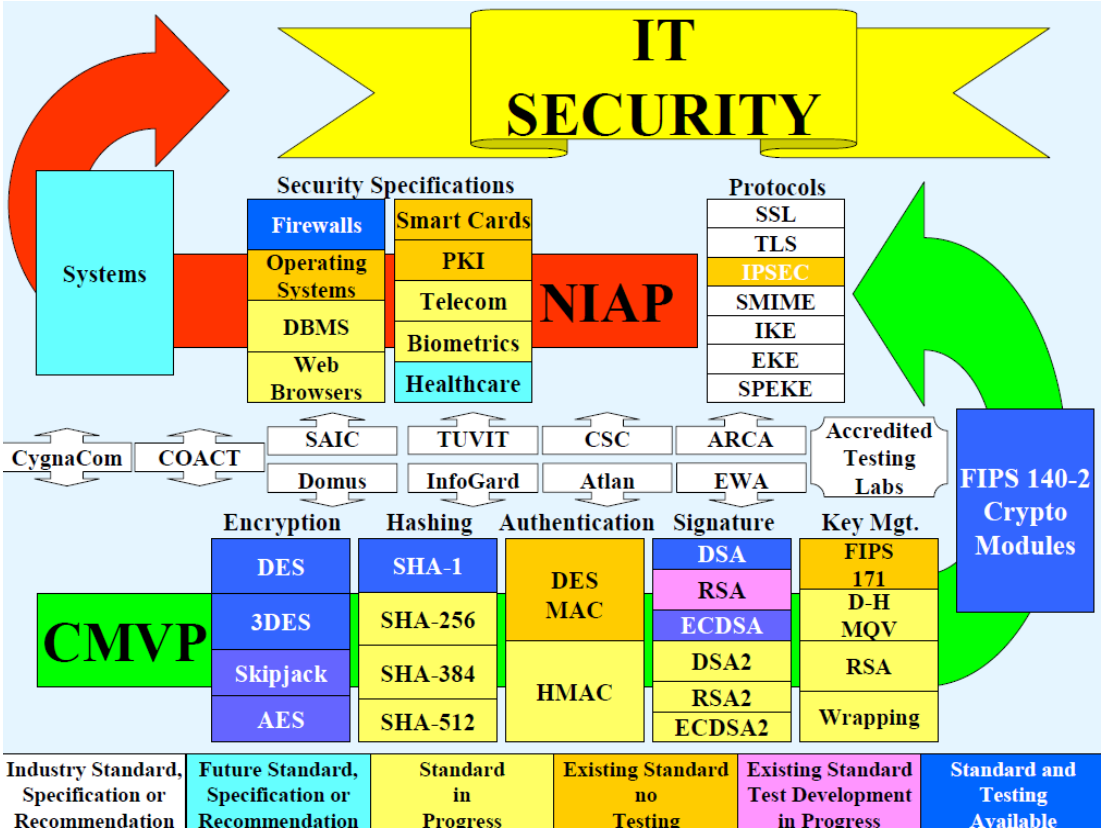
米国では、CMVP conference 2002 での資料からもわかるように、暗号アルゴリズムが製品・システムとして IT セキュリティに寄与するまでには要求される考え方・思想が異なる複数の階層が存在していることを 10 年以上前から認識しており、それぞれの階層で役割を分担しつつ、有機的に連携

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

2.3 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）

3　暗号技術活用委員会での議論概要

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。

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。

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。

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。

- 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
- 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。

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。

4　今後の検討にあたっての留意点

1節から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　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ると考えられ、そのようなニーズを有する者が積極的に役割を担っていくことについて整理すべきである。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、サイバーセキュリティ基本法の制定を踏まえ、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づくり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。

資料２
別添２

SSL/TLS 暗号設定ガイドライン

（2015.3.23 版）

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いないので、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　新規事業開発部 SkyDesk サービスセンター　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よく知られた脆弱性が発見され、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 SSL 3.0 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。ここで TLS1.0 はプロトコルとして下位互換性を持っており SSL3.0 しか対応していないエンティティとも通信可能なように設計されている。

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-256/384)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。

また、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。

表 2　SSL/TLS のバージョン概要

バージョン（開発年）	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

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

	● 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。なお、SSL3.0 で問題となった POODLE 攻撃に関しては 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 攻撃など）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の SHA-256 と SHA-384、並びに 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 攻撃など）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のことであり、TLS1.1 同様、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] 更新：SSL 3.0 の脆弱性対策について(CVE-2014-3566)、
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。

＊ Handshake Protocol w/ シーケンシャル図
＊ 暗号化の仕組み
＊ MAC の仕組み
＊ サーバ認証，クライアント認証の仕組み

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 [9]。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） では以下のとおり記載されており、 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）

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）

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

2.2.2 異なる暗号アルゴリズムにおけるバランスがとれた安全性

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[10]）で表現され、鍵長[11]とは別に求められる。表記上、解読計算量が 2^xである場合に“xビット安全性”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 SP800-5 Part 1 revision 3[12]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。SSL/TLS では、サーバ証明書で利用される。

---

[10] 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

[11] ハッシュ関数の場合はダイジェスト長に相当する

[12] 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>2014 年から 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：

サーバ構築での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の責任で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。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の 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する。

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** ＜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

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

3.2 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

要件		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
想定対象		G2G	一般	レガシー携帯電話含む
暗号スイートの（暗号化の）セキュリティレベル		①256 bit ②128 bit	①128 bit ②256 bit	① 128 bit ② 256 bit ③ RC4, Triple DES
暗号アルゴリズム	鍵交換	DHE 2048 bit ECDHE 256 bit	DHE 1024 bit 以上 ECDHE 256 bit RSA 2048 bit ECDH 256 bit	DHE 1024 bit 以上 ECDHE 256 bit RSA 2048 bit ECDH 256 bit
	暗号化	AES 256, 128 CAMELLIA　256, 128	AES 256, 128 CAMELLIA　256, 128	AES 256, 128 CAMELLIA　256, 128 RC4 Triple DES
	モード	GCM	GCM, CBC	
	ハッシュ関数	SHA-384, SHA-256	SHA-384, SHA-256, SHA-1	
プロトコルバージョン		TLS1.2 のみ	TLS1.2 ～ TLS1.0	TLS1.2～1.0, SSL 3.0
証明書鍵長		鍵長 2048 ビット以上の RSA または 鍵長 256 ビット以上の ECDSA		
証明書でのハッシュ関数		SHA-256		SHA-256, SHA-1

3.3 チェックリスト

図 1 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。

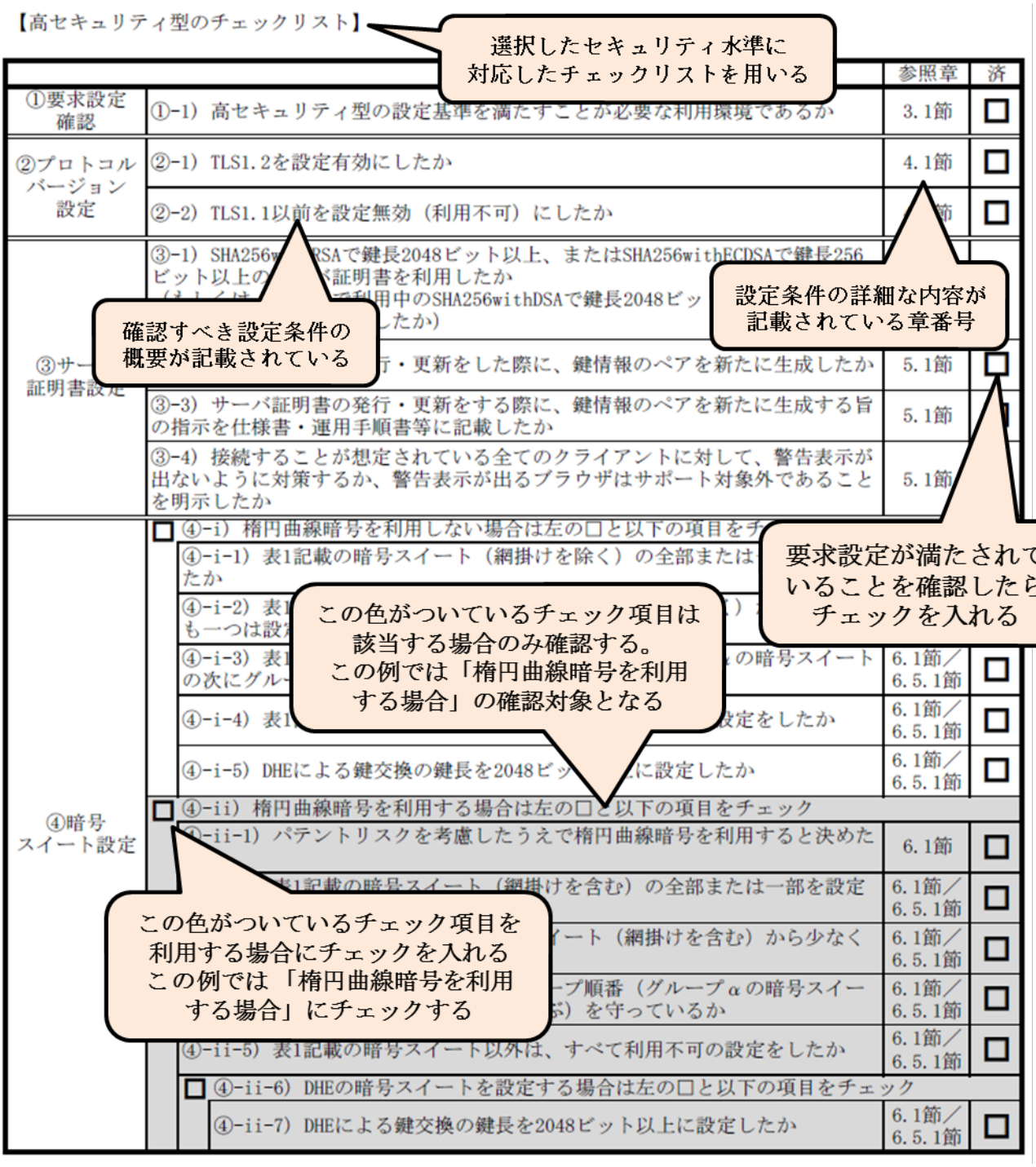


図 1　チェックリスト（高セキュリティ型の例）

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での安全性の違い

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
利用できる暗号アルゴリズム	**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】

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のに対して、POODLE 攻撃に対しては 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ため、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から、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。

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 節から 5.3.4 節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

5.2 節と 5.3 節での内容を踏まえ、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においては、推奨セキュリティ型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[13]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、5.2 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。

【高セキュリティ型の要求設定】

- 本ガイドライン公開時点で、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	RSA と SHA-256 の組合せ（SHA256withRSA）で鍵長は 2048 ビット以上、 または ECDSA と SHA-256 の組合せ（SHA256withECDSA）で鍵長は 256 ビット（NIST P-256）以上、 を必須とする

[13] 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

	※ ただし、すでに DSA と SHA-256 の組合せ（SHA256withDSA）で鍵長 2048 ビット以上を利用している場合には現行の有効期限内での継続利用を容認する
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 ➢ 5.3.2 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	RSA と SHA-256 の組合せ（SHA256withRSA）で鍵長は 2048 ビット以上、または ECDSA と SHA-256 の組合せ（SHA256withECDSA）で鍵長は 256 ビット（NIST P-256）以上、 を必須とする ※ ただし、すでに DSA と SHA-256 の組合せ（SHA256withDSA）で鍵長 2048 ビット以上を利用している場合には現行の有効期限内での継続利用を容認する
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 ➢ 5.3.2 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。

サーバ証明書の暗号アルゴリズムと鍵長	RSA の鍵長は 2048 ビット、ハッシュ関数は SHA-256 または SHA-1 を必須とする。SHA-256 との組合せ（SHA256withRSA）のほうが望ましいが、SHA-1 との組合せ（SHA1withRSA）を選んでもよい ※ 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 ➢ 5.3.2 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 サーバ証明書で利用できる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できる候補となる暗号アルゴリズム」とは、サ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 7 の示した「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 7　サーバ証明書で利用できる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

5.3 サーバ証明書で考慮すべきこと

5.3.1 信頼できないサーバ証明書の利用は止めるべき

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。

5.3.2 ルート CA 証明書の安易な手動インストールは避けるべき

5.3.1 節のようにして発行されたサーバ証明書を利用したサーバを“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、5.3.1 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、当該組織の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。

5.3.3 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が出ている。また、離散対数問題の困難性に関する調査研究も行われている（詳細についてはCRYPTREC Report 2013[14]を参照のこと）。

14 http://www.cryptrec.go.jp/report/c13_eval_web_final.pdf

以上の報告と、内閣官房情報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[15]」を踏まえれば、本ガイドライン公開時点でサーバ証明書が利用すべき鍵長は、RSAとDSAは 2048 ビット以上、ECDSAは 256 ビット以上が妥当である。

5.3.4 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を正しく生成・運用していないと、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。

[15] http://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf

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ており、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 の特性を満たす。
- 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。
- 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。
  - 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、RSA 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 1048 ビット以上 [16]、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。
  - 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。
  - 今までほとんど使われていない楕円暗号と 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、楕円暗号と Triple DES、RC4 の組み合わせは選定しない。
- 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、・・・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 1048 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。

[16] 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

6.2 暗号スイートで利用できる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できる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 8 に示した暗号アルゴリズムである。

表 8　暗号スイートで利用できる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

以下は SSL 3.0 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[17]やENISA[18]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が共有する必要があるため、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 8 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。

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。スノーデンの告白にもあるように､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

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

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

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

5.3.3 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。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おり（表 3 参照）、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、これ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の予測を踏まえれば、本ガイドライン公開時点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 2 の 2015 年 1 月のAlexaの調査結果 [19]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。

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
- Nginx

これらについては Appendix B 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。

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

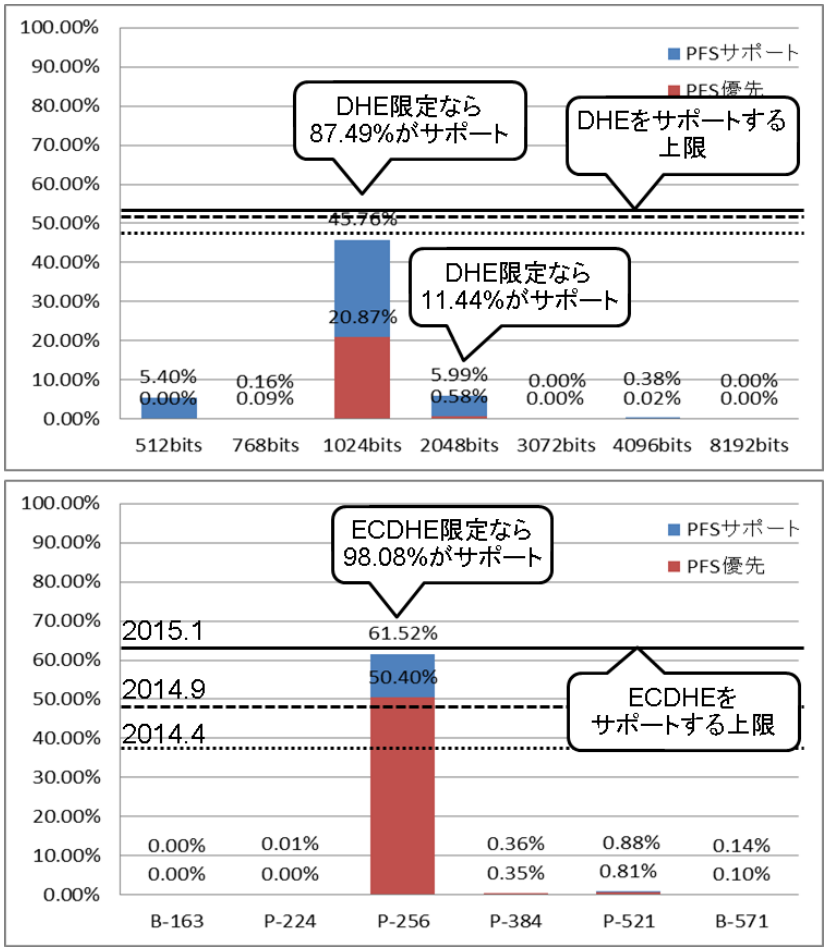


図 2　DHE/ECDHE の鍵長の設定状況

【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。

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

ートを多い。

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 9 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 [20]。

表 9　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	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
グループ β	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
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 10 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 10 の暗号スイートグループを追加してよい。

[20] 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規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

表 10　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	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
グループβへの追加または代替	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256

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 11 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。なお、（RFC必須）は、TLS1.2 においてサポートが必須と規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ことが推奨される暗号スイートである [21]。

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 11　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	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	DHE-DSS-AES128-SHA256
	DHE-RSA-AES128-SHA256
	DHE-DSS-CAMELLIA128-SHA256
	DHE-RSA-CAMELLIA128-SHA256
	DHE-DSS-AES128-SHA
	DHE-RSA-AES128-SHA
	DHE-DSS-CAMELLIA128-SHA
	DHE-RSA-CAMELLIA128-SHA

[21] TLS1.1 及び TLS1.0 でのサポートが必須と規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 攻撃など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

グループ B	AES128-GCM-SHA256
	CAMELLIA128-GCM-SHA256
	AES128-SHA256
	CAMELLIA128-SHA256
	AES128-SHA　（RFC 必須）
	CAMELLIA128-SHA
グループ C	該当なし
グループ D	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	DHE-DSS-AES256-SHA256
	DHE-RSA-AES256-SHA256
	DHE-DSS-CAMELLIA256-SHA256
	DHE-RSA-CAMELLIA256-SHA256
	DHE-DSS-AES256-SHA
	DHE-RSA-AES256-SHA
	DHE-DSS-CAMELLIA256-SHA
	DHE-RSA-CAMELLIA256-SHA
グループ E	AES256-GCM-SHA384
	CAMELLIA256-GCM-SHA384
	AES256-SHA256
	CAMELLIA256-SHA256
	AES256-SHA
	CAMELLIA256-SHA
グループ F	該当なし
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 12 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 12 の暗号スイートグループを追加してよい。

表 12　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	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-SHA256
	ECDHE-RSA-AES128-SHA256
	ECDHE-ECDSA-CAMELLIA128-SHA256
	ECDHE-RSA-CAMELLIA128-SHA256
	ECDHE-ECDSA-AES128-SHA
	ECDHE-RSA-AES128-SHA
グループ C への追加	ECDH-ECDSA-AES128-GCM-SHA256
	ECDH-RSA-AES128-GCM-SHA256
	ECDH-ECDSA-CAMELLIA128-GCM-SHA256
	ECDH-RSA-CAMELLIA128-GCM-SHA256
	ECDH-ECDSA-AES128-SHA256
	ECDH-RSA-AES128-SHA256
	ECDH-ECDSA-CAMELLIA128-SHA256
	ECDH-RSA-CAMELLIA128-SHA256
	ECDH-ECDSA-AES128-SHA
	ECDH-RSA-AES128-SHA
グループ D への追加または代替	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-SHA384
	ECDHE-RSA-AES256-SHA384
	ECDHE-ECDSA-CAMELLIA256-SHA384
	ECDHE-RSA-CAMELLIA256-SHA384
	ECDHE-ECDSA-AES256-SHA
	ECDHE-RSA-AES256-SHA
グループ F への追加	ECDH-ECDSA-AES256-GCM-SHA384
	ECDH-RSA-AES256-GCM-SHA384
	ECDH-ECDSA-CAMELLIA256-GCM-SHA384
	ECDH-RSA-CAMELLIA256-GCM-SHA384
	ECDH-ECDSA-AES256-SHA384
	ECDH-RSA-AES256-SHA384
	ECDH-ECDSA-CAMELLIA256-SHA384
	ECDH-RSA-CAMELLIA256-SHA384

	ECDH-ECDSA-AES256-SHA
	ECDH-RSA-AES256-SHA

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 13 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。

なお、(RFC 必須) は、TLS1.2、TLS1.1 及び TLS1.0 においてサポートが必須と規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 13　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	DHE-DSS-RC4-SHA
	RC4-SHA
グループ H	DHE-DSS-DES-CBC3-SHA　（RFC 必須）
	DHE-RSA-DES-CBC3-SHA
	DES-CBC3-SHA　（RFC 必須）
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 12 以外のすべての暗号スイートを利用除外とする

【コラム②】

FREAK[22]攻撃 は、中間者攻撃のなかの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、鍵交換の際に、たとえサーバ証明書で鍵長 2048 ビットの RSA を使う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 が解読されたとしてもおかしくはない。

なお、FREAK 攻撃が成功するためには、サーバとブラウザ（クライアント）の両方が RSA EXPORT を受け付ける設定になっている必要がある。

しかし、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、たとえブラウザ（クライアント）が RSA EXPORT を受け付ける設定になっていたとしても RSA EXPORT が実際に選択されることはなく、FREAK 攻撃は成功しない。

[22] 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たま起きる。

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、脆弱性が指摘されていない暗号モジュールを利用して鍵ペアを生成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 [23]。

7.1.2 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 [24]（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）

---

23 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

24 安全性上の問題が生じ、信用できなくなる状態のこと

7.1.3 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

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。

7.1.4 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 を参照されたい。

7.1.5 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。

7.1.6 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 14 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。

表 14　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」機能により、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

7.2 さらなる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：設定により可能
  - lighttpd：設定により可能
  - nginx：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。これにより、例えば、攻撃者が挿入した HTTP リクエストをサーバに送信するといったことができる。

公になっている攻撃例の概要を簡単に示すと以下のようになる。なお、攻撃方法の技術的な詳細については文献などに記載があるので、そちらを参照されたい。

1. クライアントとサーバとの間でハンドシェイクが行われる際に、攻撃者が通信に介入して、クライアントーサーバ間のハンドシェイクを中断させるとともに、攻撃者とサーバとの間でハンドシェイクを実行して先にセッションを確立する
2. 攻撃者が、確立したセッションを使って、挿入したいデータをサーバに送信しておく。この際、通信が完了していない状態（データが不完全な状態）で通信を中断する。これにより、サーバでは、データが完全なものになるまで受信待機に入る
3. 攻撃者がリネゴシエーションを要求し、1.で中断していたハンドシェイクを再開させる。この時、暗号化されているセッションが、攻撃者―サーバ間であることがポイントである。つまり、サーバは、クライアントとの間で暗号化されたセッションを使って二回目のハンドシェイクをしているつもりになっているが、実際には攻撃者との間でハンドシェイクをしているに過ぎない。
4. 攻撃者からのリネゴシエーションの要求でサーバが攻撃者に送信したハンドシェイクの情報をそのままクライアントに転送する。
5. クライアントとサーバとの間でセッションが確立する。ただし、クライアントは 1.によるセッション確立なのに対して、サーバは 3.によるセッション確立であることに注意。
6. 確立したセッションを使って、クライアントがサーバにデータを送信する。その際、サーバでは、3.でのリネゴシエーションによるセッション確立だと誤認しているので、2.で受信待機になっていたデータに連結して受信する

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ところである。

実際には、クライアントは一回目のハンドシェイクで確立した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になっているにも関わらず、両者がその食い違いを認識できない。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とリネゴシエーション後のクライアントからの通信を、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。

一方、アプリケーション側では圧縮機能を利用するほうがメリットがあるケースも少なくないことから、利用することによるアプリケーション側でのメリットと、圧縮機能を利用した攻撃を実際に受けるリスクを踏まえて、利用可否の設定を決めるべきである。

7.2.4 OCSP Stapling の設定有効化

サーバの運用を止めたり、サーバの秘密鍵が漏洩したりなどの理由で、サーバ証明書を失効させることができる。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 と OCSP 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) 多くのブラウザでは、認証局の運用する OCSP レスポンダ（サーバ）に通信障害やバグ等の理由で OCSP 応答を取得できなかった場合、これを SSL のエラーとはせず、証明書チェーンの検証が有効であるとして、SSL 通信を許可してしまう。そのため、あるサイトのサーバ証明書が失効していたとしても、DoS などにより意図的に OCSP レスポンダに接続させない事により、当該サイトは有効であるとして SSL 通信ができてしまう

2) あるサイトにアクセスがあったことを、サイトの運営者、通信事業者、利用者組織に加えて、OCSP を使った場合には認証局も知り得てしまうため、プライバシー上の懸念が持たれている。例えば、ある利用者が、ある特別な会員制の HTTPS サイトにブラウザでアクセスした場合、ブラウザは失効検証のためにその HTTPS サイトの OCSP 応答を取得しようとする。この際、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、その会員であると認証局が知り得ることになる

上記二つ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、TLS セッションの確立の際に、OCSP 応答を認証局の OCSP レスポンダからではなく、サーバから取得して TLS 通信を開始することができる。

これにより、

1) OCSP レスポンスはウェブサーバから取得するので、ブラウザと認証局の OCSP レスポンダとの通信が何らかの原因でできなかったとしても、必ず（サーバが OCSP 応答をキャッシュしている限りは）OCSP 応答による失効検証を行うことができる

2) OCSP レスポンスは認証局からではなく、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認証局が知ることは無くなる

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。

- 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を含む SPKI（SubjectPublicKeyInfo）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
  - Mozilla Firefox

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のリストを保持し、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。現在、IETF において、Public Key Pinning Extension for HTTP として標準策定中である。参考までに、いくつかの設定例を Appendix B.3.で紹介する。

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 15）。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 15　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

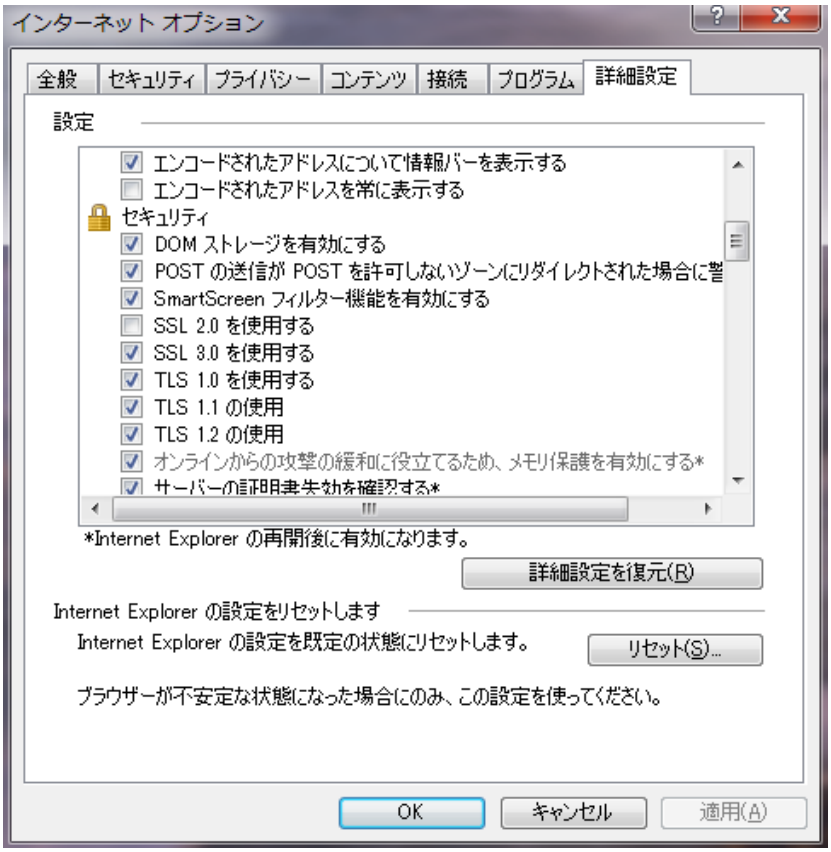
  http://msdn.microsoft.com/ja-jp/ie/cc844005.aspx

  【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、「SSL 2.0 を使用する」「SSL 3.0 を使用する」「TLS 1.0 を使用する」「TLS 1.1 を使用」「TLS 1.2 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のデフォルト設定では、IE6 では「SSL 2.0 を使用する」にチェックが入っている一方、IE8 以降では TLS 1.1 や TLS 1.2 をサポートしているものの「TLS 1.1 を使用」「TLS 1.2 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。



	TLS 1.2	TLS 1.1	TLS 1.0	SSL 3.0	SSL 2.0
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

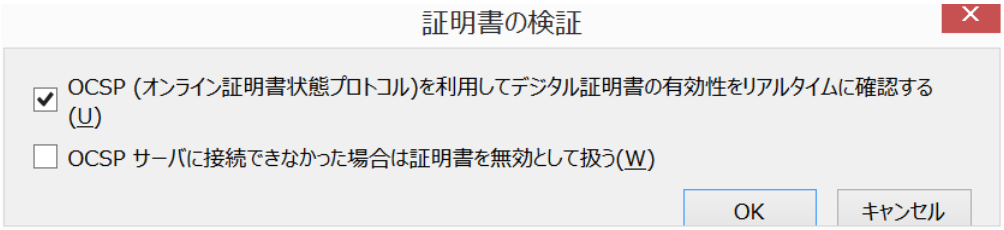
- Firefox

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、理由がない以外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 [25]。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 [26]。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
41	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	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 [27]。
  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 [28]。

25 http://blogs.technet.com/b/pki/archive/2013/11/12/sha1-deprecation-policy.aspx
26 http://blog.chromium.org/2014/09/gradually-sunsetting-sha-1.html
27 https://wiki.mozilla.org/CA:MD5and1024
28 https://blog.mozilla.org/security/2014/09/23/phasing-out-certificates-with-sha-1-based-signat

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

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
  当面の回避策として新たに「Fix it」が公開された。 下記 URL に記載されている回避策の「Fix it ツールを利用する」を実施することで、SSL3.0 を無効化できる。
  なお、今後数ヵ月以内に、デフォルトでSSL3.0 を無効化する措置を講じる予定がある [29]。

  参考：
  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
  本ガイドライン公開時点で Google 社からは SSL3.0 を無効化する方法は公開されていないが、数ヵ月後には SSL 3.0 のサポートを完全に打ち切る予定であると発表している。

  This POODLE bites: exploiting the SSL 3.0 fallback(English)
  http://googleonlinesecurity.blogspot.jp/2014/10/this-poodle-bites-exploiting-ssl-30.html

- Firefox
  次期バージョンからデフォルトで SSL 3.0 を無効化すると発表している。
  Mozilla Security Blog (English)
  https://blog.mozilla.org/security/2014/10/14/the-poodle-attack-and-the-end-of-ssl-3-0/

urealgorithms/

[29] http://blogs.technet.com/b/jpsecurity/archive/2014/10/30/3009008-ssl3-rev.aspx

また、現行バージョンの Firefox で SSL 3.0 を無効化するアドオンも公開された。
Mozilla Japan ブログ
http://www.mozilla.jp/blog/entry/10433/

8.3.3 サーバ証明書の検証方法

- 鍵マーク、グリーンバー（EV のみ）の確認

9. その他のトピック

9.1 リモートアクセス VPN on 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 をインストールしてきて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。

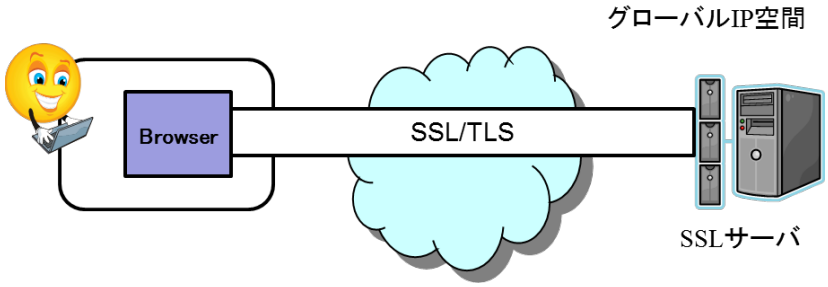
クライアントレス型は、ブラウザさえあればどの端末からでもアクセス可能であり、利便性に優れる一方、SSL との最大の差はグローバル IP をインターネットに晒しているか否かぐらいの違いといえる。結果として、最初のクライアント認証を SSL サーバが受け持つか、SSL VPN 装置が受け持つか程度の差でしかなく、VPN というよりも、本質的には SSL と同じものとみるべきである。

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。

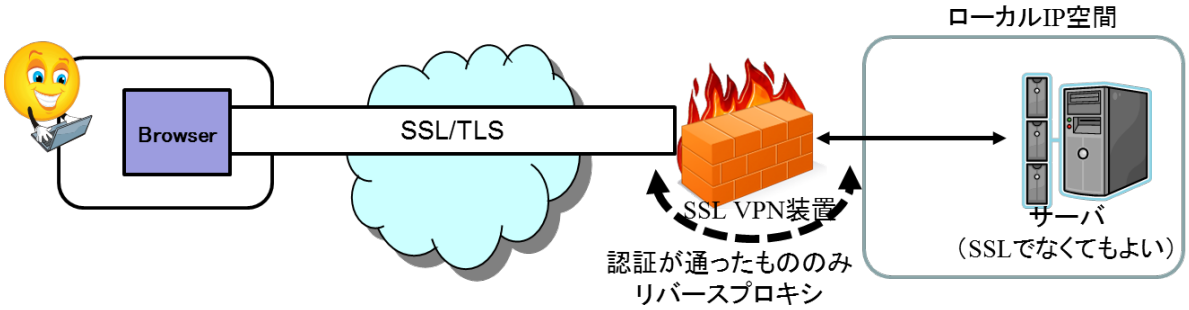
一方、クライアント型は上記の 2 つのタイプとは明らかに異なり、専用のクライアントソフトがインストールされた端末との間でのみアクセスする。つまり、誤って偽サーバに接続することがなく、また内部サーバにアクセスできる端末も厳格に制限できるため、端末に IPsec-VPN ソフトをインストールして構成するモバイル型の IPsec-VPN に近い形での運用形態となる。

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。

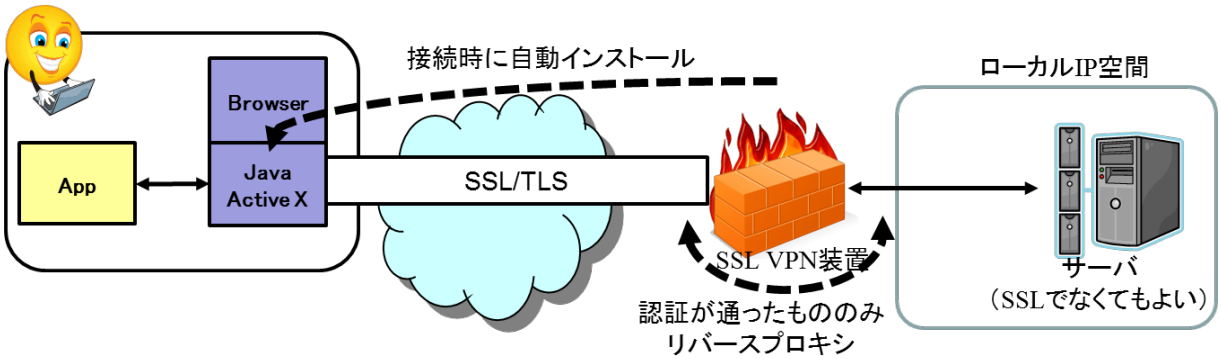
【参考：通常の SSL/TLS】



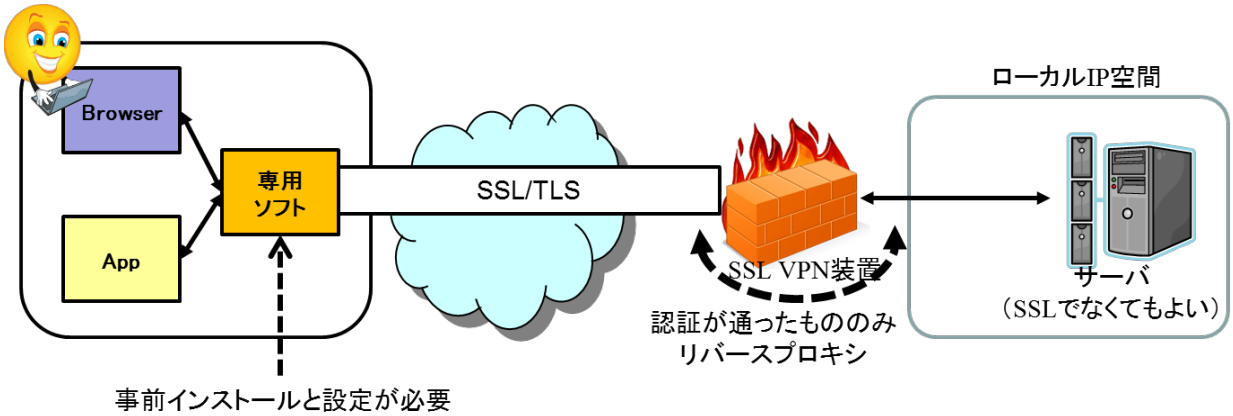
【クライアントレス型（ブラウザベース）】



【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 から入手可能である。

http://xxxxxxxxxxxxx

A. 1. チェックリストの利用方法

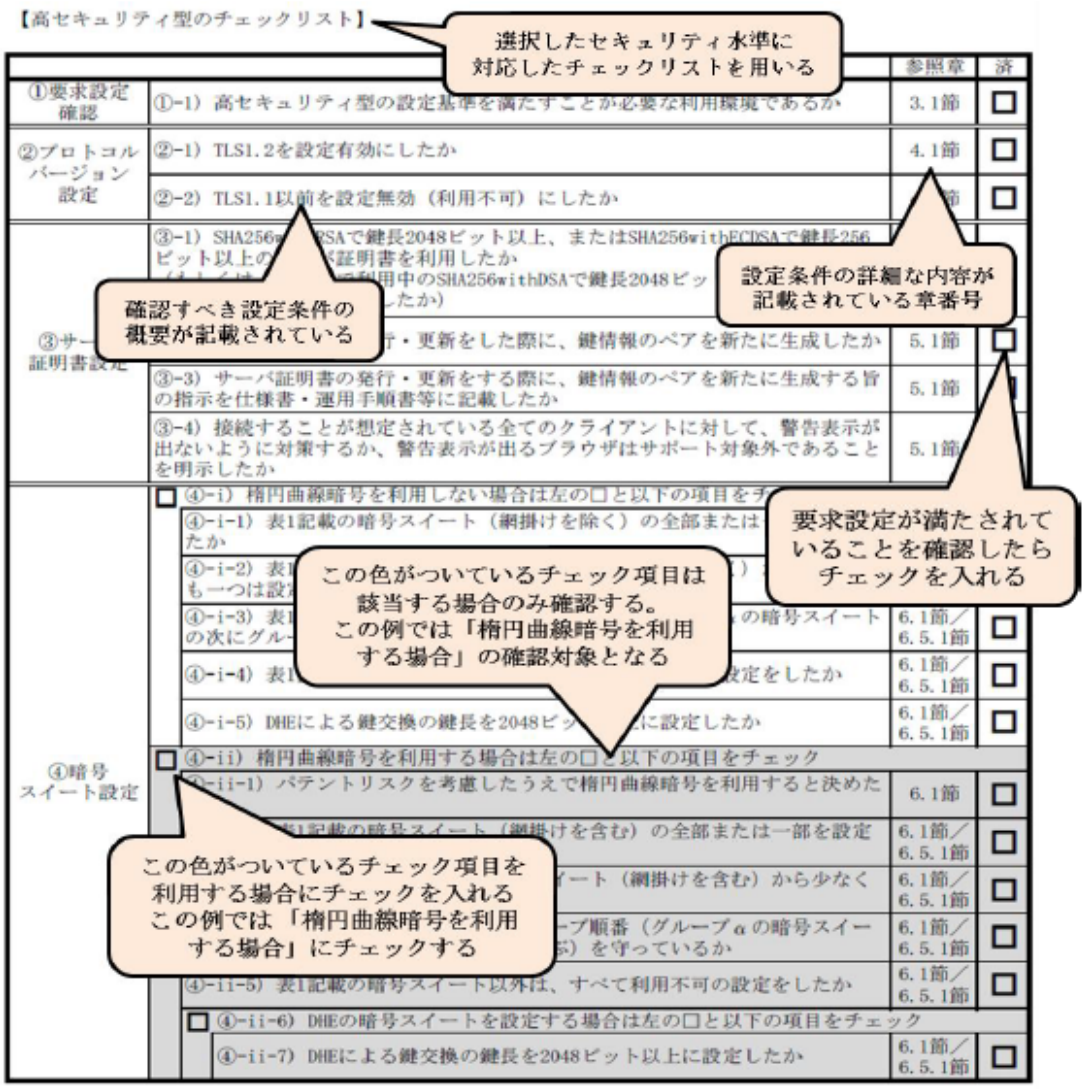
SSL/TLS暗号設定ガイドラインチェックリスト

【チェックリストの使い方】

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。

◇ プロトコルバージョンの設定（ガイドライン第4章）
◇ サーバ証明書の設定（ガイドライン第5章）
◇ 暗号スイートの設定（ガイドライン第6章）

＜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】

		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）SHA256withRSAで鍵長2048ビット以上、またはSHA256withECDSAで鍵長256ビット以上のサーバ証明書を利用したか （もしくは、現時点で利用中のSHA256withDSAで鍵長2048ビット以上のサーバ証明書をそのまま継続利用したか）	5.1節	□
	③-2）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	□
	③-3）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	□
	③-4）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）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-7）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】

グループ α	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
グループ β	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256

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】

		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）SHA256withRSAで鍵長2048ビット以上、またはSHA256withECDSAで鍵長256ビット以上のサーバ証明書を利用したか（もしくは、現時点で利用中のSHA256withDSAで鍵長2048ビット以上のサーバ証明書をそのまま継続利用したか）	5.1節	☐
	③-2）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	☐
	③-3）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	☐
	③-4）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それぞれの暗号スイート（網掛けを除く）から少なくとも一つずつは設定したか	6.1節／6.5.2節	☐
	☐ ④-i-3）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-4）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	☐
	④-i-5）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-6）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）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それぞれの暗号スイート（網掛けを含む）から少なくとも一つずつは設定したか	6.1節／6.5.2節	☐
	☐ ④-ii-4）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-5）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	☐
	④-ii-5）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-6）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	☐

【表2】

グループA	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	DHE-DSS-AES128-SHA256
	DHE-RSA-AES128-SHA256
	DHE-DSS-CAMELLIA128-SHA256
	DHE-RSA-CAMELLIA128-SHA256
	DHE-DSS-AES128-SHA
	DHE-RSA-AES128-SHA
	DHE-DSS-CAMELLIA128-SHA
	DHE-RSA-CAMELLIA128-SHA
	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-SHA256
	ECDHE-RSA-AES128-SHA256
	ECDHE-ECDSA-CAMELLIA128-SHA256
	ECDHE-RSA-CAMELLIA128-SHA256
	ECDHE-ECDSA-AES128-SHA
	ECDHE-RSA-AES128-SHA
グループB	AES128-GCM-SHA256
	CAMELLIA128-GCM-SHA256
	AES128-SHA256
	CAMELLIA128-SHA256
	AES128-SHA
	CAMELLIA128-SHA
グループC	ECDH-ECDSA-AES128-GCM-SHA256
	ECDH-RSA-AES128-GCM-SHA256
	ECDH-ECDSA-CAMELLIA128-GCM-SHA256
	ECDH-RSA-CAMELLIA128-GCM-SHA256
	ECDH-ECDSA-AES128-SHA256
	ECDH-RSA-AES128-SHA256
	ECDH-ECDSA-CAMELLIA128-SHA256
	ECDH-RSA-CAMELLIA128-SHA256
	ECDH-ECDSA-AES128-SHA
	ECDH-RSA-AES128-SHA

グループD	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	DHE-DSS-AES256-SHA256
	DHE-RSA-AES256-SHA256
	DHE-DSS-CAMELLIA256-SHA256
	DHE-RSA-CAMELLIA256-SHA256
	DHE-DSS-AES256-SHA
	DHE-RSA-AES256-SHA
	DHE-DSS-CAMELLIA256-SHA
	DHE-RSA-CAMELLIA256-SHA
	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-SHA384
	ECDHE-RSA-AES256-SHA384
	ECDHE-ECDSA-CAMELLIA256-SHA384
	ECDHE-RSA-CAMELLIA256-SHA384
	ECDHE-ECDSA-AES256-SHA
	ECDHE-RSA-AES256-SHA
グループE	AES256-GCM-SHA384
	CAMELLIA256-GCM-SHA384
	AES256-SHA256
	CAMELLIA256-SHA256
	AES256-SHA
	CAMELLIA256-SHA
グループF	ECDH-ECDSA-AES256-GCM-SHA384
	ECDH-RSA-AES256-GCM-SHA384
	ECDH-ECDSA-CAMELLIA256-GCM-SHA384
	ECDH-RSA-CAMELLIA256-GCM-SHA384
	ECDH-ECDSA-AES256-SHA384
	ECDH-RSA-AES256-SHA384
	ECDH-ECDSA-CAMELLIA256-SHA384
	ECDH-RSA-CAMELLIA256-SHA384
	ECDH-ECDSA-AES256-SHA
	ECDH-RSA-AES256-SHA

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】

		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		
	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）鍵長2048ビットで、ハッシュ関数にSHA-256またはSHA-1を利用するRSAのサーバ証明書（SHA256withRSAまたはSHA1withRSA）を利用したか	5.1節	☐
	③-2）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	☐
	③-3）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	☐
	③-4）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／6.5.2節	☐
	④-i-2）表3記載のグループA及びグループBそれぞれの暗号スイート（網掛けを除く）から少なくとも一つずつは設定したか	6.1節／6.5.2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／6.5.2節	☐
	☐ ④-i-4）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-5）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	☐
	④-i-6）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／6.5.2節	☐
	④-i-7）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／6.5.2節	☐
	④-i-8）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／6.5.2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／6.5.2節	☐
	④-ii-3）表3記載のグループA及びグループBそれぞれの暗号スイート（網掛けを含む）から少なくとも一つずつは設定したか	6.1節／6.5.2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／6.5.2節	☐
		（続く）	

□	④-ii-5) 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-6) AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	□
	④-ii-7) 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／6.5.2節	□
	④-ii-8) 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／6.5.2節	□
	④-ii-9) 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／6.5.2節	□

【表3】

グループA	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	DHE-DSS-AES128-SHA256
	DHE-RSA-AES128-SHA256
	DHE-DSS-CAMELLIA128-SHA256
	DHE-RSA-CAMELLIA128-SHA256
	DHE-DSS-AES128-SHA
	DHE-RSA-AES128-SHA
	DHE-DSS-CAMELLIA128-SHA
	DHE-RSA-CAMELLIA128-SHA
	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-SHA256
	ECDHE-RSA-AES128-SHA256
	ECDHE-ECDSA-CAMELLIA128-SHA256
	ECDHE-RSA-CAMELLIA128-SHA256
	ECDHE-ECDSA-AES128-SHA
	ECDHE-RSA-AES128-SHA
グループB	AES128-GCM-SHA256
	CAMELLIA128-GCM-SHA256
	AES128-SHA256
	CAMELLIA128-SHA256
	AES128-SHA
	CAMELLIA128-SHA
グループC	ECDH-ECDSA-AES128-GCM-SHA256
	ECDH-RSA-AES128-GCM-SHA256
	ECDH-ECDSA-CAMELLIA128-GCM-SHA256
	ECDH-RSA-CAMELLIA128-GCM-SHA256
	ECDH-ECDSA-AES128-SHA256
	ECDH-RSA-AES128-SHA256
	ECDH-ECDSA-CAMELLIA128-SHA256
	ECDH-RSA-CAMELLIA128-SHA256
	ECDH-ECDSA-AES128-SHA
	ECDH-RSA-AES128-SHA

グループD	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	DHE-DSS-AES256-SHA256
	DHE-RSA-AES256-SHA256
	DHE-DSS-CAMELLIA256-SHA256
	DHE-RSA-CAMELLIA256-SHA256
	DHE-DSS-AES256-SHA
	DHE-RSA-AES256-SHA
	DHE-DSS-CAMELLIA256-SHA
	DHE-RSA-CAMELLIA256-SHA
	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-SHA384
	ECDHE-RSA-AES256-SHA384
	ECDHE-ECDSA-CAMELLIA256-SHA384
	ECDHE-RSA-CAMELLIA256-SHA384
	ECDHE-ECDSA-AES256-SHA
	ECDHE-RSA-AES256-SHA
グループE	AES256-GCM-SHA384
	CAMELLIA256-GCM-SHA384
	AES256-SHA256
	CAMELLIA256-SHA256
	AES256-SHA
	CAMELLIA256-SHA
グループF	ECDH-ECDSA-AES256-GCM-SHA384
	ECDH-RSA-AES256-GCM-SHA384
	ECDH-ECDSA-CAMELLIA256-GCM-SHA384
	ECDH-RSA-CAMELLIA256-GCM-SHA384
	ECDH-ECDSA-AES256-SHA384
	ECDH-RSA-AES256-SHA384
	ECDH-ECDSA-CAMELLIA256-SHA384
	ECDH-RSA-CAMELLIA256-SHA384
	ECDH-ECDSA-AES256-SHA
	ECDH-RSA-AES256-SHA
グループG	DHE-DSS-RC4-SHA
	RC4-SHA
グループH	DHE-DSS-DES-CBC3-SHA
	DHE-RSA-DES-CBC3-SHA
	DES-CBC3-SHA

Appendix B：サーバ設定編

B. 1. 鍵パラメータファイルの設定方法例

B.1.1. OpenSSL による DH、DHE、ECDH、ECDHE 鍵パラメータファイルの生成

OpenSSL コマンドにより、DH、DHE の 2048 ビットの鍵パラメータファイルを生成するには以下を実行する。

```
openssl dhparam -out dh2048.pem -outform PEM -2 2048
```

また、ECDH, ECDHE 鍵パラメータファイル（521 ビット素体）は以下のようにして生成することができる。

```
openssl ecparam -out secp521r1.pem -name secp521r1
```

B.1.2. Apache における DH、DHE、ECDH、ECDHE 鍵パラメータ設定

Apache 2.4.7 以降では、DH、DHE、ECDH、ECDHE の鍵パラメータファイルを明示的に指定することができる。

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だが、DH、DHE、ECDH、ECDHE の鍵長を示すパラメータファイルを指定することができる。以下に例を示す。

サーバー証明書の指定例

```
SSLCertificateFile /etc/httpd/conf/ssl/server.crt
```

DH、DHE 鍵パラメータファイルの指定例

```
SSLCertificateFile /etc/httpd/conf/ssl/dh2048.pem
```

ECDH, ECDHE 鍵パラメータファイルの指定例

```
SSLCertificateFile /etc/httpd/conf/ssl/secp521r1.pem
```

B.1.3. lighttpd における DH、DHE、ECDH、ECDHE 鍵パラメータ設定

lighttpd では以下のように設定する。

DH、DHE の鍵パラメータファイルの指定例

```
ssl.dh-file = "/etc/lighttpd/ssl/dh2048.pem"
```

ECDH、ECDHE の楕円曲線パラメータの指定例

```
ssl.ec-curve = "secp256r1"
```

B.1.4. nginx における DH、DHE、ECDH、ECDHE 鍵パラメータ設定

nginx では以下のように設定する。

DH、DHE の鍵パラメータファイルの指定例

```
http { ...
  server {
    ssl_dhparam /etc/nginx/ssl/dh2048.pem;
  } ...
}
```

ECDH、ECDHE の楕円曲線パラメータの指定例

```
http { ...
  server {
    ssl_ecdh_curve secp256r1;
  } ...
}
```

B. 2. OCSP Stapling の設定方法例

B.2.1. Apache 2.3.3 以降における設定

Apache 2.3.3 での設定例を以下に示す。

```
SSLStaplingCache shmcb:/tmp/stapling_cache(128000)
<VirtualHost *:443>
...中略...
SSLCACertificateFile /etc/ssl/ca-certs.pem
SSLUseStapling on
</VirtualHost>
```

B.2.2. nginx 1.3.7 以降における設定

nginx 1.3.7 以降での設定例を以下に示す。

```
server {
  ... 中略 ...
  ssl_stapling on;
```

```
    ssl_stapling_verify on;
    ssl_trusted_certificate /etc/ssl/ca-certs.pem;
}
```

B.2.3. Microsoft IIS における設定

Windows Server 2008 以降の IIS ではデフォルトで OCSP Stapling が設定されている。
それ以外では、レジストリキーにより以下のように設定する。

HKEY_LOCAL_MACHINES¥SYSTEM¥CurrentControlSet¥Control¥Lsa¥Kerberos¥Parameters¥RequestOCSP

B. 3. 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="QtXc8+scL7K6HiPksQ8mqIyY08Xdc4Z5raHT+xSh9/s=";
pin-sha256="kb6xLprt35abNnSn74my4Dkfya9arbk5zN5a60YzuqE="; max-age=3000
Public-Key-Pins: pin-sha1="FhxvMPhD7Q+byiiwygLO0mL7L70=";
pin-sha1="KqqJgAYLy9ogXOWETcR36ioKf20="; max-age=3000
```

例に示す通り、

- エンドエンティティ（SSL サーバー証明書）から、最上位の中間証明書まで全てを列挙する
- ハッシュ値を計算するハッシュアルゴリズム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
- max-age により有効期間を指定する

B.3.1. ハッシュ値や HTTP ヘッダの計算方法

ハッシュ値の計算方法はいくつかあるが、PEM 形式のサーバ証明書を入力することより、Public-Key-Pins ヘッダを自動作成できるサイトがあるので、これを活用すると簡単に計算できる。

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

B.3.2. Apache の場合

mod_headers モジュールを有効にし、以下の設定を追加する。

```
Header add Public-Key-Pins 'pin-sha256="QtXc8+scL7K6HiPksQ8mqIyY08Xdc4Z5raHT+xSh9/s=";
pin-sha256="kb6xLprt35abNnSn74my4Dkfya9arbk5zN5a60YzuqE="; max-age=3000'
Header add Public-Key-Pins 'pin-sha1="FhxvMPhD7Q+byiiwygLO0mL7L70=";
pin-sha1="KqqJgAYLy9ogXOWETcR36ioKf20="; max-age=3000'
```

B.3.3. nginx の場合

設定ファイルにおいて、add_header により、ヘッダを追加する。

```
add_header Public-Key-Pins 'pin-sha256="QtXc8+scL7K6HiPksQ8mqIyY08Xdc4Z5raHT+xSh9/s=";
pin-sha256="kb6xLprt35abNnSn74my4Dkfya9arbk5zN5a60YzuqE="; max-age=3000';
add_header Public-Key-Pins 'pin-sha1="FhxvMPhD7Q+byiiwygLO0mL7L70=";
pin-sha1="KqqJgAYLy9ogXOWETcR36ioKf20="; max-age=3000';
```

B.3.4. IIS の場合

IIS では任意の HTTP ヘッダを追加することが可能である。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 レスポンス ヘッダ」をダブルクリックする
4) 「アクション」のペインで「追加」をクリックする
5) 「ヘッダ名」「値」の箇所を、上述の設定を参考に設定する。
6) 「OK」をクリックする。

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

B.4.1.Apache HTTP Server 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。

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

HTTP の場合に強制的に HTTPS にリダイレクトするために、以下のように設定を追記する。

```
<VirtualHost *:80>
    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、設定ファイルに以下の記述を追加する。

```
setenv.add-response-header = (
    "Strict-Transport-Security" => ""
)
```

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。

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

B.4.4.Microsoft IIS の場合

```
<outboundRules>
    <rule name="SSLStrictTransportSecurity" preCondition="IsSSLReq">
        <match serverVariable="RESPONSE_Strict_Transport_Security" pattern=".*" />
        <action type="Rewrite" value="max-age=3600; includeSubDomains" />
    </rule>
    <preConditions>
        <preCondition name="IsSSLReq">
            <add input="{HTTPS}" pattern="on" />
        </preCondition>
    </preConditions>
</outboundRules>
```

【TBD】

※対象は要検討

例：

openssl 系のコマンドを使用して鍵生成する場合には他のユーザの鍵と重複することが無いように乱数生成器を使用することを推奨する。

(例)

Unix 系の場合

```
% openssl genrsa -rand /dev/random -out rsa2048.key 2048
```

```
% openssl genrsa -rand /dev/urandom -out rsa2048.key 2048
```

Windows 系の場合、適当なユーザ文書やログのあるディレクトリのファイルから乱数の種を生成して使用する。

```
% openssl sha512 *.* > rand.dat
% openssl genrsa -rand rand.dat -out rsa2048.key 2048
```

Appendix C：ブラウザ設定編

C.1. ブラウザ設定のリセット方法

【Internet Explorer】

1. “ツール”メニューの“インターネット オプション”をクリックする。“ツール” メニューが表示されていない場合は、Alt キーを押す。
2. “インターネット オプション”ウィンドウの“詳細設定”タブをクリックする。
3. “詳細設定を復元”をクリックする。
4. “リセット”をクリックする。Windows Internet Explorer 6 を使用している場合は、“標準に戻す”をクリックします。
5. “Internet Explorer の設定のリセット”ダイアログボックスで、“リセット”をクリックする。
   注） 閲覧の履歴、検索プロバイダー、アクセラレータ、ホーム ページ、追跡防止、および ActiveX フィルタのデータも削除する場合は、“個人設定を削除する”チェックボックスをオンにする。
6. Internet Explorer で既定の設定の適用が完了したら、“閉じる”をクリックし、“OK”をクリックします。
7. Internet Explorer を終了してから再起動する。

参考

http://support.microsoft.com/kb/923737/ja

http://support.microsoft.com/kb/2539155/ja

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

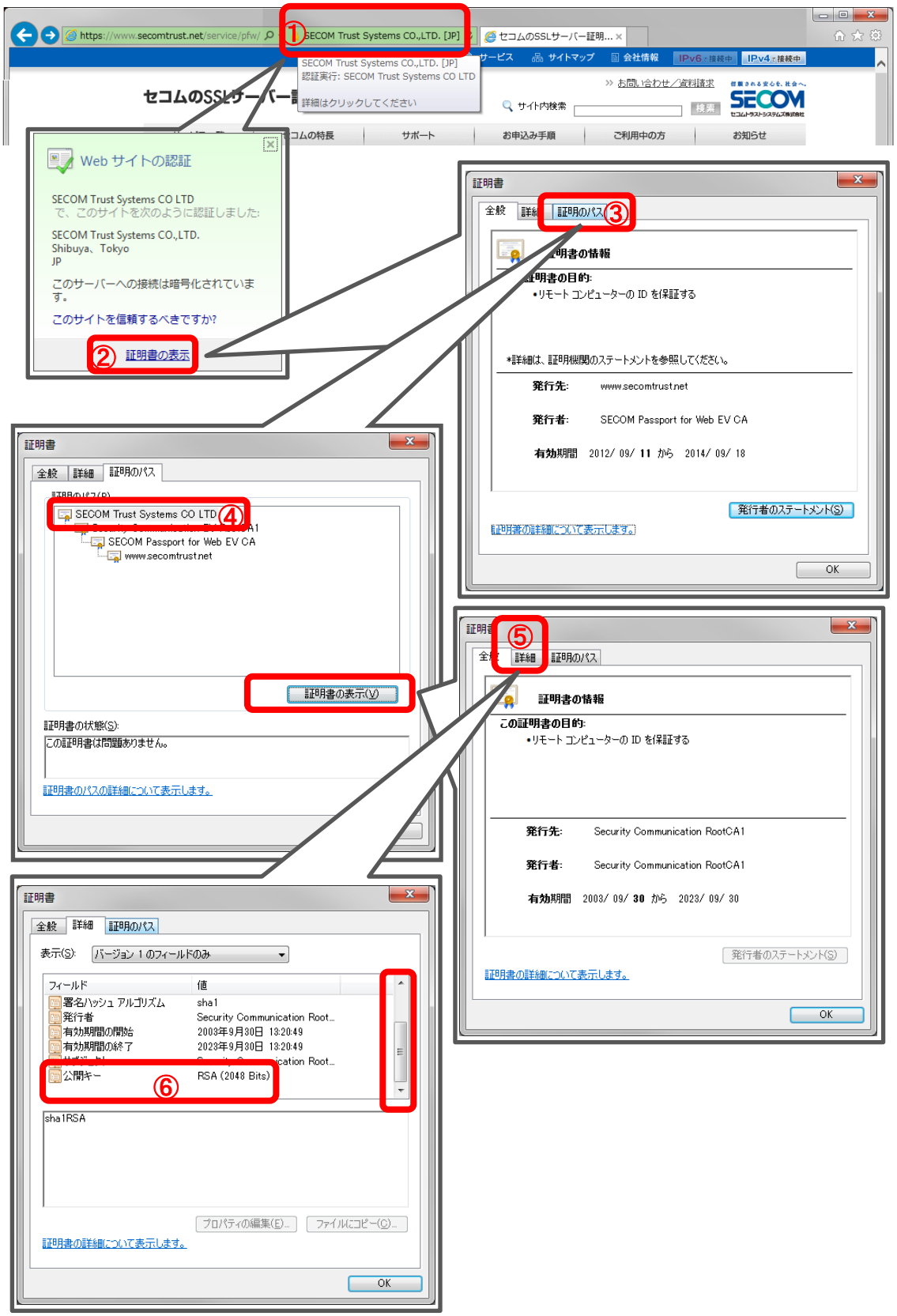
主要な認証事業者のルート CA 証明書の暗号アルゴリズムおよび鍵長を別表に掲載する。

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。

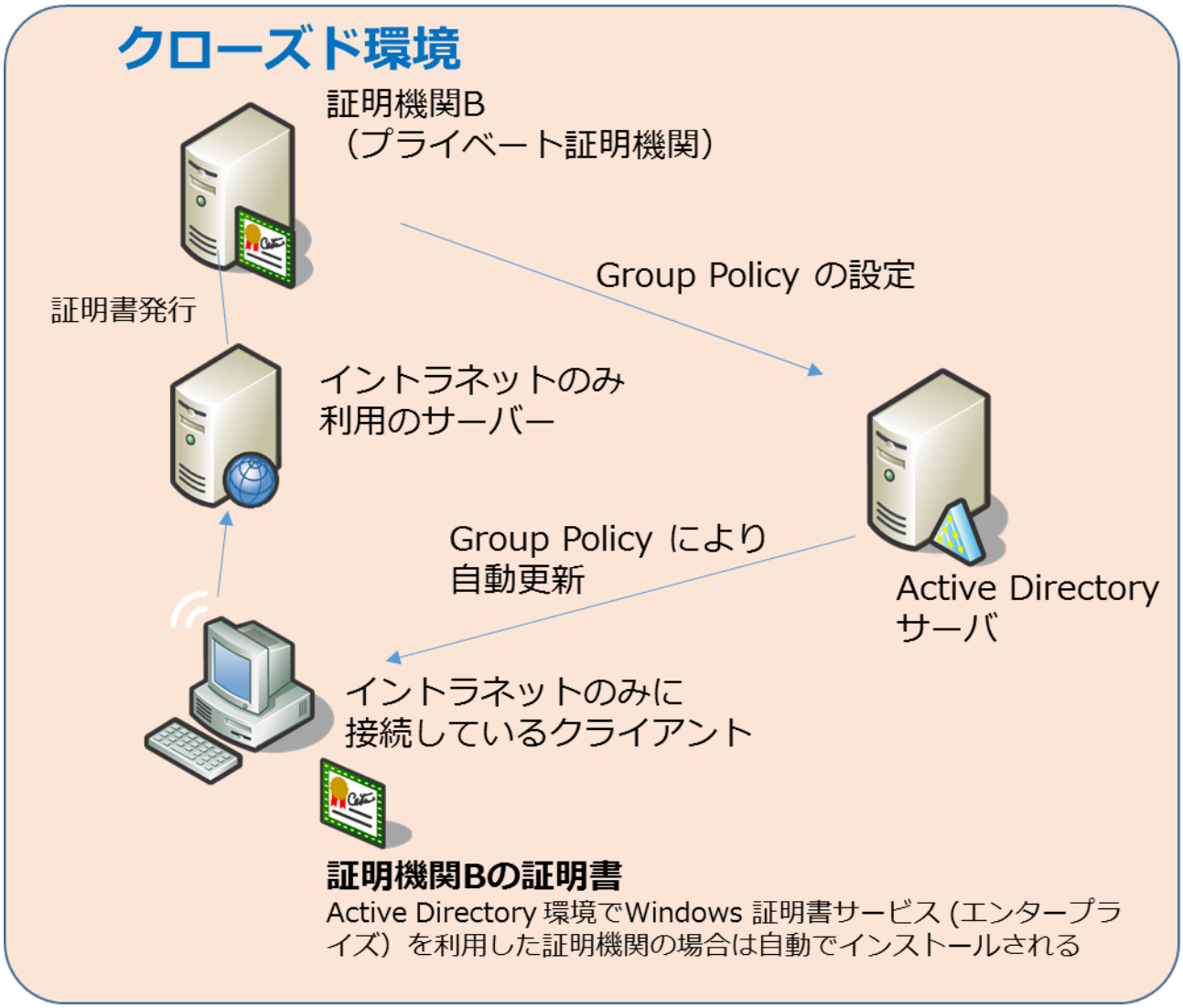
なお､サーバ証明書を発行するサービスから発行された既存のサーバ証明書を利用したサイト、あるいはテストサイトなどの URL がわかっている場合には、当該 URL にアクセスして、以下のような手順を経ることで、ルート CA の公開鍵暗号アルゴリズムおよび鍵長を確認することが可能である。

【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 2048bit 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 E：暗号スイートの設定例

資料２
別添３

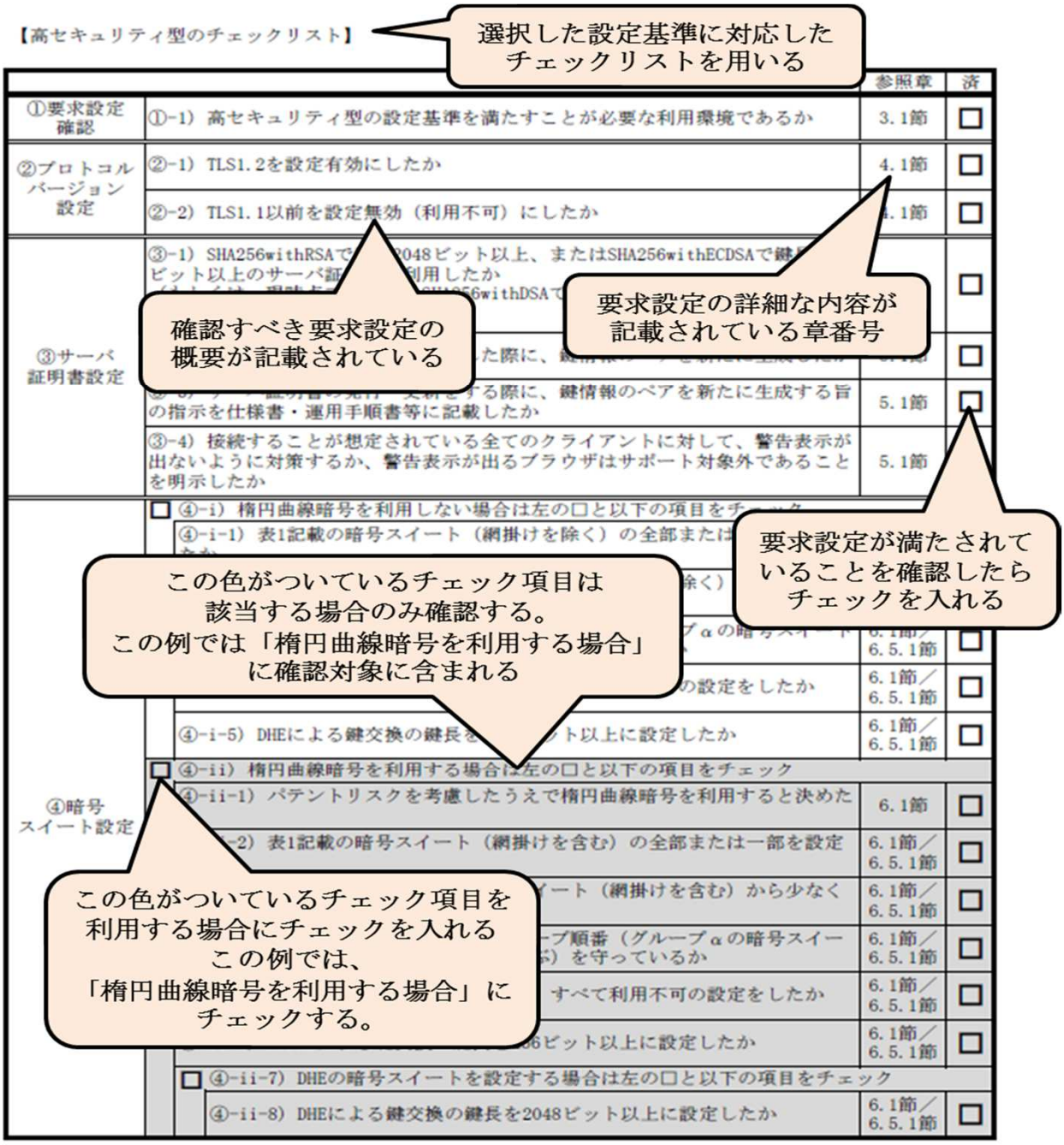
SSL/TLS暗号設定ガイドラインチェックリスト

【チェックリストの使い方】

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。

◇ プロトコルバージョンの設定（ガイドライン第4章）
◇ サーバ証明書の設定（ガイドライン第5章）
◇ 暗号スイートの設定（ガイドライン第6章）

＜チェックリストの例＞



【高セキュリティ型のチェックリスト】

		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）SHA256withRSAで鍵長2048ビット以上、またはSHA256withECDSAで鍵長256ビット以上のサーバ証明書を利用したか （もしくは、現時点で利用中のSHA256withDSAで鍵長2048ビット以上のサーバ証明書をそのまま継続利用したか）	5.1節	□
	③-2）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	□
	③-3）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	□
	③-4）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】

グループα	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
グループβ	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256

【推奨セキュリティ型のチェックリスト】

		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）SHA256withRSAで鍵長2048ビット以上、またはSHA256withECDSAで鍵長256ビット以上のサーバ証明書を利用したか （もしくは、現時点で利用中のSHA256withDSAで鍵長2048ビット以上のサーバ証明書をそのまま継続利用したか）	5.1節	□
	③-2）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	□
	③-3）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	□
	③-4）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それぞれの暗号スイート（網掛けを除く）から少なくとも一つずつは設定したか	6.1節／6.5.2節	□
	□ ④-i-3）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-4）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	□
	④-i-5）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-6）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-7）DHEの鍵長が明示的に設定可能な場合は左の□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／6.5.1節	□
	④-i-9）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／6.5.1節	□

（続く）

④暗号 スイート設定	☐ 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それぞれの暗号スイート（網掛けを含む）から少なくとも一つずつは設定したか	6.1節／6.5.2節	☐
	☐ ④-ii-4）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-5）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	☐
	④-ii-5）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-6）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-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／6.5.1節	☐
	☐ ④-ii-8）DHEを利用する暗号スイートを設定する場合であって、DHEの鍵長が明示的に設定可能な場合は左の□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／6.5.1節	☐
	④-ii-10）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／6.5.1節	☐

【表2】

グループA	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	DHE-DSS-AES128-SHA256
	DHE-RSA-AES128-SHA256
	DHE-DSS-CAMELLIA128-SHA256
	DHE-RSA-CAMELLIA128-SHA256
	DHE-DSS-AES128-SHA
	DHE-RSA-AES128-SHA
	DHE-DSS-CAMELLIA128-SHA
	DHE-RSA-CAMELLIA128-SHA
	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-SHA256
	ECDHE-RSA-AES128-SHA256
	ECDHE-ECDSA-CAMELLIA128-SHA256
	ECDHE-RSA-CAMELLIA128-SHA256
	ECDHE-ECDSA-AES128-SHA
	ECDHE-RSA-AES128-SHA
グループB	AES128-GCM-SHA256
	CAMELLIA128-GCM-SHA256
	AES128-SHA256
	CAMELLIA128-SHA256
	AES128-SHA
	CAMELLIA128-SHA
グループC	ECDH-ECDSA-AES128-GCM-SHA256
	ECDH-RSA-AES128-GCM-SHA256
	ECDH-ECDSA-CAMELLIA128-GCM-SHA256
	ECDH-RSA-CAMELLIA128-GCM-SHA256
	ECDH-ECDSA-AES128-SHA256
	ECDH-RSA-AES128-SHA256
	ECDH-ECDSA-CAMELLIA128-SHA256
	ECDH-RSA-CAMELLIA128-SHA256
	ECDH-ECDSA-AES128-SHA
	ECDH-RSA-AES128-SHA

グループD	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	DHE-DSS-AES256-SHA256
	DHE-RSA-AES256-SHA256
	DHE-DSS-CAMELLIA256-SHA256
	DHE-RSA-CAMELLIA256-SHA256
	DHE-DSS-AES256-SHA
	DHE-RSA-AES256-SHA
	DHE-DSS-CAMELLIA256-SHA
	DHE-RSA-CAMELLIA256-SHA
	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-SHA384
	ECDHE-RSA-AES256-SHA384
	ECDHE-ECDSA-CAMELLIA256-SHA384
	ECDHE-RSA-CAMELLIA256-SHA384
	ECDHE-ECDSA-AES256-SHA
	ECDHE-RSA-AES256-SHA
グループE	AES256-GCM-SHA384
	CAMELLIA256-GCM-SHA384
	AES256-SHA256
	CAMELLIA256-SHA256
	AES256-SHA
	CAMELLIA256-SHA
グループF	ECDH-ECDSA-AES256-GCM-SHA384
	ECDH-RSA-AES256-GCM-SHA384
	ECDH-ECDSA-CAMELLIA256-GCM-SHA384
	ECDH-RSA-CAMELLIA256-GCM-SHA384
	ECDH-ECDSA-AES256-SHA384
	ECDH-RSA-AES256-SHA384
	ECDH-ECDSA-CAMELLIA256-SHA384
	ECDH-RSA-CAMELLIA256-SHA384
	ECDH-ECDSA-AES256-SHA
	ECDH-RSA-AES256-SHA

【セキュリティ例外型のチェックリスト】

		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		
	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）鍵長2048ビットで、ハッシュ関数にSHA-256またはSHA-1を利用するRSAのサーバ証明書（SHA256withRSAまたはSHA1withRSA）を利用したか	5.1節	□
	③-2）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	□
	③-3）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	□
	③-4）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／6.5.2節	□
	④-i-2）表3記載のグループA及びグループBそれぞれの暗号スイート（網掛けを除く）から少なくとも一つずつは設定したか	6.1節／6.5.2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／6.5.2節	□
	□ ④-i-4）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-5）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	□
	④-i-6）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／6.5.2節	□
	④-i-7）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／6.5.2節	□
	④-i-8）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／6.5.2節	□
	□ ④-i-9）DHEの鍵長が明示的に設定可能な場合は左の□と以下の項目をチェック		
	④-i-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／6.5.1節	□
	④-i-11）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／6.5.1節	□

（続く）

④暗号スイート設定	☐ 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／6.5.2節	☐
	④-ii-3）表3記載のグループA及びグループBそれぞれの暗号スイート（網掛けを含む）から少なくとも一つずつは設定したか	6.1節／6.5.2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／6.5.2節	☐
	☐ ④-ii-5）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-6）AES128-SHAの暗号スイートを設定したか	6.1節／6.5.2節	☐
	④-ii-7）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／6.5.2節	☐
	④-ii-8）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／6.5.2節	☐
	④-ii-9）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／6.5.2節	☐
	④-ii-10）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／6.5.1節	☐
	☐ ④-ii-11）DHEを利用する暗号スイートを設定する場合であって、DHEの鍵長が明示的に設定可能な場合は左の□と以下の項目をチェック		
	④-ii-12）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／6.5.1節	☐
	④-ii-13）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／6.5.1節	☐

【表3】

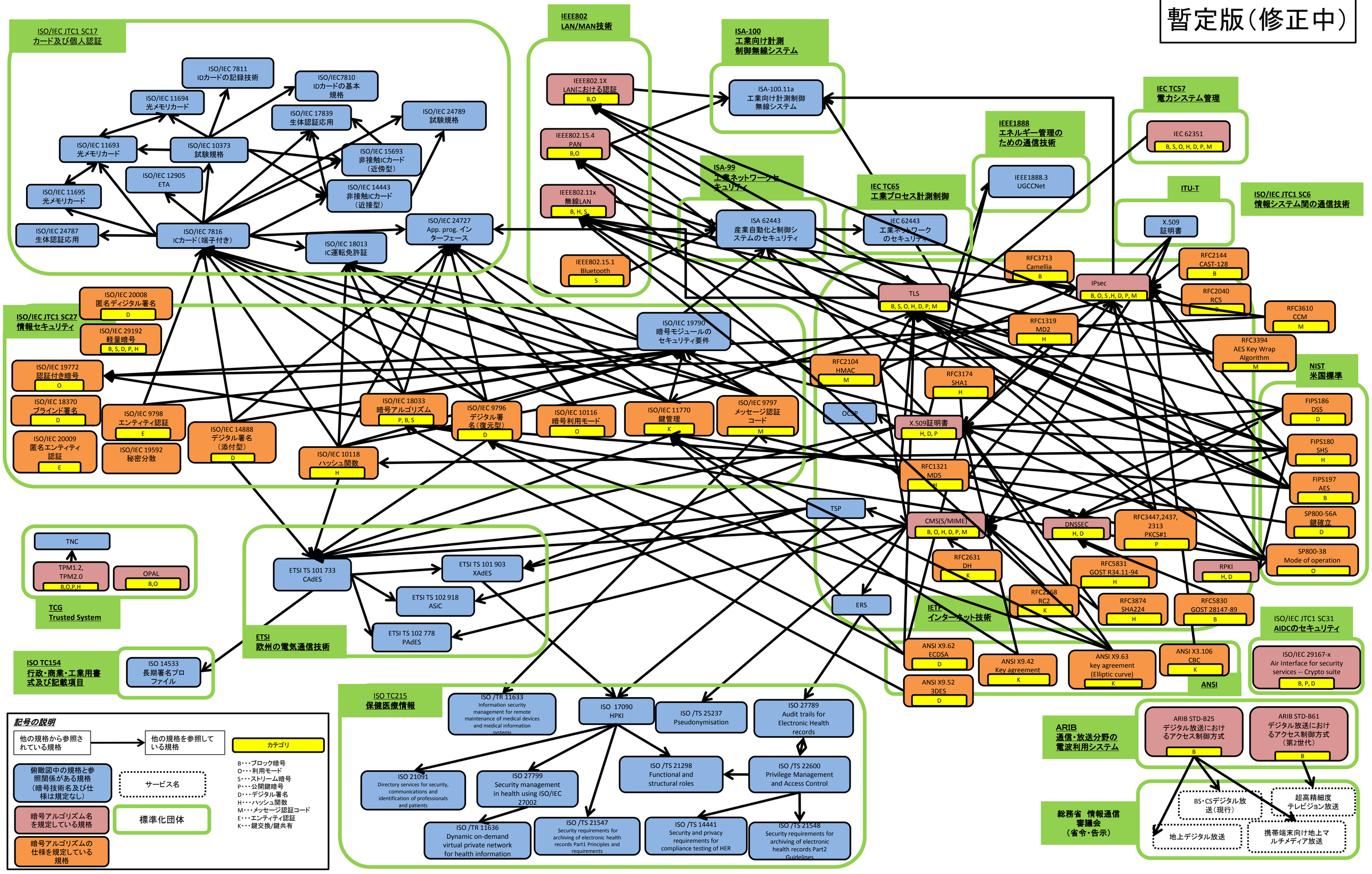
グループA	DHE-DSS-AES128-GCM-SHA256
	DHE-RSA-AES128-GCM-SHA256
	DHE-DSS-CAMELLIA128-GCM-SHA256
	DHE-RSA-CAMELLIA128-GCM-SHA256
	DHE-DSS-AES128-SHA256
	DHE-RSA-AES128-SHA256
	DHE-DSS-CAMELLIA128-SHA256
	DHE-RSA-CAMELLIA128-SHA256
	DHE-DSS-AES128-SHA
	DHE-RSA-AES128-SHA
	DHE-DSS-CAMELLIA128-SHA
	DHE-RSA-CAMELLIA128-SHA
	ECDHE-ECDSA-AES128-GCM-SHA256
	ECDHE-RSA-AES128-GCM-SHA256
	ECDHE-ECDSA-CAMELLIA128-GCM-SHA256
	ECDHE-RSA-CAMELLIA128-GCM-SHA256
	ECDHE-ECDSA-AES128-SHA256
	ECDHE-RSA-AES128-SHA256
	ECDHE-ECDSA-CAMELLIA128-SHA256
	ECDHE-RSA-CAMELLIA128-SHA256
	ECDHE-ECDSA-AES128-SHA
	ECDHE-RSA-AES128-SHA
グループB	AES128-GCM-SHA256
	CAMELLIA128-GCM-SHA256
	AES128-SHA256
	CAMELLIA128-SHA256
	AES128-SHA
	CAMELLIA128-SHA
グループC	ECDH-ECDSA-AES128-GCM-SHA256
	ECDH-RSA-AES128-GCM-SHA256
	ECDH-ECDSA-CAMELLIA128-GCM-SHA256
	ECDH-RSA-CAMELLIA128-GCM-SHA256
	ECDH-ECDSA-AES128-SHA256
	ECDH-RSA-AES128-SHA256
	ECDH-ECDSA-CAMELLIA128-SHA256
	ECDH-RSA-CAMELLIA128-SHA256
	ECDH-ECDSA-AES128-SHA
	ECDH-RSA-AES128-SHA

グループD	DHE-DSS-AES256-GCM-SHA384
	DHE-RSA-AES256-GCM-SHA384
	DHE-DSS-CAMELLIA256-GCM-SHA384
	DHE-RSA-CAMELLIA256-GCM-SHA384
	DHE-DSS-AES256-SHA256
	DHE-RSA-AES256-SHA256
	DHE-DSS-CAMELLIA256-SHA256
	DHE-RSA-CAMELLIA256-SHA256
	DHE-DSS-AES256-SHA
	DHE-RSA-AES256-SHA
	DHE-DSS-CAMELLIA256-SHA
	DHE-RSA-CAMELLIA256-SHA
	ECDHE-ECDSA-AES256-GCM-SHA384
	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ECDSA-CAMELLIA256-GCM-SHA384
	ECDHE-RSA-CAMELLIA256-GCM-SHA384
	ECDHE-ECDSA-AES256-SHA384
	ECDHE-RSA-AES256-SHA384
	ECDHE-ECDSA-CAMELLIA256-SHA384
	ECDHE-RSA-CAMELLIA256-SHA384
	ECDHE-ECDSA-AES256-SHA
	ECDHE-RSA-AES256-SHA
グループE	AES256-GCM-SHA384
	CAMELLIA256-GCM-SHA384
	AES256-SHA256
	CAMELLIA256-SHA256
	AES256-SHA
	CAMELLIA256-SHA
グループF	ECDH-ECDSA-AES256-GCM-SHA384
	ECDH-RSA-AES256-GCM-SHA384
	ECDH-ECDSA-CAMELLIA256-GCM-SHA384
	ECDH-RSA-CAMELLIA256-GCM-SHA384
	ECDH-ECDSA-AES256-SHA384
	ECDH-RSA-AES256-SHA384
	ECDH-ECDSA-CAMELLIA256-SHA384
	ECDH-RSA-CAMELLIA256-SHA384
	ECDH-ECDSA-AES256-SHA
	ECDH-RSA-AES256-SHA
グループG	DHE-DSS-RC4-SHA
	RC4-SHA
グループH	DHE-DSS-DES-CBC3-SHA
	DHE-RSA-DES-CBC3-SHA
	DES-CBC3-SHA

暗号技術参照関係の俯瞰図（全体像）

資料2別添4-1

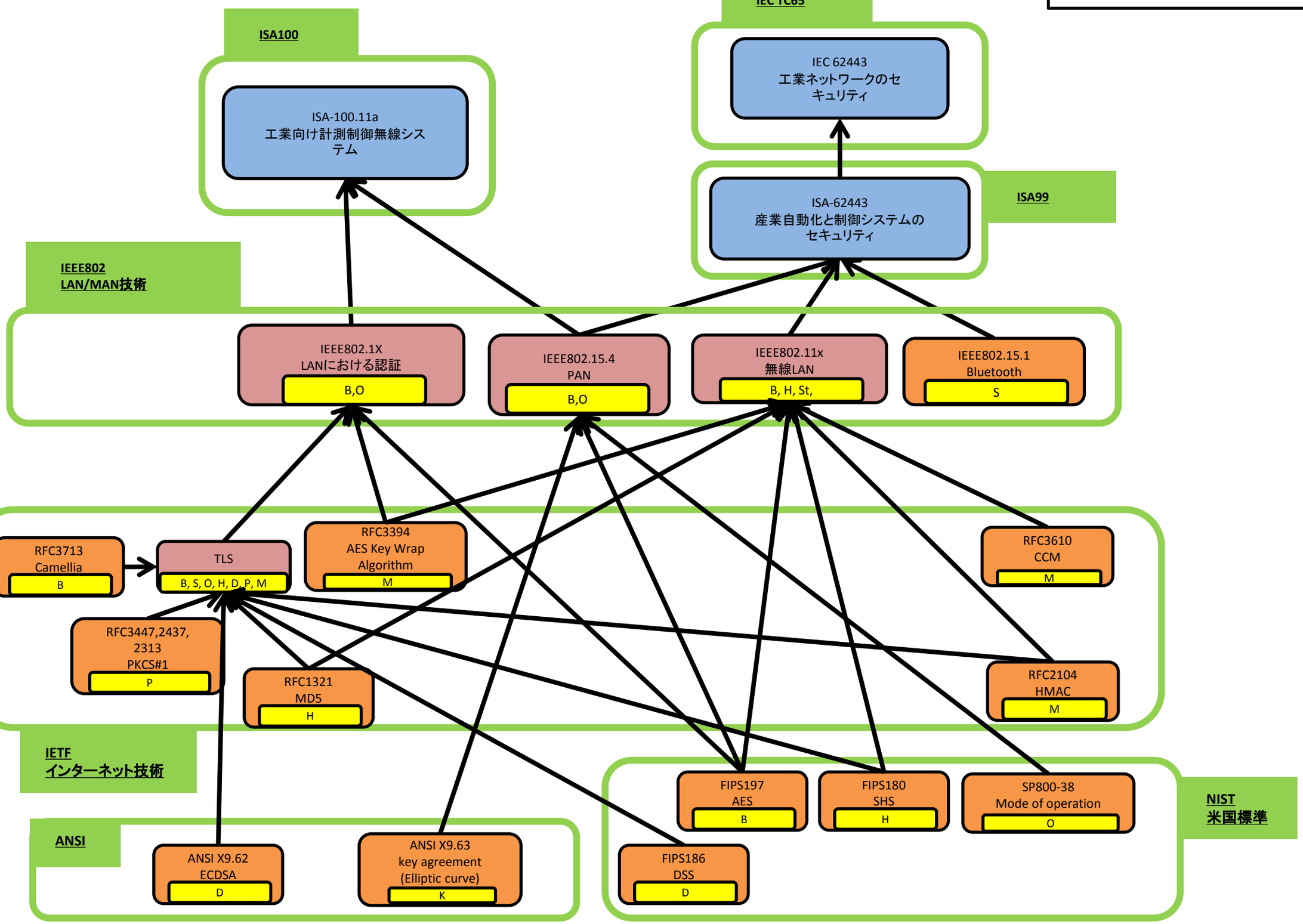
暫定版（修正中）



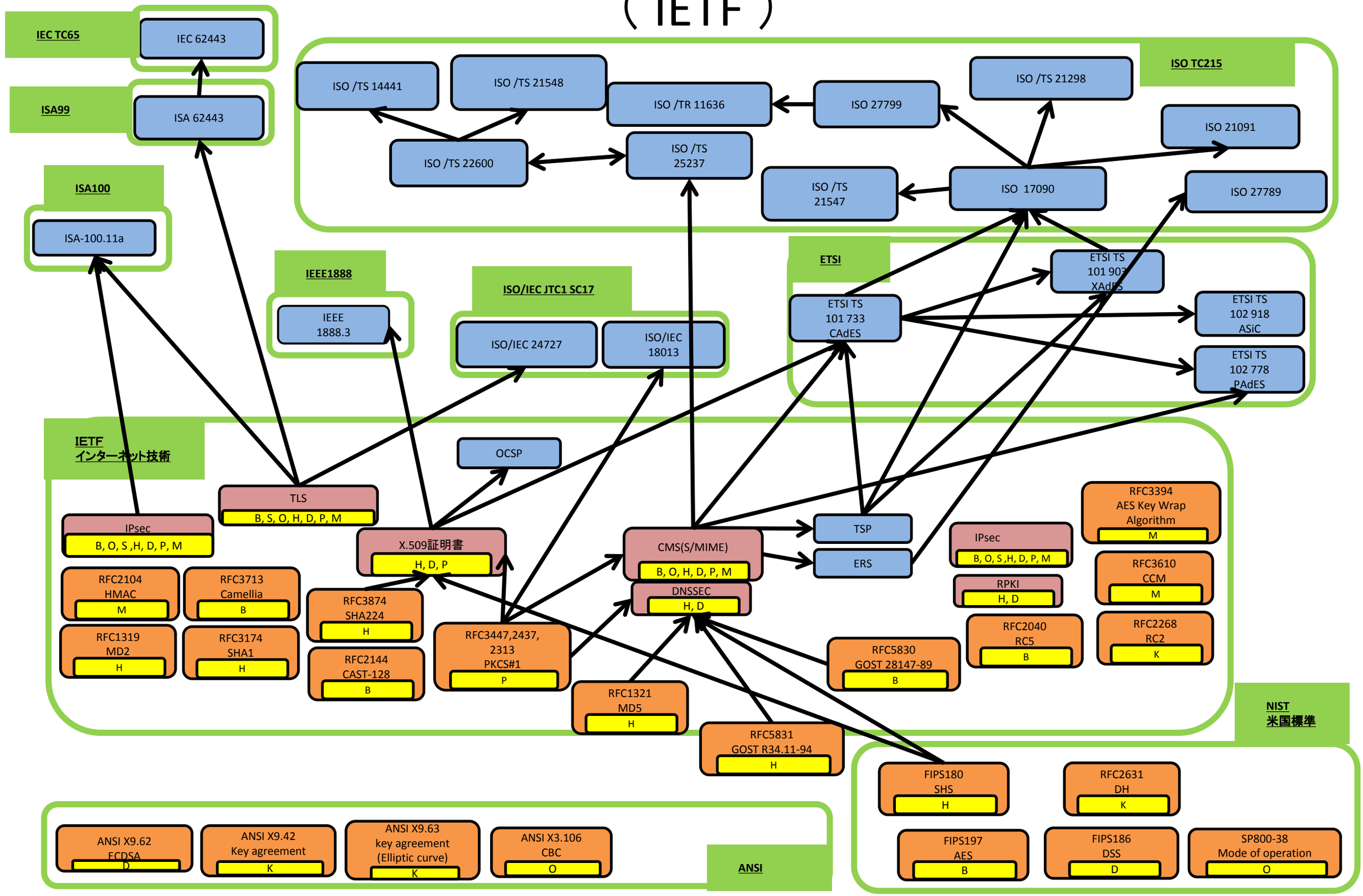
暗号技術参照関係の俯瞰図（IEEE802）

資料2別添4-2

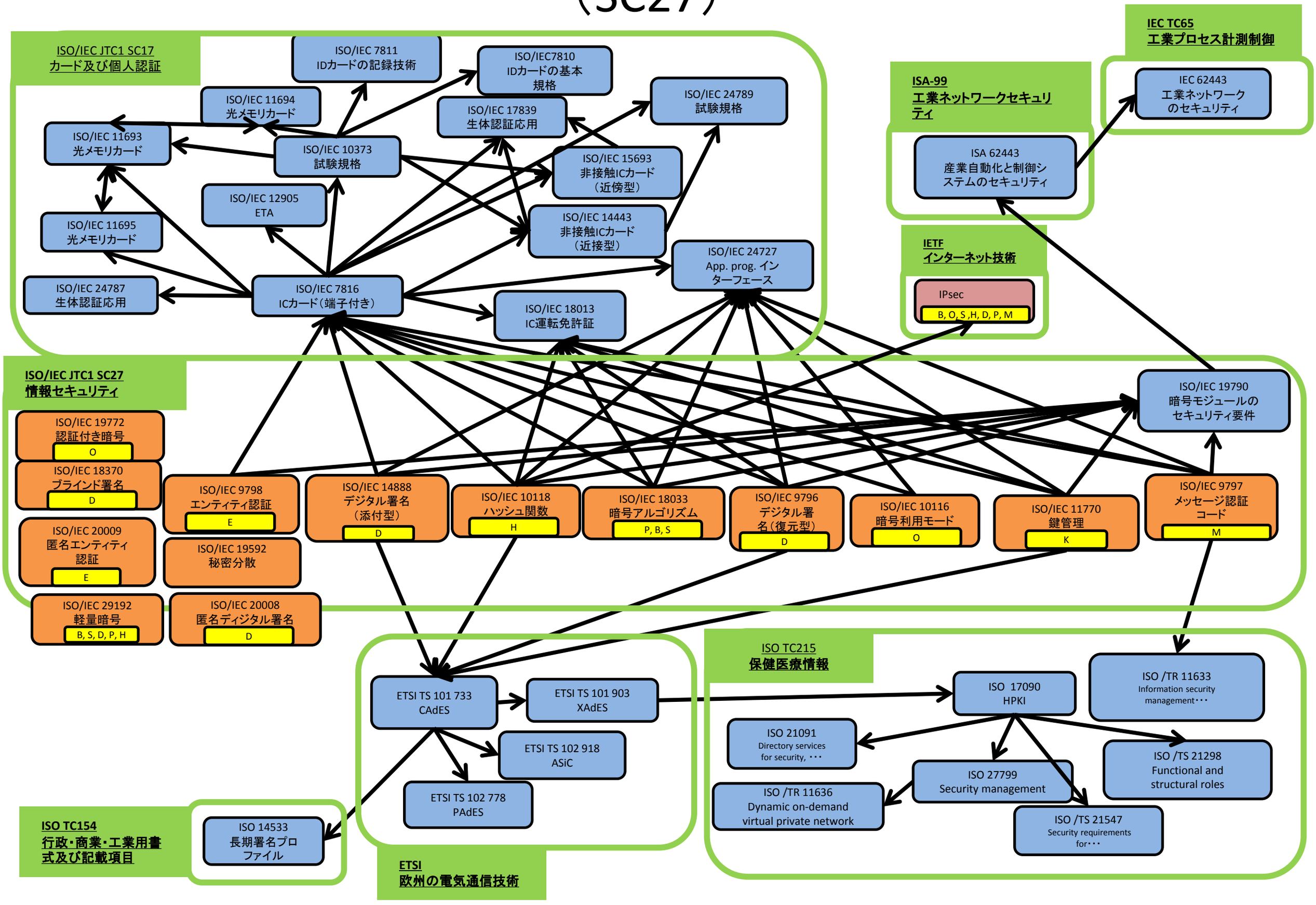
暫定版（修正中）



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）



資料 2
別添 5

[修正中版]

標準化提案におけるノウハウ・課題・基本的な情報の整理

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。

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 TC65/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・基本的な情報の整理」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。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。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・基本的な情報の整理」は、本 WG に集まっていただいた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) 団体毎の基本的な情報、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)団体毎の基本的な情報、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、団体毎の基本的な情報として、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含まれている。

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>について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）団体毎の基本的な情報、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<ノウハウ>及び<課題>については下記のとおりである。

<基本的な情報>

● 投票権を取る条件について
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について
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）

● 規格化のプロセスについて
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、Fast-Track の場合等がある。

● 規格策定のために利用するツールについて
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/ファストトラックには、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、<基本的な情報>、<ノウハウ>、及び<課題>は下記のとおりである。

<基本的な情報>

● 何年おきに規格の改定があるのかについて
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について書いてください。
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があれば、記載してください。
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
NB (National Body)提案とする場合、Expert 提案と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について
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について
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。

● 会議の開催場所について
各国からの提案ベースで決定されている。これまで新しい方から順にメキシコ、香港、韓国、フランス、イタリア、スウェーデン…の順に開催されてきた。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について
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について

<ノウハウ>

ノウハウ①
学術論文や国家的なプロジェクトでの評価が大きく信頼されているので、学術論文等で評価された技術であるとよい。

<課題>

課題①
海外では、巧く整理されているが、国内では、ベンダ同士で提案競争が起こっていて、国内調整に手間がかかる。

1.2. ISO/IEC JTC 1/SC 17

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

<基本的な情報>

● 何年おきに規格の改定があるのかについて
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（技術の募集）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について
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。

● 会議の開催場所について
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について
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について
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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 何年おきに規格の改定があるのかについて
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（技術の募集）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

● 技術提案の手続き等

● 会合が年何回あるか

● 電話会議の時間帯と時間帯の決め方について

● 会議の開催場所について

● 提案者に最低限必要となる作業について
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について

<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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 会議の開催場所について
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について

<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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 何年おきに規格の改定があるのかについて
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について
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について
ISO/TC215 会議の中で調整して決定する。

● 会議の開催場所について
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について
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について

● 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 JointWG 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・ISA99 関連

IEC/TC 65 / ISA99 について、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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について
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。**

● 投票権を取る条件について
SA 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について
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。

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

● 会議の開催場所について
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について
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→DIS→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について

<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<基本的な情報>は下記のとおりである。

<基本的な情報>

● 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。

● 投票権を取る条件について
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。

● 規格化のプロセスについて
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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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について
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について
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について
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について
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。

● 会議の開催場所について
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について
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について
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について
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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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について
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について
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について
2 年程度である。

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

● 会議の開催場所について
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について
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について
Microsoft Office Word

● 国内におけるコンタクト先について
JISC（日本工業規格調査会）

<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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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。

● 投票権を取る条件について
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について
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について
数年程度である。

● 電話会議の時間帯と時間帯の決め方について
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。

● 規格化のプロセスについて
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

● 会議の開催場所について
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について
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について
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について
JRF

● その他
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
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 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

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
  - 自動車
  - 電気・ガスなどのインフラ
  - 銀行・金融
- テクノロジー
  - IOT
  - Cloud
  - Embedded System

5. ETSI

ETSI について、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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について
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。

● 規格化のプロセスについて
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。

● 会議の開催場所について
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。

● 国内におけるコンタクト先について
JNSA

<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、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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について
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。

● 投票権を取る条件について
ARIB への入会が条件となる。

● 投票権は誰に帰属しているかについて
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について
通常は数年程度である。

● 規格化のプロセスについて
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。

● 会議の開催場所について
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について
※「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について

7. ISA 100

ISA100 について、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 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。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。

● 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。

● 投票権を取る条件について
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について
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について
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について

WG によって異なるが、概ね EU と米国の時間に合わせる。

● 会議の開催場所について
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について
Microsoft Office Word

● 国内におけるコンタクト先について

<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。

8. IETF

IETF について、<基本的な情報>、<ノウハウ>、<課題>は下記のとおりである。

<基本的な情報>

● 何年おきに規格の改定があるのかについて
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について
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について
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。

● 投票権を取る条件について
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、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について
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について
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について
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。

● 規格化のプロセスについて
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。

● 会議の開催場所について

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
draft-palet-ietf-meeting-venue-selection-criteria-04.txt

● 提案者に最低限必要となる作業について

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について

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について

国内では ISOC 日本支部によって 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。

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

資料３

CRYPTREC 暗号リストの注釈の一部変更について

運用監視暗号リストに掲載されている「128-bit RC4」の注釈について、暗号技術評価委員会及び暗号技術活用委員会において、「２．変更案」で示すとおり変更されるべきであるとの結論を得た。注釈変更の是非について、本検討会でご審議いただき、ご承認いただいた場合は、速やかに注釈の変更を実施する予定である。

１．背景

FSE 2001 において、Mantin と Shamir により、RC4 のキーストリームの 2 バイト目が統計的に偏っているという報告があった[1]。さらに、[2, 3]において、他の複数のバイトにおいても同様の傾向があると報告された。RC4 を使い放送型の暗号化通信を行った場合に、攻撃者は、この統計的偏りを利用し、平文を求めることが可能となる。ここで、放送型の暗号化通信とは、同じ平文を複数の異なるキーストリームで暗号化し、送受信することを指し、SSL/TLS においてもこのようなケースが起こりうる。実際に、USENIX Security 2013 において、AlFardan らはブラウザサイドスクリプト言語 JavaScript を用いることにより、このようなケースを引き起こせることを示した[3]。従って、[2, 3]により、SSL/TLS において、RC4 を攻撃する有効な手法が示されたこととなる。これらの背景を踏まえ、現在の RC4 に関する注釈は改善されることが望ましいと考えられるため、暗号技術評価委員会及び暗号技術活用委員会において注釈変更に関する議論を経た結果、以下の変更案が取りまとめられた。

２．変更案

＜現行及び変更案＞

現行	128-bit RC4 は、SSL（TLS1.0 以上）に限定して利用すること。
変更案	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。

なお、具体的な変更後の CRYPTREC 暗号リストの案は別添のとおりとし、現在の CRYPTREC 暗号リストの注釈について、直接変更することとする。

また、注釈の変更経緯が分かるように、CRYPTREC 暗号リストの最下部に変更履歴情報を付ける。

参考文献

[1] Itsik Mantin, Adi Shamir, “A Practical Attack on Broadcast RC4,” FSE 2001.

[2] 五十部, ストリーム暗号 RC4 の安全性評価, 2012, http://www.cryptrec.go.jp/estimation/techrep_id2205.pdf

[3] AlFardan, Bernstein, Paterson, Poettering, Schuldt, “On the Security of RC4 in TLS,” USENIX Security 2013.

資料３
別　添

CRYPTREC 暗号リストの変更案

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）

平成２５年３月１日
総　　務　　省
経　済　産　業　省

電子政府推奨暗号リスト

暗号技術検討会[1]及び関連委員会（以下、「CRYPTREC」という。）により安全性及び実装性能が確認された暗号技術[2]について、市場における利用実績が十分であるか今後の普及が見込まれると判断され、当該技術の利用を推奨するもののリスト。

技術分類		名称
公開鍵暗号	署名	DSA
		ECDSA
		RSA-PSS(注1)
		RSASSA-PKCS1-v1_5(注1)
	守秘	RSA-OAEP(注1)
	鍵共有	DH
		ECDH
共通鍵暗号	64 ビットブロック暗号(注2)	3-key Triple DES(注3)
	128 ビットブロック暗号	AES
		Camellia
	ストリーム暗号	KCipher-2
ハッシュ関数		SHA-256
		SHA-384
		SHA-512
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		OFB
	認証付き秘匿モード	CCM
		GCM(注4)
メッセージ認証コード		CMAC
		HMAC
エンティティ認証		ISO/IEC 9798-2
		ISO/IEC 9798-3

[1] 総務省政策統括官（情報通信担当）及び経済産業省商務情報政策局長が有識者の参集を求め、暗号技術の普及による情報セキュリティ対策の推進を図る観点から、専門家による意見等を聴取することにより、総務省及び経済産業省における施策の検討に資することを目的として開催。

[2] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合 CRYPTREC 暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

（注1）　「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」（平成20年4月 情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
http://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf
（平成 25 年 3 月 1 日現在）

（注2）　より長いブロック長の暗号が利用できるのであれば、128 ビットブロック暗号を選択することが望ましい。

（注3）　3-key Triple DES は、以下の条件を考慮し、当面の利用を認める。
1） NIST SP 800-67 として規定されていること。
2） デファクトスタンダードとしての位置を保っていること。

（注4）　初期化ベクトル長は 96 ビットを推奨する。

推奨候補暗号リスト

CRYPTRECにより安全性及び実装性能が確認され、今後、電子政府推奨暗号リストに掲載される可能性のある暗号技術[3]のリスト。

技術分類		名称
公開鍵暗号	署名	該当なし
	守秘	該当なし
	鍵共有	PSEC-KEM(注5)
共通鍵暗号	64ビットブロック暗号(注6)	CIPHERUNICORN-E
		Hierocrypt-L1
		MISTY1
	128ビットブロック暗号	CIPHERUNICORN-A
		CLEFIA
		Hierocrypt-3
		SC2000
	ストリーム暗号	Enocoro-128v2
		MUGI
		MULTI-S01(注7)
ハッシュ関数		該当なし
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード	該当なし
メッセージ認証コード		PC-MAC-AES
エンティティ認証		ISO/IEC 9798-4

（注5） KEM（Key Encapsulating Mechanism）－ DEM（Data Encapsulating Mechanism）構成における利用を前提とする。

（注6） より長いブロック長の暗号が利用できるのであれば、128ビットブロック暗号を選択することが望ましい。

（注7） 平文サイズは64ビットの倍数に限る。

[3] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合 CRYPTREC 暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

運用監視暗号リスト

実際に解読されるリスクが高まるなど、推奨すべき状態ではなくなった暗号技術[4]のうち、互換性維持のために継続利用を容認するもののリスト。互換性維持以外の目的での利用は推奨しない。

技術分類		名称
公開鍵暗号	署名	該当なし
	守秘	RSAES-PKCS1-v1_5(注8)(注9)
	鍵共有	該当なし
共通鍵暗号	64 ビットブロック暗号	該当なし
	128 ビットブロック暗号	該当なし
	ストリーム暗号	128-bit RC4(注10)
ハッシュ関数		RIPEMD-160
		SHA-1(注8)
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード	該当なし
メッセージ認証コード		CBC-MAC(注11)
エンティティ認証		該当なし

（注8） 「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」（平成20年4月 情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
http://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf
（平成 25 年 3 月 1 日現在）

（注9） SSL 3.0 / TLS 1.0, 1.1, 1.2 で利用実績があることから当面の利用を認める。

（注10） 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。

（注11） 安全性の観点から、メッセージ長を固定して利用すべきである。

[4] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合 CRYPTREC 暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

変更履歴情報

変更日付	変更箇所	変更前の記述	変更後の記述
平成 27 年*月**日	（注10）	128-bit RC4 は 、SSL（TLS1.0 以上）に限定して利用すること。	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。

資料4

暗号技術検討会
２０１４年度　報告書（案）

２０１５年３月

目　　次

１．はじめに

情報通信技術を安心・安全に利用できる環境を構築していくにあたり、セキュリティの基盤技術として暗号技術の必要性は益々大きくなっている。昨年 11 月 6 日に「サイバーセキュリティ基本法」が成立し、今年 1 月 9 日に「サイバーセキュリティ戦略本部」が設置されるなど、政府もサイバーセキュリティ対策をより一層強く推進している。「政府機関の情報セキュリティ対策のための統一基準」にも、暗号化及び電子署名のアルゴリズムについて、CRYPTREC 暗号リストに記載されたアルゴリズムを使用することが定められており、CRYPTREC としても、暗号に関する技術的な評価等を通じて、政府のこれらの動きを適切に支援していく。

また、同法では「情報の自由な流通を確保しつつ、サイバーセキュリティの確保を図ることが喫緊の課題」であることが謳われているが、情報の自由な流通とサイバーセキュリティを両立させるための基盤技術として、暗号技術に対する社会からの要請は一段と大きく、多様化していくものと考えられる。来たるべき IoT 社会においては、情報の流通量が顕著に増加するだけでなく、接続される機器の種類や性能も千差万別となり、それぞれの使用目的に合わせた暗号技術の利用が飛躍的に広がることが期待される。

CRYPTREC は、「CRYPTREC 暗号リストの大改定」及び「暗号技術評価委員会及び暗号技術活用委員会からなる 2 委員会体制への移行」を行ってから 2 年が経過した。CRYPTREC として、従来通りの暗号技術の評価・監視活動を引き続き堅持するとともに、軽量暗号等の新暗号技術への取組や、一般での利用も想定した形での「SSL/TLS 暗号設定ガイドライン」の策定など、上記のような暗号技術に対する社会の要請に応えるべく、新しい領域の活動も推進してきたが、これらの取組を行っていく中で、改めて重要性を認識した点、新たに課題として浮き彫りにされた点などが確認された。日本の暗号政策の要石を担う存在として、新しい時代の要請にあわせ CRYPTREC が今後どう在るべきか、改めて検討することが有意義であると思われる。

今年度の委員会別の活動として、暗号技術評価委員会では、暗号技術の安全性及び実装に係る監視及び評価、軽量暗号などの新世代暗号に係る調査、暗号技術の安全な利用方法に関する調査等を行った。暗号技術活用委員会では、運用ガイドラインの策定、標準化推進に向けた取組、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等を行った。なお、2014 年度の活動のうち、詳細な技術的事項については、暗号技術評価委員会及び暗号技術活用委員会における議論を踏まえて、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取りまとめる「CRYPTREC Report 2014」を参照いただきたい。

末筆であるが、本検討会及び関係委員会に御協力いただいた構成員、オブザーバの方々をはじめ、関係者の皆様に心から謝意を表する次第である。

2015 年 3 月

暗号技術検討会

座長　今井　秀樹

２．暗号技術検討会開催の背景及び開催状況

２．１．暗号技術検討会開催の背景

暗号技術は情報セキュリティの基盤技術であり、その安全性を暗号技術の専門家により技術的、専門的な見地から客観的に評価することが重要である。電子政府のセキュリティ確保のためには、安全性に優れた暗号技術を利用することが不可欠である。

このため、総務省及び経済産業省は、客観的な評価により安全性及び実装性に優れると判断される暗号技術をリスト化し、各府省に対してその利用を推奨することにより、高度な安全性と信頼性に支えられ、国民が安心して利用できる電子政府の構築に貢献することを目指し、2001 年度から暗号技術検討会を開催した。

暗号技術検討会において 2002 年度に策定された電子政府推奨暗号リストは、2012 年度に 10 年ぶりの改定が行われ、「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）」（以下、「CRYPTREC 暗号リスト」という。）として発表されたが、その後においても、電子政府推奨暗号等の安全性を評価・監視し、暗号技術の更なる普及促進を行うため、総務省及び経済産業省は、継続的に暗号技術検討会を開催している。

２．２． CRYPTREC の体制

CRYPTREC とは、Cryptography Research and Evaluation Committees の略であり、総務省及び経済産業省が共同で開催する暗号技術検討会（座長：今井秀樹東京大学名誉教授）と、独立行政法人情報通信研究機構（NICT）及び独立行政法人情報処理推進機構（IPA）が共同で開催する委員会から構成される暗号技術評価プロジェクトをいう。

2014 年度の CRYPTREC の体制は、前年度に引き続き、暗号技術検討会の下に、暗号技術評価委員会及び暗号技術活用委員会の２つの委員会を設置し、調査・検討を行った。

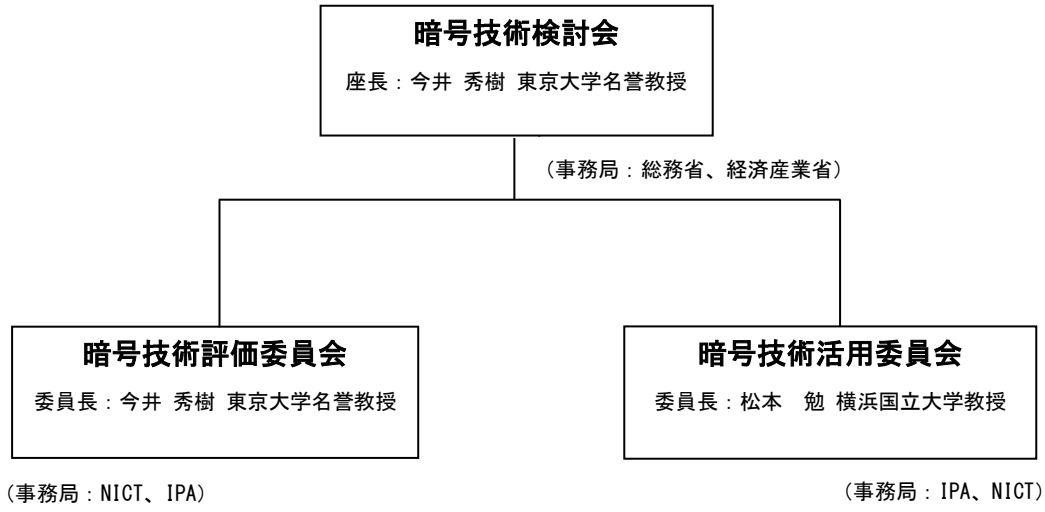


図 2.1　2014 年度 CRYPTREC の体制図

２．３．暗号技術検討会の開催状況

2014 年度の暗号技術検討会は、暗号技術評価委員会及び暗号技術活用委員会の活動報告、CRYPTREC 暗号リストの注釈変更等を審議するために２回開催した。

【第 1 回】2014 年 10 月 9 日（木）14:00～15:05

（主な議題）

- 暗号技術評価委員会及び暗号技術活用委員会の中間報告について

（概要）

- 暗号技術検討会の下部委員会である、暗号技術評価委員会及び暗号技術活用委員会の 2014 年度の活動の中間報告を行った。
- 軽量暗号について、国内でどういったアプリケーションを利用し、どのように利用を推進していくのかという視点での検討が必要との意見があった。
- 「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」及び「暗号政策の中長期的視点からの取組の検討」に関して、国産暗号、暗号アルゴリズムといった枠組みにとらわれず広い意味での暗号技術と産業競争力や課題解決との関係性を整理するべきという意見があった。

【第 2 回】2015 年 3 月 27 日（金）10:00～12:00

（主な議題）

- 暗号技術評価委員会、暗号技術活用委員会の活動報告について
- CRYPTREC 暗号リストの注釈変更について（RC4）
- 2014 年度暗号技術検討会報告書（案）について
- 暗号技術検討会における小グループの設置について
- 2015 年度の暗号技術評価委員会活動計画について
- 2015 年度の暗号技術活用委員会の活動について

（概要）

- 本日の議論を踏まえ記載。

３．各委員会の活動報告

３．１．暗号技術評価委員会

３．１．１．活動の概要

暗号技術評価委員会は、2013 年度に新たに発足した委員会であり、CRYPTREC 暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。主要な検討課題は以下のとおりである。

・暗号技術の安全性及び実装に係る監視及び評価
・新世代暗号に係る調査
・暗号技術の安全な利用方法に関する調査

これらの課題について 2014 年度に行った具体的な検討内容を、以下のとおり報告する。

３．１．２．2014 年度の活動内容

暗号技術の安全性及び実装に係る監視及び評価

2014 年度は、① 学会等での情報収集に基づく CRYPTREC 暗号等の監視、② 128-bit RC4 の注釈に係る検討、③ 推奨候補暗号リストへの追加のための外部評価等を実施した。

①について、研究集会、国際会議、研究論文誌の情報等を収集し、リスト掲載暗号の安全性について監視活動を行った。攻撃研究等に関して早急な対処が必要なものは存在しなかったが、暗号解読技術等の進展が見られ、これらについて引き続き注視していく必要がある。

②について、128-bit RC4 に対する有力な攻撃手法が明らかにされたことから、速やかに他のアルゴリズムに移行されることが望ましいという合意に至り、「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。」という注釈変更案を採択した。

③について、推奨候補暗号リストへの追加を検討する事務局選出の暗号アルゴリズムとして、SHA-2 ファミリーに新たに追加されたハッシュ関数や SHA-3 について安全性に係る外部評価を実施した。

新世代暗号に係る調査

本項目に係る活動に関しては、暗号技術評価委員会の下に暗号技術調査 WG（暗号解析評価）及び暗号技術調査 WG（軽量暗号）を設置し、議論した。暗号技術調査 WG（暗号解析評価）では、格子問題や離散対数問題の困難性等、暗号技術の安全性を支える数学的問題の困難性に係る調査を実施した。暗号技術調査 WG（軽量暗号）では、昨年度から実施してきた現状調査、実装評価等に基づき、来年度以降に暗号技術ガイドラインを作成する等の CRYPTREC としての活動方針について暗号技術評価委員会に対して提言を行った。

暗号技術の安全な利用方法に関する調査

昨年度発行した 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」について、2014 年 10 月に SSL3.0 における CBC モードに対する新たな攻撃として「POODLE 攻撃」が公表されたことを受け、当該攻撃に関する記載を追加した。

３．１．３．暗号技術評価委員会の開催状況

2014 年度、暗号技術評価委員会は計 3 回開催した。各回会合の概要は表 3.1 のとおりである。

表 3.1　暗号技術評価委員会の開催

回	年月日	議題
第 1 回	2014 年　8 月 4 日	暗号技術評価委員会活動方針の検討 WG 活動方針の検討 外部評価についての検討 128-bit RC4 の注釈変更に関する検討 監視状況報告 CRYPTREC 暗号リスト掲載暗号技術の仕様書に関する検討
第 2 回	2014 年　12 月 25 日	WG 中間活動報告 外部評価についての中間報告 監視状況報告 暗号技術ガイドラインの更新に関する検討 CRYPTREC 暗号リスト掲載暗号技術の仕様書に関する検討
第 3 回	2015 年　3 月 2 日	WG 今年度活動報告 外部評価についての報告 監視状況報告 128-bit RC4 の注釈変更に関する検討 暗号技術ガイドラインの更新に関する検討 CRYPTREC 暗号リスト掲載暗号技術の仕様書に関する検討 CRYPTREC2014 の目次案に関する検討 次年度の活動計画に関する検討

３．２．暗号技術活用委員会

３．２．１．活動の概要

暗号技術活用委員会は、2013 年度から新たに設置された委員会であり、CRYPTREC 暗号リスト改定の一環である暗号技術の利用状況に係る調査、暗号技術における国際競争力の向上及び運用面での安全性向上に関する検討を実施する。主要な検討課題は以下のとおりである。

・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（運用ガイドラインの整備、教育啓発資料の作成等）
・暗号技術の利用状況に係る調査及び必要な対策の検討等
・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）

2014 年度は、2013 年度から 2 年間かけて取り組んだ、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、暗号政策の中長期的視点からの取組の検討について、取りまとめを行った。以下に、その具体的内容を報告する。

３．２．２．2014 年度の活動内容

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

暗号技術の普及促進・セキュリティ産業の競争力強化については、2013 年度に実施した電子政府推奨暗号リストの活用状況や国産暗号に対する考え方に関するヒアリング内容を踏まえ、課題分析を行い、今後の検討にあたっての留意点を取りまとめた。

また、暗号の普及促進の具体的な方策について検討するため、暗号技術活用委員会の下に運用ガイドライン WG 及び標準化推進 WG を設置した。

運用ガイドライン WG では、暗号システムを安全に利用できるようにすることを目的として、利用者が多い SSL/TLS について、「SSL/TLS 暗号設定ガイドライン」の策定を行い、利用者が必要な設定を確認しやすいよう、「SSL/TLS 暗号設定ガイドラインチェックリスト」についても策定した。

標準化推進 WG では、今後暗号技術を提案する者が提案先を選定する際に参考となるような資料として、規格の参照関係を整理した、「暗号技術参照関係の俯瞰図」を作成した。また、様々な標準化機関に対する日本からの暗号アルゴリズム提案を支援するため、標準化団体における基本的な情報、提案活動に関する交渉ノウハウや課題等を取りまとめた。

暗号政策の中長期的視点からの取組の検討

暗号政策の中長期的視点からの取組である暗号人材育成については、2013 年度に実施した必要な人材像に関するヒアリング内容を踏まえ、今後実施していくべき人材育成策について検討するにあたっての留意点を取りまとめた。

RC4 の注釈について

CRYPTREC 暗号リストにおける RC4 の注釈について検討を行い、早期に RC4 からの移行を進めることが望ましく、今後は極力利用すべきでないという意図を明確化する観点から、暗号技術活用委員会として、「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。」という注釈変更案を採択した。

３．２．３．暗号技術活用委員会の開催状況

2014 年度、暗号技術活用委員会は、計 3 回開催された。各回会合の概要は表 3.2 のとおりである。

表 3.2　暗号技術活用委員会の開催

回	年月日	議題
第 1 回	2014 年　10 月 30 日	本年度の活動計画の確認 運用ガイドライン WG 及び標準化推進 WG の活動について 最終報告書とりまとめに向けた論点整理 128-bit RC4 の注釈変更について
第 2 回	2015 年　1 月 26 日	SSL/TLS 暗号設定ガイドラインの中間審議 標準化推進 WG の活動についての中間審議 最終報告書の内容に関する中間審議 128-bit RC4 の注釈変更について
第 3 回	2015 年　3 月 10 日	運用ガイドライン WG 及び標準化推進 WG の活動報告 最終報告書の審議

４．今後の CRYPTREC の活動について

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、暗号技術の安全性及び実装に係る監視及び評価を引き続き実施するとともに、現在の２委員会体制に移行してから２年間が経過したことを踏まえ、暗号技術検討会に２年間の活動の評価と今後の CRYPTREC のあり方について議論を行う小グループを設置し、日本の安全な ICT 基盤確立にむけて CRYPTREC が取り組むべき活動の範囲や方針について提言を行う。

資料5

暗号技術検討会における小グループの設置について（案）

平成 27 年 3 月 27 日
暗号技術検討会事務局

１．設置目的

2001 年に CRYPTREC が発足した当初の目的は、安全でない暗号アルゴリズムが乱立する中で、電子政府において利用が推奨される安全な暗号アルゴリズムを確定させることであり､活動成果として 2003 年に｢電子政府推奨暗号リスト」を策定し、2013 年にその改定版である「CRYPTREC 暗号リスト」を策定した。

また、2013 年のリスト改定を契機として、従来からの「CRYPTREC 暗号リストの安全性維持に係る取組」に加え、新たな試みとして「新しい暗号技術の調査」、「暗号技術の普及促進に係る取組」､「中長期的視点に立った暗号政策に係る検討」等を行った。

上記活動を通じて明らかにされた、暗号技術を取り巻く環境、サイバーセキュリティ基本法の施行といった社会情勢の変化等を鑑みると、CRYPTREC が果たすべき役割は、CRYPTREC 暗号リストの策定及び維持に限られるものではなく、より柔軟に活動することが望ましいと考えられる。

よって、暗号技術検討会に小グループを設置し、今後の CRYPTREC の活動内容及び対象範囲について集中的に検討を行うこととする。

２．検討項目

- 我が国の暗号技術を取り巻く環境の変化に応じた CRYPTREC の活動内容及び対象範囲について
- 暗号技術の安全性確保等に係る活動のあり方
- 暗号技術の普及促進、産業振興及び人材育成に係る活動のあり方

等

３．参加メンバー

暗号技術検討会の構成員、総務省、経済産業省、NICT、IPA を想定。必要に応じて関係省庁のオブザーバの参加を認める。

４．検討スケジュール（予定）

2015 年度上半期中の早期に検討を行い、検討結果を暗号技術検討会に報告する。

資料 6

2015 年度暗号技術評価委員会活動計画(案)

1. 活動目的

CRYPTREC 暗号リストに掲載されている暗号技術や電子政府システム等で利用される暗号技術の安全性維持及び信頼性確保のために安全性及び実装に係る監視及び評価を行う。

2. 活動概要

(1) 暗号技術の安全性及び実装に係る監視及び評価

下記の通り、暗号技術の安全性に係る監視・評価 及び 実装に係る技術の監視・評価を実施する。

① CRYPTREC 暗号等の監視

国際会議等で発表される CRYPTREC 暗号リストの安全性及び実装に係る技術（暗号モジュールに対する攻撃とその対策も含む）に関する監視を行う。報告は、なるべく直近の暗号技術評価委員会で報告することを目標とする。

② 電子政府推奨暗号リスト及び推奨候補暗号リストからの運用監視暗号リストへの降格及び 運用監視暗号リストからの危殆化が進んだ暗号の削除

CRYPTREC 暗号リストの安全性に係る監視活動を継続的に行い、急速に危殆化が進んだ暗号技術やその予兆のある暗号技術の安全性について評価を行う。また、リストからの降格や削除、注釈の改訂が必要か検討を行う。

③ CRYPTREC 注意喚起レポートの発行

CRYPTREC 暗号リストの安全性及び実装に係る技術の監視活動を通じて、国際会議等で発表された攻撃等の概要や想定される影響範囲、対処方法について早期に公開することが望ましいと判断された場合、注意喚起レポートを発行する。

④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加

標準化動向に鑑み電子政府システム等での利用が見込まれると判断される暗号技術の追加を検討する。

⑤ 新技術に関する調査及び評価

(将来的に)有用になると考えられる技術について、暗号技術調査ワーキンググループにて調査および評価を行う。また、外部評価等を通して新技術やリストに関わる技術の安全性・性能評価を行う。

(2) 新技術に係る調査

- 暗号技術調査ワーキンググループ(暗号解析評価) は、実施すべき課題を検討し、必要に応じて開催する。 具体的な活動については、2015 年度第 1 回 暗号技術評価委員会にて検討する。
- 暗号技術調査ワーキンググループ(軽量暗号)は、詳細評価対象となる技術分類の選定、詳細評価内容の策定を行い、具体的な詳細評価を実施する。詳細評価 の対象技術分類としては、軽量認証暗号の評価を優先的に行う予定である。

(3) 暗号技術の安全な利用方法に関する調査 (技術ガイドラインの整備、学術的な安全性の調査・公表等)

- 「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))」の更新を行う。
- 暗号技術を利用する際の技術面での注意点に関する調査、新技術の安全性・性能に関する調査・評価を行う。
- 具体的な内容については、2015 年度第 1 回暗号技術評価委員会にて検討する。

以上

資料７

2015 年度暗号技術活用委員会の活動について（案）

2015 年度以降の暗号技術活用委員会の担当内容及び体制については暗号技術検討会に設置される小グループで検討を行う予定であり、実施項目の詳細及び実施体制は議論結果に準ずるが、並行して行う準備作業として想定される項目は以下の通りである。

１．現時点で想定される項目

(1) 運用ガイドラインの検討準備

- 一般（民間事業者）向けにも利用できる「ガイドライン」の作成に関わる事前準備を行う

  テーマ案：IPsec、SSH 等

(2) 暗号アルゴリズム利用実績調査の準備

- CRYPTREC 暗号リストの小改定に関して必要な調査に係る準備作業を行う

(3) 2014 年度の成果の普及啓発やフィードバック

- SSL/TLS 暗号設定ガイドライン等の広報活動を行い、使い勝手等の利用者からの声を収集する

２．今後の進め方

上記作業を進めるに当たっては、小グループにおける議論の結果を踏まえ、柔軟な対応を行えるよう作業を実施していくこととする。

以上

参考資料１

２０１４年度　第１回暗号技術検討会　議事概要（案）

１．日時　平成２６年１０月９日（木）　１４：００～１５：０５

２．場所　経済産業省別館１階　１０４各省庁共用会議室

３．出席者（敬称略）

構成員：今井秀樹（座長）、今井正道、上原哲太郎、太田和夫、岡本栄司、岡本龍明、国分明男、近澤武、中山靖司、本間尚文、松井充、松尾真一郎、松本勉、松本泰、向山友也、渡辺創

オブザーバ：奥山剛、根本農史（佐藤正明 代理）、村田莉衣奈（稲垣浩 代理）、江森久子（野口宣大 代理）、大村周一郎、村田秀樹（武田一彦 代理）、田中正幸、濱田和之（鯨井佳則 代理）、岩永敏明（和泉章 代理）、木村和仙、平和昌、寶木和夫、伊藤毅志、亀田繁、西村敏信

暗号技術評価委員会事務局：盛合志帆（独立行政法人情報通信研究機構（NICT））

暗号技術活用委員会事務局：神田雅透（独立行政法人情報処理推進機構（IPA））

暗号技術検討会事務局：

総務省　南俊行、赤阪晋介、筒井邦弘

経済産業省　大橋秀行、上村昌博、中野辰実

４．配布資料

（資料番号）

資料１－１	2014 年度 「暗号技術検討会」開催要綱（案）
資料１－２	暗号技術検討会の公開について（案）
資料２	2014 年度 暗号技術検討会活動計画
資料３	2014 年度 暗号技術評価委員会活動中間報告
資料３別添	監視状況報告
資料４	暗号技術活用委員会の活動状況
参考資料１	2013 年度 第 2 回暗号技術検討会議事概要
参考資料２	2014 年度 暗号技術評価委員会活動計画
参考資料３	2014 年度 暗号技術活用委員会活動計画
参考資料４	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
参考資料５	2014 年度　暗号技術検討会　構成員・オブザーバ名簿

５．議事概要

１　開会

暗号技術検討会事務局から開会の宣言があり、総務省の南政策統括官から開会の挨拶が行われた。

参考資料５に基づき、暗号技術検討会事務局より構成員の交代（(一般社団法人情報通信ネットワーク産業協会）武市構成員→今井構成員)、オブザーバの交代（(財務省）郷氏→武田氏、(厚生労働省）三富氏→鯨井氏、(経済産業省）辻本氏→和泉氏）及び構成員の欠席（佐々木構成員）について説明が行われた。

２　議事

（１）2014 年度　暗号技術検討会　開催要綱等について

資料１－１及び１－２に基づき、「2014 年度暗号技術検討会開催要綱案」及び「暗号技術検討会の公開」について事務局より説明が行われた。質疑はなし。原案のとおり承認された。構成員の互選により、座長として今井秀樹構成員を選任した。今井座長より、座長代理として松本勉構成員が指名された。

（２）2014 年度　暗号技術検討会活動計画

資料２に基づき、事務局より説明が行われた。質疑はなし。前回検討会において承認された本活動計画について、改めて確認した。

（３）暗号技術評価委員会の活動に関する中間報告

資料３及び資料３別添に基づき、暗号技術評価委員会事務局より説明が行われた。

〇質疑応答

今井座長：RC4 の注釈変更は、なるべく RC4 を使ってもらいたくないというメッセージを意図しているという理解でよいか。

暗号技術評価委員会事務局：そのとおり。

上原構成員：「（１）④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加」に関連して、現在 RC4 が使用できなくなったことによってストリーム暗号の有力な選択肢が存在していないところ、国産暗号アルゴリズムの国際標準化の推進に係る取組も実施するのか。

暗号技術評価委員会事務局：事務局としての取組に加え、暗号技術活用委員会のもとに標準化推進ワーキンググループが設置されており、これらの活動とも連携し推進していく。

寶木オブザーバ：軽量暗号に関する技術公募は実施しないということだが、ISO で標準化が進んでいることと関係があるのか。

暗号技術評価委員会事務局：当面の目標として CRYPTREC が技術公募まで行う段階にないと判断したもの。ISO で標準化が進んでいることと直接の関係はない。

松尾構成員：軽量暗号については、既に ISO で標準化されて何年かたっているが、利用が進んでいない。CRYPTREC として、国内でどういったアプリケーションで使用していくのか、どのように利用を促進していくのか、といった点についてどのような議論を行っているのか。

暗号技術評価委員会事務局：暗号技術調査ワーキンググループ（軽量暗号）の今年度報告書に、アプリケーションについても記載していく。

松尾構成員：具体的にどのようなアプリケーションが挙げられているのか。

暗号技術評価委員会事務局：構成員からの意見としては、リアルタイム性が求められる用途として「メモリの暗号化」、「自動車の安全に関わる部分」などがあった。

（４）暗号技術活用委員会の活動状況

資料４に基づき、暗号技術活用委員会事務局より説明が行われた。

〇質疑応答

松本（泰）構成員：「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」及び「② 暗号政策の中長期的視点からの取組の検討」に関して、「暗号」はサービス提供者側の競争力強化に必要なコア技術であると認識している。国産暗号、暗号アルゴリズムといった枠組にとらわれず広い意味での暗号技術と、産業競争力や課題解決との関係性について整理された方が良いと考えるが、この点についてどう考えるか。また、電子政府向けという従来の CRYPTREC の範囲を越えて、産業の領域まで広げていくことについて、どのように考えているのか。

松本（勉）暗号技術活用委員会委員長：暗号アルゴリズム以外の、暗号プロトコル等の広い意味での暗号技術についても、本来 CRYPTREC で取り扱われるべき。しかし、どのように取り扱うかという点について今年度の暗号技術活用委員会で検討しているところであるが、数年間模索を続けている大変難しい課題であると認識している。

今井座長：今年度の暗号技術活用委員会において、課題をある程度整理した形で出してもらえるのか。

暗号技術活用委員会事務局：アプリケーションとの連携という観点からの整理は難しいと感じている。諸外国の動向という点については、まず制度面・体制面がどうなっているのか、調査を行った段階である。

今井座長：広く暗号技術について検討することは重要である。上手く進めていけば産業競争力の強化にも繋がる可能性がある。今後の検討課題である。

今井座長：近年、国際標準化を行う場も増えており、標準化推進ワーキンググループの委員だけで全てをカバーすることは難しいのではないか。

暗号技術活用委員会事務局：標準化推進ワーキンググループに担当の委員がいない分野についてはカバー出来ていないが、まずは比較的重要と思われる分野について俯瞰図を作成し、将来的に拡張していくことも検討する。

（５）その他

〇質疑応答

松本（勉）構成員：現在法案審議が進んでいるサイバーセキュリティ基本法について、法案が成立した場合に、政府における暗号技術の取扱はどのように変化すると考えられるのか。

暗号技術検討会事務局：まず本法案は議員立法である。政府として立法趣旨を推察した場合、サイバー攻撃が益々増加する中、政府として積極的に対策を講じることを示すものである。暗号技術は、セキュリティ対策を講じる際の手段として重要であると考える。

今井座長：近年、量子コンピュータについて話題に上ることが多い。最近話題になった量子アニーリングに基づく計算法について、ご説明願えないか。

松井構成員：現時点で暗号技術の安全性を脅かすような手法ではない。素因数分解を容易に行えるような量子コンピューティングは、技術的に大変難しい。

国分構成員：暗号技術を輸出する場合、諸外国の規制などにも影響を受けると考えられるので、CRYPTREC としてもグローバルな視点をより強く意識した方が良いのではないか。

暗号技術活用委員会事務局：諸外国では、安全保障の観点を含めて暗号を取り扱っており、輸出できたとしてもその国に入り込んでいけるかどうかは別の問題といった状況である。

３　閉会

経済産業省の大橋審議官から閉会の挨拶が行われた。

暗号技術検討会事務局から、2014 年度第２回暗号技術検討会は３月頃の開催を予定しており、詳細な日程、場所等については、別途連絡する旨の説明が行われた。

以上

参考資料2

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）

平成２５年３月１日
総　　務　　省
経　済　産　業　省

電子政府推奨暗号リスト

暗号技術検討会[1]及び関連委員会（以下、「CRYPTREC」という。）により安全性及び実装性能が確認された暗号技術[2]について、市場における利用実績が十分であるか今後の普及が見込まれると判断され、当該技術の利用を推奨するもののリスト。

技術分類		名称
公開鍵暗号	署名	DSA
		ECDSA
		RSA-PSS(注1)
		RSASSA-PKCS1-v1_5(注1)
	守秘	RSA-OAEP(注1)
	鍵共有	DH
		ECDH
共通鍵暗号	64ビットブロック暗号(注2)	3-key Triple DES(注3)
	128ビットブロック暗号	AES
		Camellia
	ストリーム暗号	KCipher-2
ハッシュ関数		SHA-256
		SHA-384
		SHA-512
暗号利用モード	秘匿モード	CBC
		CFB
		CTR
		OFB
	認証付き秘匿モード	CCM
		GCM(注4)
メッセージ認証コード		CMAC
		HMAC
エンティティ認証		ISO/IEC 9798-2
		ISO/IEC 9798-3

[1] 総務省政策統括官（情報通信担当）及び経済産業省商務情報政策局長が有識者の参集を求め、暗号技術の普及による情報セキュリティ対策の推進を図る観点から、専門家による意見等を聴取することにより、総務省及び経済産業省における施策の検討に資することを目的として開催。

[2] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合 CRYPTREC 暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

（注1）　「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」（平成20年4月 情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
http://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf
（平成 25 年 3 月 1 日現在）

（注2）　より長いブロック長の暗号が利用できるのであれば、128 ビットブロック暗号を選択することが望ましい。

（注3）　3-key Triple DES は、以下の条件を考慮し、当面の利用を認める。
1） NIST SP 800-67 として規定されていること。
2） デファクトスタンダードとしての位置を保っていること。

（注4）　初期化ベクトル長は 96 ビットを推奨する。

推奨候補暗号リスト

CRYPTRECにより安全性及び実装性能が確認され、今後、電子政府推奨暗号リストに掲載される可能性のある暗号技術[3]のリスト。

技術分類		名称
公開鍵暗号	署名	該当なし
	守秘	該当なし
	鍵共有	PSEC-KEM(注5)
共通鍵暗号	64 ビットブロック暗号(注6)	CIPHERUNICORN-E
		Hierocrypt-L1
		MISTY1
	128 ビットブロック暗号	CIPHERUNICORN-A
		CLEFIA
		Hierocrypt-3
		SC2000
	ストリーム暗号	Enocoro-128v2
		MUGI
		MULTI-S01(注7)
ハッシュ関数		該当なし
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード	該当なし
メッセージ認証コード		PC-MAC-AES
エンティティ認証		ISO/IEC 9798-4

（注5） KEM（Key Encapsulating Mechanism）- DEM（Data Encapsulating Mechanism）構成における利用を前提とする。

（注6） より長いブロック長の暗号が利用できるのであれば、128 ビットブロック暗号を選択することが望ましい。

（注7） 平文サイズは 64 ビットの倍数に限る。

[3] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合 CRYPTREC 暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

運用監視暗号リスト

実際に解読されるリスクが高まるなど、推奨すべき状態ではなくなった暗号技術[4]のうち、互換性維持のために継続利用を容認するもののリスト。互換性維持以外の目的での利用は推奨しない。

技術分類		名称
公開鍵暗号	署名	該当なし
	守秘	RSAES-PKCS1-v1_5(注8)(注9)
	鍵共有	該当なし
共通鍵暗号	64 ビットブロック暗号	該当なし
	128 ビットブロック暗号	該当なし
	ストリーム暗号	128-bit RC4(注10)
ハッシュ関数		RIPEMD-160
		SHA-1(注8)
暗号利用モード	秘匿モード	該当なし
	認証付き秘匿モード	該当なし
メッセージ認証コード		CBC-MAC(注11)
エンティティ認証		該当なし

（注8） 「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」（平成20年4月 情報セキュリティ政策会議決定、平成24年10月情報セキュリティ対策推進会議改定）を踏まえて利用すること。
http://www.nisc.go.jp/active/general/pdf/angou_ikoushishin.pdf
（平成 25 年 3 月 1 日現在）

（注9） SSL 3.0 / TLS 1.0, 1.1, 1.2 で利用実績があることから当面の利用を認める。

（注10） 128-bit RC4 は、SSL （TLS 1.0 以上）に限定して利用すること。

（注11） 安全性の観点から、メッセージ長を固定して利用すべきである。

[4] 暗号利用モード、メッセージ認証コード、エンティティ認証は、他の技術分類の暗号技術と組み合わせて利用することとされているが、その場合 CRYPTREC 暗号リストに掲載されたいずれかの暗号技術と組み合わせること。

参考資料３

2014 年度　暗号技術検討会　構成員・オブザーバ名簿

2015.3.27 現在

（構成員）

今井　秀樹	東京大学　名誉教授
今井　正道	一般社団法人情報通信ネットワーク産業協会　常務理事
上原　哲太郎	立命館大学　情報理工学部情報システム学科　教授
太田　和夫	電気通信大学大学院　情報理工学研究科 総合情報学専攻（セキュリティ情報学コース）　教授
岡本　栄司	筑波大学大学院　システム情報工学研究科　教授
岡本　龍明	日本電信電話株式会社　セキュアプラットフォーム研究所 岡本特別研究室　室長（社団法人電気通信事業者協会代表兼務）
金子　敏信	東京理科大学　理工学部電気電子情報工学科　教授
国分　明男	一般財団法人ニューメディア開発協会　顧問・首席研究員
佐々木　良一	東京電機大学　未来科学部情報メディア学科　教授
近澤　　武	独立行政法人情報処理推進機構　セキュリティセンター暗号グループ グループリーダー（ISO/IEC JTC 1/SC27/WG2 Convenor（国際主査））
中山　靖司	日本銀行　金融研究所情報技術研究センター　企画役
本間　尚文	東北大学大学院　情報科学研究科　情報基礎科学専攻　准教授
松井　　充	三菱電機株式会社　情報技術総合研究所　技師長
松尾　真一郎	独立行政法人情報通信研究機構　社会還元促進部門　統括 （ISO/IEC JTC1 SC27/WG2 （国内小委員会主査））
松本　　勉	横浜国立大学　大学院環境情報研究院　教授
松本　　泰	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　マネージャー
向山　友也	社団法人テレコムサービス協会　技術・サービス委員会　委員長
渡辺　　創	ISO/IEC JTC1 SC27　国内委員会　委員長

（オブザーバ）

奥山　　剛	内閣官房情報セキュリティセンター内閣企画官
村田　利見	警察庁情報通信局情報管理課長
稲垣　　浩	総務省行政管理局行政情報システム企画課情報システム企画官
増田　直樹	総務省自治行政局地域政策課地域情報政策室長
篠原　俊博	総務省自治行政局住民制度課長
野口　宣大	法務省民事局商事課長
大村　周一郎	外務省大臣官房情報通信課長
武田　一彦	財務省大臣官房文書課業務企画室長
溝口　浩和	文部科学省大臣官房政策課情報システム企画室長
鯨井　佳則	厚生労働省政策統括官付情報政策担当参事官
和泉　　章	経済産業省産業技術環境局基準認証ユニット国際電気標準課長
木村　和仙	防衛省運用企画局情報通信・研究課サイバー攻撃対処・情報保証企画室長
平　　和昌	独立行政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長
寶木　和夫	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 副研究部門長
伊藤　毅志	独立行政法人情報処理推進機構セキュリティセンター長
竹内　英二	一般財団法人日本情報経済社会推進協会電子署名・認証センター長
西村　敏信	公益財団法人金融情報システムセンター監査安全部長

（五十音順、敬称略）

参考資料１

２０１４年度　第１回暗号技術検討会　議事概要（案）

１．日時　平成２６年１０月９日（木）　１４：００～１５：０５

２．場所　経済産業省別館１階　１０４各省庁共用会議室

３．出席者（敬称略）

構成員：今井秀樹（座長）、今井正道、上原哲太郎、太田和夫、岡本栄司、岡本龍明、国分明男、近澤武、中山靖司、本間尚文、松井充、松尾真一郎、松本勉、松本泰、向山友也、渡辺創

オブザーバ：奥山剛、根本農史（佐藤正明 代理）、村田莉衣奈（稲垣浩 代理）、江森久子（野口宣大 代理）、大村周一郎、村田秀樹（武田一彦 代理）、田中正幸、濱田和之（鯨井佳則 代理）、岩永敏明（和泉章 代理）、木村和仙、平和昌、寶木和夫、伊藤毅志、亀田繁、西村敏信

暗号技術評価委員会事務局：盛合志帆（独立行政法人情報通信研究機構（NICT））

暗号技術活用委員会事務局：神田雅透（独立行政法人情報処理推進機構（IPA））

暗号技術検討会事務局：

総務省　南俊行、赤阪晋介、筒井邦弘

経済産業省　大橋秀行、上村昌博、中野辰実

４．配布資料

（資料番号）

資料１－１	2014 年度 「暗号技術検討会」開催要綱（案）
資料１－２	暗号技術検討会の公開について（案）
資料２	2014 年度 暗号技術検討会活動計画
資料３	2014 年度 暗号技術評価委員会活動中間報告
資料３別添	監視状況報告
資料４	暗号技術活用委員会の活動状況
参考資料１	2013 年度 第 2 回暗号技術検討会議事概要
参考資料２	2014 年度 暗号技術評価委員会活動計画
参考資料３	2014 年度 暗号技術活用委員会活動計画
参考資料４	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
参考資料５	2014 年度　暗号技術検討会　構成員・オブザーバ名簿

５．議事概要

１　開会

暗号技術検討会事務局から開会の宣言があり、総務省の南政策統括官から開会の挨拶が行われた。

参考資料５に基づき、暗号技術検討会事務局より構成員の交代（(一般社団法人情報通信ネットワーク産業協会）武市構成員→今井構成員)、オブザーバの交代（(財務省）郷氏→武田氏、(厚生労働省）三富氏→鯨井氏、(経済産業省）辻本氏→和泉氏）及び構成員の欠席（佐々木構成員）について説明が行われた。

２　議事

（１）2014 年度　暗号技術検討会　開催要綱等について

資料１－１及び１－２に基づき、「2014 年度暗号技術検討会開催要綱案」及び「暗号技術検討会の公開」について事務局より説明が行われた。質疑はなし。原案のとおり承認された。構成員の互選により、座長として今井秀樹構成員を選任した。今井座長より、座長代理として松本勉構成員が指名された。

（２）2014 年度　暗号技術検討会活動計画

資料２に基づき、事務局より説明が行われた。質疑はなし。前回検討会において承認された本活動計画について、改めて確認した。

（３）暗号技術評価委員会の活動に関する中間報告

資料３及び資料３別添に基づき、暗号技術評価委員会事務局より説明が行われた。

〇質疑応答

今井座長：RC4 の注釈変更は、なるべく RC4 を使ってもらいたくないというメッセージを意図しているという理解でよいか。

暗号技術評価委員会事務局：そのとおり。

上原構成員：「（１）④ 推奨候補暗号リストへの新規暗号（事務局選出）の追加」に関連して、現在 RC4 が使用できなくなったことによってストリーム暗号の有力な選択肢が存在していないところ、国産暗号アルゴリズムの国際標準化の推進に係る取組も実施するのか。

暗号技術評価委員会事務局：事務局としての取組に加え、暗号技術活用委員会のもとに標準化推進ワーキンググループが設置されており、これらの活動とも連携し推進していく。

寶木オブザーバ：軽量暗号に関する技術公募は実施しないということだが、ISO で標準化が進んでいることと関係があるのか。

暗号技術評価委員会事務局：当面の目標として CRYPTREC が技術公募まで行う段階にないと判断したもの。ISO で標準化が進んでいることと直接の関係はない。

松尾構成員：軽量暗号については、既に ISO で標準化されて何年かたっているが、利用が進んでいない。CRYPTREC として、国内でどういったアプリケーションで使用していくのか、どのように利用を促進していくのか、といった点についてどのような議論を行っているのか。

暗号技術評価委員会事務局：暗号技術調査ワーキンググループ（軽量暗号）の今年度報告書に、アプリケーションについても記載していく。

松尾構成員：具体的にどのようなアプリケーションが挙げられているのか。

暗号技術評価委員会事務局：委員からの意見としては、リアルタイム性が求められる用途として「メモリの暗号化」、「自動車の安全に関わる部分」などがあった。

（４）暗号技術活用委員会の活動状況

資料４に基づき、暗号技術活用委員会事務局より説明が行われた。

〇質疑応答

松本（泰）構成員：「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」及び「② 暗号政策の中長期的視点からの取組の検討」に関して、「暗号」はサービス提供者側の競争力強化に必要なコア技術であると認識している。国産暗号、暗号アルゴリズムといった枠組にとらわれず広い意味での暗号技術と、産業競争力や課題解決との関係性について整理された方が良いと考えるが、この点についてどう考えるか。また、電子政府向けという従来の CRYPTREC の範囲を越えて、産業の領域まで広げていくことについて、どのように考えているのか。

松本（勉）暗号技術活用委員会委員長：暗号アルゴリズム以外の、暗号プロトコル等の広い意味での暗号技術についても、本来 CRYPTREC で取り扱われるべき。しかし、どのように取り扱うかという点について今年度の暗号技術活用委員会で検討しているところであるが、数年間模索を続けている大変難しい課題であると認識している。

今井座長：今年度の暗号技術活用委員会において、課題をある程度整理した形で出してもらえるのか。

暗号技術活用委員会事務局：アプリケーションとの連携という観点からの整理は難しいと感じている。諸外国の動向という点については、まず制度面・体制面がどうなっているのか、調査を行った段階である。

今井座長：広く暗号技術について検討することは重要である。上手く進めていけば産業競争力の強化にも繋がる可能性がある。今後の検討課題である。

今井座長：近年、国際標準化を行う場も増えており、標準化推進ワーキンググループの委員だけで全てをカバーすることは難しいのではないか。

暗号技術活用委員会事務局：標準化推進ワーキンググループに担当の委員がいない分野についてはカバー出来ていないが、まずは比較的重要と思われる分野について俯瞰図を作成し、将来的に拡張していくことも検討する。

（５）その他

〇質疑応答

松本（勉）構成員：現在法案審議が進んでいるサイバーセキュリティ基本法について、法案が成立した場合に、政府における暗号技術の取扱はどのように変化すると考えられるのか。

暗号技術検討会事務局：まず本法案は議員立法である。政府として立法趣旨を推察した場合、サイバー攻撃が益々増加する中、政府として積極的に対策を講じることを示すものである。暗号技術は、セキュリティ対策を講じる際の手段として重要であると考える。

今井座長：近年、量子コンピュータについて話題に上ることが多い。最近話題になった量子アニーリングに基づく計算法について、ご説明願えないか。

松井構成員：現時点で暗号技術の安全性を脅かすような手法ではない。素因数分解を容易に行えるような量子コンピューティングは、技術的に大変難しい。

国分構成員：暗号技術を輸出する場合、諸外国の規制などにも影響を受けると考えられるので、CRYPTREC としてもグローバルな視点をより強く意識した方が良いのではないか。

暗号技術活用委員会事務局：諸外国では、安全保障の観点を含めて暗号を取り扱っており、輸出できたとしてもその国に入り込んでいけるかどうかは別の問題といった状況である。

３　閉会

経済産業省の大橋審議官から閉会の挨拶が行われた。

暗号技術検討会事務局から、2014 年度第２回暗号技術検討会は３月頃の開催を予定しており、詳細な日程、場所等については、別途連絡する旨の説明が行われた。

以上