暗号技術評価報告書 (2002 年度版)
CRYPTREC Report 2002

平成 15 年 3 月
情報処理振興事業協会
通信・放送機構

目次

はじめに

本報告書は、電子政府で利用可能な暗号技術の評価を目的として設立された暗号技術評価委員会の 2002 年度の活動報告である。同委員会では、2003 年度までに基盤構築が予定されている我が国の電子政府 (e-Government) で利用可能な暗号技術のリストアップを目的とした暗号技術評価プロジェクト (CRYPTREC プロジェクト) を推進してきた。本年度はその節目とも言える年度であり、2000 年度以来の 3 年間の集大成を本プロジェクトの成果として報告する。

各種申請届出手続きや政府調達など行政手続きの電子化を実現する電子政府を構築するためには、サービスをより安心して利用できるようにする暗号技術の利用が不可欠である。現在、様々な暗号技術が開発され、それを組み込んだ多くの製品・ソフトウェアが市場に提供されているが、それらの安全性は保証されている訳ではない。このため、暗号技術を電子政府で利用していくためには、暗号技術の適正な評価が行われ、その情報が容易に入手できることが極めて重要となる。

今回の暗号技術評価委員会の活動に先立ち、1999 年度 (平成 11 年度) には情報処理振興事業協会 (IPA) の「政府調達情報セキュリティ標準 (基準) に関する調査研究」と郵政省 (現総務省) 「暗号通信の普及・高度化に関する研究会 (委員長:辻井重男中央大学教授)」との両者で、「暗号技術の評価」を実施すべきであるとの提言がなされた。この両者の報告における共通点は、情報セキュリティの基盤技術である暗号技術について、その信頼性等を技術的・専門的見地から客観的に検証する必要性が強調されていたことである。

この報告を踏まえ、2000 年 5 月に IPA の「政府調達情報セキュリティ標準 (基準) に関する調査研究」のコンサルティング委員会を発展的に解消し、通商産業省 (現経済産業省) の委託事業として情報処理振興事業協会を事務局とする暗号技術評価委員会が設立された。この暗号技術評価委員会は、高度な専門的知識を有する学識経験者により構成され、関係省庁のオブザーバ参加のもとに、暗号技術の安全性等を評価することになった。

さらに 2001 年度は、暗号技術評価をより政府横断的な事業とすべく、暗号技術評価委員会の事務局を通信・放送機構 (TAO) と IPA が担う一方、政策面より暗号技術の検討を行う暗号技術検討会が総務省・経済産業省を事務局として新設され、両者の連携の下、暗号技術評価を実施した。

2002 年度は、2003 年度から電子政府で利用されることを前提にした電子政府推奨暗号の選定に向けて、引続き暗号技術評価を実施してきた。

ここで、本プロジェクトの意義について、もう少し敷衍しておこう。ネットワーク社会における電子政府システムは、インターネットに代表されるオープンなネットワークをベースとして構

築されると想定される。このオープンなネットワーク上では、扱われる情報のセキュリティを確保する方策が本質的な重要性を持つ。この情報セキュリティ技術の骨格をなすのが暗号技術である。したがって、暗号技術を評価することは、ネットワーク社会における電子政府を実現する上で、最も意義深い事業の一つと言えよう。

暗号技術の厳正な評価は決して容易なことではない。一般に広く用いられる暗号は、暗号アルゴリズムを完全に公開し、十分な期間、多くの研究者の攻撃を受けることにより、ある程度信頼できる評価が可能となる。しかし、限られた期間内に電子政府システムを構築するためには、当事者が、現在の技術に基づいて、暗号の安全性レベルに対する判断を下さざるを得ない。このためには、適正な暗号技術評価の情報が必須である。OECD の暗号政策に関する勧告にもあるとおり、国民生活の基盤とも言える電子政府システムで使用する暗号技術については、システム構築者である政府機関が、自らの責任で主体的にこのような評価を行うべきである。暗号技術評価プロジェクトは、政府に課せられたこの責務の一翼を担うべく設立された事業であり、その役割は極めて重い。

とはいえ、現在の技術レベルでは、暗号の安全性を厳密に評価することは極めて困難である。また、将来にわたる安全性を保証することもできない。たとえば、「証明可能安全性」と呼ばれるものもあるが、これも現状では、ある仮定のもとに成立する安全性であり、安全性を判断する際の一つの重要な要素ではあるものの、これだけで安全と判断できるわけではない。暗号の安全性は、結局は、高度な専門知識を持ち経験を積んだ専門家により総合的に判断するしかないのである。

もちろん、専門家の間で意見が相違することもあるが、国際的に活躍し、第一線に立っている専門家の間には、多くの場合安全性に対し共通する感覚がある。本報告書では、できる限り、このような感覚を抽出し適切に表現するよう試みた。ただし、どうしても意見が一致しない場合には、あらゆる角度から十分な議論を尽くした上で、安全サイド、すなわち評価としては厳しい側に、結論を傾けた。これは、電子政府で実際に用いられる暗号技術を評価するという立場からはやむを得ないことである。

本暗号技術評価は、我が国では初めての事業であるため、米国 AES の公募選定事業を参考にし、さらに、欧州における暗号評価プロジェクト (NESSIE)、ISO/IEC の暗号技術国際標準化などの活動を考慮に入れながら、2000 年 6 月、2001 年 8 月に暗号技術を公募し、10 月に応募暗号説明会、2002 年 1 月に暗号技術評価ワークショップをそれぞれ実施し、評価の公平性・透明性を最大限に尊重しつつ進めてきた。その成果としての本報告書は、電子政府推奨暗号の選定に向けて、現時点で望み得る最も質の高い評価結果として「電子政府推奨暗号リスト素案」が示されており、今後の電子政府構築に大きな役割を果たすことができると考えている。本報告書の有効な利用を心から望む次第である。

今回の暗号技術評価報告は 3 年間の節目としての成果ではあるが、全てが完了したという訳ではない。暗号に関連する技術の進展とともに暗号の安全性は大きく変動するため、暗号技術評価事業の継続と、さらには、暗号技術を評価する専門機関の設立も望まれるところである。今回の暗号技術評価委員会の活動がこの礎になることを強く希望する。

暗号技術評価委員会はもとより、共通鍵暗号評価小委員会及び公開鍵暗号評価小委員会にも、

現在我が国の暗号技術開発の最前線に立っている研究者に出来うる限りご参加いただいた。各委員は多忙な日常業務があるにもかかわらず、この暗号技術評価が、電子政府の構築、ひいては 21 世紀におけるネットワーク社会の健全な発展に不可欠な歴史的意義を持つ事業であるとの認識の下、献身的にご尽力いただいた。特に、金子敏信東京理科大学教授、松本勉横浜国立大学教授には、共通鍵、公開鍵各評価小委員会委員長として本報告書の取りまとめに膨大な時間と労力とを割いていただいた。さらに、委員会における評価活動では、委員各位が持つ知識・経験だけでなく研究者のネットワークを全面的に活用していただいた。その結果、内外の学会を代表する多数の暗号専門家による評価を踏まえた厳正な評価が行われたものと確信している。

本評価事業が大きな成果を得たのは、暗号技術検討会の事務局として政策面の検討を推進した総務省、経済産業省の関係各位、暗号技術評価委員会の事務局として暗号評価を推進した TAO、IPA の関係各位に負う所が多い。このように、本報告書は多くの関係者の協力を得て、はじめて生み出すことのできた果実である。

末筆であるが、我が国初の暗号評価事業に、さまざまな立場でご協力いただいた関係者の皆様に併せて謝意を表する次第である。

2003 年 3 月

暗号技術評価委員会 委員長 今井 秀樹

本報告書の利用にあたって

本報告書の想定読者は、一般的な情報セキュリティの基礎知識を有している方である。たとえば、電子政府において、電子署名や GPKI システムなど暗号関連の電子政府関連システムに関係する業務についている方などを想定している。但し、個別暗号評価結果の記述部分などについては、ある程度の暗号技術の知識を備えていることが望まれる。

本評価報告書の 1 章には、暗号技術評価活動の概要を、2 章から 5 章には、各暗号技術の評価結果をまとめた。公開鍵暗号技術は 2 章、ブロック暗号やストリーム暗号の共通鍵暗号技術は 3 章、ハッシュ関数は 4 章、擬似乱数生成系は 5 章に記述した。実装攻撃に関する調査報告を 6 章に記述した。調達に際して必要な情報を 7 章にまとめる。

本暗号技術評価は、我が国最高水準の暗号専門家で構成される「暗号技術評価委員会」において、評価した結果であるが、暗号技術の特性から、その安全性を将来にわたって保証をしたものではなく、今後とも継続して評価・監視を実施することが必要であると考える。

今回の各暗号技術の評価は、2000 年及び 2001 年に実施した暗号技術の公募に応募された技術仕様に基づいて実施しているので、同一名称の製品版、または同様名称の暗号技術の ISO/IEC など他機関への提案暗号技術とは異なる場合がある。また、評価した暗号技術は、既にその暗号技術仕様が公開されているものを対象とした。その技術仕様については、2003 年 3 月の時点で CRYPTREC 事務局において応募者の Web サイトから情報を得ることができることを確認しているが、それら情報の不備などについては、本委員会は一切責任をもっていない。

なお、本報告書で報告する暗号技術の評価はアルゴリズムの評価であり、アルゴリズムの実装方法の評価、暗号製品や暗号モジュールの評価までは含まれていないことに注意を要する。アルゴリズムについて安全と判断される暗号技術に関しても、不注意な実装を行うことで安全性を低下させることもある。暗号技術の実装に伴い生じる恐れのある情報の漏洩などへの対策に関しては、暗号技術に関する「専門知識」を有する専門家の助言を受けるか、暗号技術に習熟した専門家が作成した「暗号ツール (ライブラリ)」を利用することを薦める。

本評価報告書に対する、意見や問合せなどのコメントは、CRYPTREC 事務局までご連絡していただけると幸いである。

評価委員会 委員名簿

暗号技術評価委員会 (肩書等は 2003 年 3 月現在)

委員長	今井 秀樹	東京大学 教授
顧問	辻井 重男	中央大学 教授
委員	岡本 栄司	筑波大学 教授
委員	岡本 龍明	日本電信電話株式会社 主席研究員
委員	金子 敏信	東京理科大学 教授
委員	松井 充	三菱電機株式会社 主席研究員
委員	松本 勉	横浜国立大学 大学院 教授

公開鍵暗号評価小委員会 (肩書等は 2003 年 3 月現在)

委員長	松本 勉	横浜国立大学 大学院 教授
委員	有田 正剛	日本電気株式会社 主任
委員	太田 和夫	電気通信大学 教授
委員	小暮 淳	株式会社富士通研究所 主任研究員
委員	酒井 康行	三菱電機株式会社 主任研究員
委員	静谷 啓樹	東北大学 教授
委員	新保 淳	株式会社東芝 主任研究員
委員	洲崎 誠一	株式会社日立製作所 ユニットリーダ研究員
委員	松崎 なつめ	松下電器産業株式会社 主席技師
委員	渡辺 創	独立行政法人産業技術総合研究所

共通鍵暗号評価小委員会 (肩書等は 2003 年 3 月現在)

委員長	金子 敏信	東京理科大学 教授
委員	青木 和麻呂	日本電信電話株式会社 研究主任
委員	荒木 純道	東京工業大学 大学院 教授
委員	川村 信一	株式会社東芝 主任研究員
委員	香田 徹	九州大学 大学院 教授

委員	古原 和邦	東京大学 助手
委員	櫻井 幸一	九州大学 大学院 教授
委員	佐藤 証	日本アイ・ビー・エム株式会社 主任研究員
委員	下山 武司	株式会社富士通研究所 研究員
委員	宝木 和夫	株式会社日立製作所 主管研究員
委員	館林 誠	松下電器産業株式会社 主席研究員
委員	角尾 幸保	日本電気株式会社 主任研究員
委員	時田 俊雄	三菱電機株式会社 主席研究員
委員	森井 昌克	徳島大学 教授

オブザーバー (肩書等は 2003 年 3 月現在)

山田 浩一	警察庁 情報通信局
飯田 晋一	警察庁 情報通信局
鳥居 秀行	警察庁 情報通信局
赤岩 司	警察庁 情報通信局
知識 親美	警察庁 警察情報通信研究センター
桝賀 政浩	防衛庁 長官官房情報通信課 (2002 年 7 月まで)
富田 哲	防衛庁 長官官房情報通信課
松井 教安	防衛庁 陸上幕僚監部 (2002 年 7 月まで)
一條 靖彦	防衛庁 陸上幕僚監部
喜安 拓	総務省 情報通信政策局 (2002 年 7 月まで)
金谷 学	総務省 情報通信政策局
門馬 弘	総務省 情報通信政策局 (2002 年 7 月まで)
藤本 昌彦	総務省 情報通信政策局
佐藤 憲一郎	総務省 情報通信政策局
福岡 晃	総務省 情報通信政策局
浦谷 眞人	総務省 行政管理局 (2002 年 7 月まで)
山本 寛繁	総務省 行政管理局
稲垣 浩	総務省 行政管理局 (2002 年 10 月まで)
中原 和郎	総務省 行政管理局
奥村 定夫	外務省 大臣官房
大野 秀敏	経済産業省 商務情報政策局
山本 文土	経済産業省 商務情報政策局 (2002 年 7 月まで)
田辺 雄史	経済産業省 商務情報政策局 (2002 年 7 月まで)
北浦 康弘	経済産業省 商務情報政策局
今井 俊博	経済産業省 商務情報政策局
木戸 達雄	経済産業省 産業技術環境局
深沢 太郎	経済産業省 産業技術環境局

滝澤 修　　通信総合研究所 情報通信部門

事務局

情報処理振興事業協会セキュリティセンター

内藤理、河内浩明、網島和博、黒川恭一、黒川貴司、武田仁己、竹谷清康 (2002 年 4 月まで)、田中公明、矢田健一、山岸篤弘

通信・放送機構

鈴木薫 (2002 年 7 月まで)、喜安拓、山田和晴 (2002 年 7 月まで)、横山隆裕、天野滋、笠井祥、神田雅透、熊谷正康 (2002 年 10 月まで)、田中秀磨、半澤則之、山村明弘

第 1 章

活動概要

オープンなネットワーク技術であるインターネットの発達はネットワーク上の様々なサービスを生み出した。安心してオープンネットワークの恩恵を受けるにはネットワーク上のセキュリティの確保が最重要課題となる。暗号技術はオープンなネットワークにおけるセキュリティ基盤技術であり、第三者による評価を通してその信頼性が高まる。暗号技術評価活動はわが国で初めての評価活動であり、産学官が協力し実施されたものである。3 年間の暗号技術の評価活動について紹介する。

1.1　沿革

暗号技術検討会及び暗号技術評価委員会 (CRYPTREC, Cryptography Research & Evaluation Committees の略) においては、2003 年度に構築予定の電子政府システムにおけるセキュリティの基盤技術である暗号技術の評価を、2000 年度から、3 年をかけて行ってきた。

1999 年度に情報処理振興事業協会 (IPA) において「政府調達情報セキュリティ標準 (基準) に関する調査研究」が実施される一方、総務省 (旧郵政省) においては「暗号通信の普及・高度化に関する研究会」が開催された。これらの研究会において、暗号技術が情報セキュリティ基盤技術として位置づけられたことによりその信頼性等を技術的専門的見地から客観的に検証する必要性が認識され、今後直ちに、暗号技術の評価体制を整備して暗号技術の評価を実施するべきであるとの提言がなされた。

これを受けて 2000 年 5 月には、電子政府において利用可能な暗号技術の評価に関する業務が経済産業省 (旧通商産業省) から IPA に委託された。暗号技術評価委員会を設立し、暗号技術の評価活動 (CRYPTREC) を開始した。同年 6 月には、暗号技術が公募され、47 件の応募が国内外から寄せられた。

2001 年度からは通信・放送機構 (TAO) が CRYPTREC に事務局として加わり、IPA と共同して暗号技術評価委員会の運営にあたった。また、暗号技術の利用に関し政策的な観点から検討を行うことを目的として、総務省技術総括審議官と経済産業省商務情報局長の連名の下で暗号技術検討会が組織され、電子政府システムにおける暗号技術が備えるべき要件の整理、暗号の利用方法、調達方法に関する調査・検討が行なわれた。同年 8 月には 2 回目の公募が行われ、1 回目の

公募も含めて合計 63 件の応募が国内外から寄せられた。また暗号技術検討会から要請があり、SSL3.0/TLS1.0 で利用される暗号技術の評価を 2001 年度、2002 年度にわたり実施した。同年 4 月には電子署名及び認証業務に関する法律が施行されて、電子署名の利用に関する法整備も整い、電子署名及び認証業務に関する法律の指針に示される電子署名の安全性評価も暗号技術評価委員会で担うことになった。

2002 年度は、暗号技術評価委員会は、暗号技術検討会と引き続き連携して、暗号技術評価を進め、電子政府システムに資する暗号技術を掲載する「電子政府推奨暗号リスト素案」(表 1.3) を完成させた[*1]。また、ISO/IEC JTC1/SC27 WG2 ワルシャワ会議と NESSIE に CRYPTREC の活動と評価状況を紹介した。暗号技術の評価活動の概略を表 1.1 にまとめる。

表 1.1: 暗号技術評価委員会の主な活動

2000 年 5 月	暗号技術評価委員会の設置
2000 年 6 月 –7 月	2000 年度暗号技術の公募
2000 年 8 月 –10 月	2000 年度暗号技術スクリーニング評価
2000 年 10 月	暗号技術シンポジウム (CRYPTREC の目的紹介)
2000 年 10 月 –2001 年 3 月	2000 年度暗号技術の詳細評価
2001 年 3 月	CRYPTREC Report 2000 の発行
2001 年 4 月	暗号技術評価報告会 (2000 年度 CRYPTREC 報告)
2001 年 8 月 –9 月	2001 年度暗号技術の公募
2001 年 10 月	応募暗号説明会
2001 年 10 月 –2002 年 3 月	暗号技術の詳細評価
	2001 年度新規応募暗号技術のスクリーニング評価
2002 年 1 月	暗号技術評価ワークショップ (評価の状況報告)
2002 年 3 月	CRYPTREC Report 2001 の発行
2002 年 4 月	暗号技術評価報告会 (2001 年度 CRYPTREC 報告)
2002 年 4 月 –2003 年 2 月	暗号技術の詳細評価
	電子政府推奨暗号リスト素案の作成
2002 年 10 月	CRYPTREC Report 2001 (英語版) の発行
2002 年 10 月	ISO/IEC JTC1/SC27 WG2 ワルシャワ会議で CRYPTREC の活動を紹介
2003 年 2 月	NESSIE にて CRYPTREC の活動を紹介
2003 年 4 月	CRYPTREC Report 2002 の発行
2003 年 5 月	暗号技術評価報告会開催予定 (2002 年度の CRYPTREC 報告、他)

1.2 CRYPTREC の体制

CRYPTREC の体制は、暗号技術検討会と暗号技術評価員会の 2 つの委員会から構成される。暗号技術検討会は主に政策的な判断を行い、暗号技術評価委員会は技術的評価を実施した。暗号技術評価委員会と暗号技術検討会の各委員には産学官から情報セキュリティに関する有識者を集め、我が国の情報通信セキュリティについて産業界、大学、行政の各方面からの意見が反映されるように構成した。暗号技術評価委員会においては、公開鍵暗号評価小委員会と共通鍵暗号評価小委員会の 2 つの小委員会が組織され、IPA と TAO が事務局を務めた。公開鍵暗号評価小委員会では公開鍵暗号技術の評価を、共通鍵暗号評価小委員会では共通鍵暗号技術、ハッシュ関数、擬似乱数生成系の評価をそれぞれ担当した (図 1.1 参照)。暗号技術検討会においては、暗号調達

[*1] 電子政府推奨暗号リスト素案は、その後の暗号技術検討会での審議ならびにパブリックコメント募集を経て、総務省及び経済産業省によって、原案通り、「電子政府推奨暗号リスト」に最終決定された。

ガイドブック作成ワーキンググループが組織されて暗号技術の利用と調達方法に関する「暗号調達のためのガイドブック」を作成した。暗号技術の評価活動全体の把握のためには、本報告書のみならず、「暗号技術検討会 2002 年度報告書」も併せて参照していただきたい。また、各種の電子政府システムを構築する目的で暗号技術を選択、利用する場合には、「調達のためのガイドブック」を参照[*2] していただきたい。

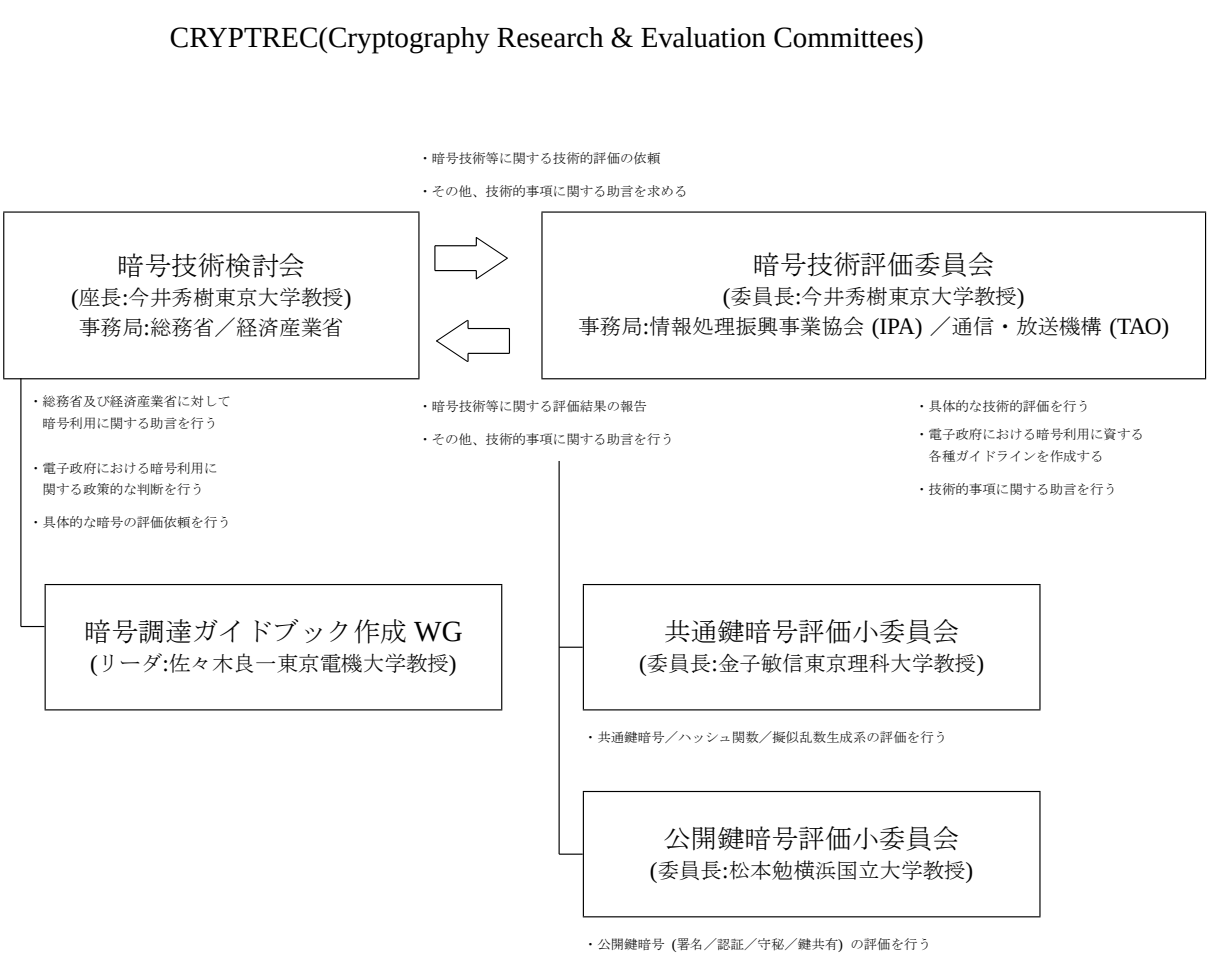

図 1.1: CRYPTREC 体制 (2002 年度)

1.3 評価活動の背景

暗号技術の評価の社会的要請につながるいくつかの要因を紹介する。

■電子政府システム構築 2001 年に高度情報通信ネットワーク社会推進戦略本部で決定された e-Japan 重点計画、及び、2002 年に同本部で決定された e-Japan 重点計画 –2002 では、行政の情

[*2] http://www.soumu.go.jp/s-news/2003/pdf/030331_4_1.pdf, http://www.meti.go.jp/kohosys/press/0003876/

報化及び公共分野における情報通信技術の活用の推進が重点政策の一つに挙げられており、その施策の意義として「電子政府」の実現が掲げられている。また、高度情報通信ネットワークの安全性及び信頼性の確保のための具体的施策の一つとして、

> 客観的にその安全性が評価され、実装性に優れた暗号技術を採用するため、2002 年度中に、ISO、ITU 等における暗号技術の国際標準化の動向を踏まえ、専門家による暗号技術検討会の開催等を通じて電子政府利用等に資する暗号技術の評価及び標準化を行う。

と報告された。

2000 年 (平成 12 年) に「高度情報通信ネットワーク社会形成基本法 (IT 基本法) (平成 12 年法律第 144 号)[*3]」が整備され、IT 関連製品の評価・認証制度が整った。さらに企業も電子政府システムソリューションを揃えることとなり、電子政府システムの本格的な始動が目前に迫った。e-Japan 重点計画の決定を受けて、電子政府システムにおけるセキュリティ確保が重要な案件となり、内閣官房の情報セキュリティ対策推進会議[*4] において議論され、「電子政府の情報セキュリティ確保のためのアクションプラン[*5]」(2001 年:平成 13 年 10 月 10 日) として提言がなされた。暗号技術の評価については、

> 「電子政府」におけるセキュリティ確保のためには、政府調達における一定水準のセキュリティ確保のための情報機器等に関する基準 (具体的には ISO/IEC 15408) を可能な限り利用することと同様、暗号についても、一定水準以上の安全性及び信頼性を有するものの利用が不可欠であり、これを推進することが必要である。このため、総務省及び経済産業省は、両省で実施している研究会の成果等も踏まえ、2002 年度中に「電子政府」における調達のための推奨すべき暗号のリストを作成し、これを踏まえ、各省庁における暗号の利用方針について合意を目指す。

と報告されており、「電子政府システムにおける基盤技術である暗号技術を評価し省庁間の合意を得て推奨すること」が重要な課題とされた。政府の情報セキュリティ施策については http://www.bits.go.jp/about/about.html に詳しい情報がある。

■電子署名及び認証業務に関する法律 「電子署名及び認証業務に関する法律 (平成 12 年法律第 102 号)[*6] 」が 2001 年 (平成 13 年 4 月 1 日) に施行され、電子署名が手書きの署名や押印と同等に通用する法的基盤が整備された。さらに、2002 年 (平成 14 年 12 月) には、「電子政府・電子自治体の推進のための行政手続オンライン化関係三法[*7] (「電子署名に係る地方公共団体の認証業務に関する法律」[*8] 等)」が国会で成立した。2001 年度の暗号技術検討会の報告書においても「地方公共団体については、普及・広報等を通じて本利用方針の活用を奨励していく」と

[*3] http://www.kantei.go.jp/jp/it/kihonhou/honbun.html
[*4] http://www.bits.go.jp/index.html
[*5] http://www.kantei.go.jp/jp/it/security/suisinkaigi/dai4/actionplan.html
[*6] http://www.soumu.go.jp/joho_tsusin/top/ninshou-law/law-index.html 及び http://www.meti.go.jp/policy/netsecurity/digitalsign.htm
[*7] http://www.soumu.go.jp/gyoukan/kanri/sanhou.html
[*8] http://www.soumu.go.jp/kyoutsuu/syokan/pdf/020607_3c.pdf

記載されている。このため、「電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針 (平成 13 年総務省 法務省 経済産業省告示第 2 号)[*9]」に十分な安全性を有した暗号技術を選択して記載することが、電子政府及び地方公共団体における情報通信システムのセキュリティ確保のために、重要な課題となった。また、海外の電子署名法関連の制度については http://www.soumu.go.jp/joho_tsusin/top/ninshou-law/ に簡単にまとめられているように、多くの国で法制度が整備されている。

■国際的な暗号評価活動・標準化活動 暗号技術の評価においては、暗号技術の公募を提案者の出身国や組織に制限することなく行い、なおかつ、公平に評価・選択することが高い安全性を達成している暗号技術を得ることにつながる。その結果、調達者・利用者、さらには国民の利益となることはもちろんのこと、国内外からの高い信頼を得ることになる。国外における暗号技術の評価活動のうち、第三者の評価による安全性が議論・検証されてきたプロジェクトとしては下記の 2 つが代表的であり、ともに暗号技術を広く国際的に公募し、国や企業よる制限をなくして評価を行っている。

米国での AES (Advanced Encryption Stantard) プロジェクトと欧州における NESSIE (New European Schemes for Signatures, Integrity and Encryption) プロジェクトの二つが代表的である。ともに暗号技術を広く国際的に公募し、国や企業よる制限をなくして評価を行った。

1997 年に米国商務省傘下の技術標準化組織である NIST (National Institute of Standards and Technology) [*10] が Data Encryption Standard(DES、Triple DES) に替わる次世代標準化暗号の選定計画を開始し、2000 年 10 月、ベルギーの研究者から応募された Rijndael [*11] が米国政府標準暗号 Advanced Encryption Stantard (AES) として選出され、2001 年 11 月に FIPS PUB 197 として正式に制定された。

NESSIE プロジェクト[*12]は、2000 年に開始された欧州における暗号アルゴリズム評価活動であり、欧州連合 EU (European Union) の下部機関である欧州委員会 (European Commission) の情報社会技術プログラム (Information Societies Technology Programme) の一環として実施されているものである。公開鍵暗号、共通鍵暗号、ハッシュ関数を含むあらゆる暗号の部品に関する推奨リストを作成するという意欲的なプロジェクトであり、2003 年 2 月 26 日に 3 年間の評価活動の最終結果として推奨暗号リスト (NESSIE portfolio) が報告されている。詳細については http://www.cryptonessie.org を参照してほしい。

標準化としては国際標準化機構 ISO の活動がある。ISO/IEC JTC1/SC27[*13] においては、暗号アルゴリズム標準 ISO/IEC 18033 を作成中である。Part 1 が総論、Part 2 が公開鍵暗号、Part 3 がブロック暗号、Part 4 がストリーム暗号となっており、2003 年 3 月現在、18033-1 は Final

[*9] http://www.soumu.go.jp/joho_tsusin/top/ninshou-law/1-6.pdf 及び http://www.meti.go.jp/policy/netsecurity/digitsign_sisin.htm
[*10] http://www.nist.gov/
[*11] http://www.esat.kuleuven.ac.be/~rijmen/rijndael/
[*12] http://www.cryptonessie.org/
[*13] http://www.din.de/ni/sc27/

Committee Draft (FCD) に、18033-2~4 は Committee Draft (CD) に進んでいる。

また暗号技術の登録制度として ISO/IEC 9979 (JIS X 5060) がある。ISO/IEC 9979 においては暗号アルゴリズムを登録するのみで、安全性と処理性能についてまったく評価は行われていない。このため ISO/IEC 9979 に登録されていることが技術的な性能を保証するものではないことに注意を要する。

■国外の情報通信セキュリティ評価機関 暗号技術に対する評価は明確な評価基準が必要となる。一方、評価基準の設定には多くの項目が複雑に関連しているので簡単ではない。利用目的や運用状況によって基準が異なるので、暗号の選択の判断は利用者に委ねられるべきものである。実際に各国政府のセキュリティ確保のための技術・システムの評価・選択に関しては政府の機関が担当していることが多い。たとえば、米国 NIST、カナダ CSE (Communications Security Establishment)[*14]、英国 CESG (Communications Electronics Security Group)[*15]、フランス DCSSI (Direction Centrale de la Securite des Systemes d'Information)[*16]、ドイツ BSI (Bundesamt fur Sicheit in der Informationstechnic)[*17]、オーストラリア DSD (Defense Signals Directorates)[*18]、ニュージーランド GCSB (Government Communications Security Bureau)[*19]、韓国 KISA (Korea Information Security Agency)[*20]等が各国の情報通信セキュリティの評価を担当している。わが国でも遅れることなく継続的かつ中立な (開発者やベンダに左右されない) 評価機関において暗号評価を担う体制を整備していく必要がある。暗号評価機関の役割としては、暗号の技術的な評価、暗号開発・運用に関する支援、暗号技術の動向の監視とその情報を蓄積と一般への公開、利用方法の啓発と人材育成などが考えられる。上記の各国の機関においても総合的に情報通信セキュリティの確保に対応している。国際的な機関との連携と同時に、国内においては地方公共団体における情報通信セキュリティ担当部署とも情報交換をしていくことが求められる。

■ 客観的な技術情報の提供 情報機器のセキュリティに関する評価基準として ISO/IEC 15408 があるが、そこには暗号技術に関しての明確な規定はない。しかしそれは、暗号技術の評価を行う必要性が無いことを意味しているのでは決してなく、むしろ安全な暗号技術を利用することを、ISO/IEC 15408 における評価以前の、「前提条件」としているためである。

技術仕様が公知 (詳細な情報が不特定多数の方が自由に入手できる状態) でなく、第三者による安全性評価が行われていない暗号技術は、開発者が主張する安全性を本当に有しているのかの確認が客観的になされていないことを意味する。しかも、アルゴリズム仕様が公知の状態でない場合には、開発者のみが知り得る秘密情報 (トラップドア) が存在する可能性さえも排除できない。このように、アルゴリズム仕様が非公開であり、安全性の評価も第三者によって十分になされていない暗号アルゴリズムの場合、(仮にアルゴリズム仕様が知られたとしても) 安全性に問題が生

[*14] http://www.cse-cst.gc.ca/
[*15] http://www.cesg.gov.uk/
[*16] http://www.ssi.gouv.fr/
[*17] http://www.bsi.bund.de/
[*18] http://www.dsd.gov.au/infosec/
[*19] http://www.gcsb.govt.nz/
[*20] http://www.kisa.or.kr/

じないような仕様になっているのか、それともアルゴリズム仕様が知られた瞬間に安全性の問題が生じるような仕様になっているのかの判断が一般にはつきにくい。しかも、(不正な手段を含む) 何らかの理由によって、ある日突然、アルゴリズム仕様が公になる危険性が常にあり、とりわけ後者のような暗号アルゴリズムであった場合、その影響は大きい。したがって、暗号技術に関しては十分な評価を中立かつ公平な第三者により実施され、安全であると確認されたものを利用することが電子政府システム調達に関してはもちろんのこと、一般のシステムにおいても重要であると考える。なお、市販の商用ソフトウェアには、アルゴリズム仕様が非公開の上、安全性の評価が第三者によって十分になされていない暗号技術を採用しているものも相当数あり、その中には技術的に未熟と判断せざるを得ない場合もあることに注意を要する。

一方、第三者による暗号技術の客観的な評価結果は簡単に手に入る状況にはない。この点からも、暗号技術の安全性・処理性能に関する、第三者による客観的な評価結果を広く社会に提供し、誰もが必要な情報を入手できる制度や評価機関が必要となってきているといえる。

1.4 公募と評価対象

前節で紹介したさまざまな社会的状況から暗号技術評価の重要性が認識され、2003 年に開始が予定されている電子政府システムの情報セキュリティの確保のためには暗号技術の安全性の検査・確認が不可欠となり、CRYPTREC において暗号技術の評価を実施することになった。

暗号技術評価委員会は、電子政府に資する暗号技術をリストにまとめるため、2000 年度及び 2001 年度に暗号技術の公募を行った。公募に際しては、提案者の出身国や組織に制限を設けずに、すべての提案者に公平に評価を受ける機会を提供し、CRYPTREC 活動の中立・公平性を重視した。すべての提案暗号が電子政府推奨暗号の候補として差別なく平等に評価を受けることが重要である。また、暗号技術評価委員会にて自らが選定したいくつかの暗号技術を「その他評価が必要な暗号」として評価対象に指定した。さらに電子署名法や標準化機関からの依頼など特別な理由により、いくつかの暗号技術を「特定評価」対象暗号として評価した。基本的に評価対象はこれら「応募暗号技術」、「その他評価が必要な暗号」、「特定評価対象暗号技術」の 3 つに分類することが出来る。

■応募暗号技術 2000 年 6 月 13 日から 7 月 14 日及び 2001 年 8 月 1 日から 9 月 27 日の二回にわたり、公開鍵暗号技術として署名、認証、守秘、鍵共有のカテゴリで、共通鍵暗号技術として 64 ビットブロック暗号、128 ビットブロック暗号、ストリーム暗号のカテゴリで、さらに、ハッシュ関数、擬似乱数生成系のカテゴリで、2002 年度末までに調達可能なものを公募した。2000 年度と 2001 年度をあわせて 63 件の応募があった。

公募においては、提案者に、「暗号技術応募書」、「暗号技術概要説明書」、「暗号技術仕様書」、「自己評価書 (提案者自身による提案暗号の評価報告)」、「テストベクタ」、「参照 (reference) プログラム」、「公開の状況等に関する情報」、といった提出物を求めた。暗号評価には技術仕様の公開が絶対的な前提条件となるため、応募者に以下のような 2 つの条件を遵守するように要請した。

(A1) 暗号技術概要説明書、暗号技術仕様書、自己評価書が、公知 (不特定多数の方が自由に入

手できる情報) であること

(A2) 応募者側で開設したウェブ・ページ上において、仕様等が公開されているか、もしくは不特定多数の方が仕様等を自由かつ困難なく入手するための具体的方法が公開されていること

■その他評価が必要な暗号 応募された暗号技術に加えて、比較的長い期間にわたる使用実績と評価実績があるか、または広範囲に利用されてきている暗号技術について、電子政府システム構築のために必要と考えられる場合には該当する暗号技術を「その他評価が必要な暗号技術」として、応募者の有無に関わらず評価対象に指定した。

その他評価が必要な暗号として評価したものは、DSA, ECDSA (ANSI X9.62), RSASSA-PKCS1-v1_5, DH, Triple DES, AES, MD5, RIPEMD-160, SHA-1, SHA-256, SHA-384, SHA-512, PRNG for DSA in FIPS PUB 186-2 Appendix 3[*21]である。

さらに、2002 年度には、PRNG in ANSI X9.42-2001 Annex C.1/C.2, PRNG in ANSI X9.62-1998 Annex A.4, PRNG in ANSI X9.63-2001 Annex A.4, PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1, PRNG in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1/3.2 の 5 つを評価した。

■特定評価対象暗号技術 特定評価対象暗号とは、「応募暗号技術」であるかどうか、もしくは「その他評価が必要な暗号」に指定されているかどうかに関わらず、特別な要請に基づいて、CRYPTREC において評価を実施した暗号技術である。原則として、特定評価対象暗号技術はある特定の目的のために評価されるものであるため、その暗号技術を電子政府推奨暗号の候補とするかどうかはその評価目的に依存し、自動的に電子政府推奨暗号候補となるわけではない。2000−2002 年度の CRYPTREC の特定評価は基本的に以下の 3 つに分類される。

(B1) 電子署名法の指針に記載される暗号技術

平成 13 年総務省 法務省 経済産業省告示第 2 号 (平成 13 年 4 月 27 日官報号外第 86 号) の電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針に掲載されていた DSA, ECDSA (ANSI X9.62), ESIGN, RSASSA-PKCS1-v1_5 の各署名技術を「特定評価」の対象とした。電子署名法の指針に関わる暗号技術は電子政府推奨暗号候補として評価された。DSA, ECDSA (ANSI X9.62), RSASSA-PKCS1-v1_5 は「その他評価が必要な暗号技術」にも指定されている。ESIGN は 2001 年度の応募暗号技術でもある。

(B2) SSL3.0/TLS1.0 で利用される暗号技術

暗号技術検討会の要請で、高い利用実績があり、電子政府システムにおいても利用されることが予想される暗号プロトコル SSL/TLS に組み込まれている暗号技術の評価を実施した。SSL/TLS において利用される暗号技術として評価の依頼があったものは RSA (RSAES-PKCS1-v1_5, RSASSA-PKCS1-v1_5), DES/Triple DES (40, 56, 168 ビット鍵), RC2 (40, 128 ビット鍵), RC4 (40, 128 ビット鍵) である。SSL3.0/TLS1.0 で利用される暗号技術は電子

[*21] CRYPTREC Report 2001 で、PRNG based on SHA-1 としていたものを含む。

政府推奨暗号候補として評価し、そのうち電子政府で利用可能なものを電子政府推奨暗号リスト素案に掲載した。RSASSA-PKCS1-v1_5 と Triple DES は「その他評価が必要な暗号技術」でもある。

(B3) ISO/IEC JTC1/SC27 協力

2001 年度に ISO/IEC JTC1/SC27 国内委員会から暗号技術検討会に韓国政府標準暗号であり ISO/IEC 18033 に提案されている 128 ビットブロック暗号 SEED の評価依頼がなされた。暗号技術評価委員会では、国際貢献・国際協力という観点からこの依頼を受諾し、特定評価の対象として評価を実施した。SEED は ISO/IEC JTC1/SC27 への協力の観点から評価された暗号技術であり、応募もされていないので、電子政府推奨暗号候補とはしなかった。

■評価対象暗号技術 2002 年度評価対象暗号技術 (表 1.2) についてまとめる。本報告書においては、応募暗号技術の名称は応募者の提示した名称を、その他評価が必要な暗号と特定評価対象暗号については仕様書に記された名称を用いる。

表 1.2: 2002 年度評価対象暗号技術

応募暗号技術	ECDSA (SEC1), ESIGN, RSA-PSS, RSA-OAEP, ECIES (SEC1), HIME(R), ECDH (SEC1), PSEC-KEM, CIPHERUNICORN-E, Hierocrypt-L1, MISTY1, Camellia, CIPHERUNICORN-A, Hierocrypt-3, RC6 Block Cipher, SC2000, MUGI, MULTI-S01
その他評価が 必要な暗号技術	RSASSA-PKCS1-v1_5, DSA, ECDSA (ANSI X9.62), DH, Triple DES, AES, RIPEMD-160, SHA-1, SHA-256, SHA-384, SHA-512, PRNG for DSA in FIPS PUB 186-2 Appendix 3 PRNG in ANSI X9.42-2001 Annex C.1/C.2, PRNG in ANSI X9.62-1998 Annex A.4, PRNG in ANSI X9.63-2001 Annex A.4, PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1, PRNG in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1/3.2
特定評価暗号技術	RSASSA-PKCS1-v1_5, DSA, ECDSA (ANSI X9.62), ESIGN, RSAES-PKCS1-v1_5, 3-key Triple DES, RC4 (40, 128 ビット鍵)

1.5 評価・選定

電子政府推奨暗号リスト素案の作成にあたっては電子政府システムにおける利用において十分に安全と判断されたレベル (暗号強度が十分に高い) の暗号技術を選択するために安全性評価を行い、処理速度とリソース使用量を把握するためにソフトウェア実装評価およびハードウェア実装評価も実施した。公平性を確保し、十分な技術的評価を期するために、暗号技術評価委員会委員以外の複数の専門家にも評価 (外部評価と呼ばれる) を依頼した。また、同じカテゴリに分類される暗号技術に関しては、相対的に比較出来るように同様の評価手法を可能な限り適用して平等な評価を心掛けた。

近年、実装攻撃 (サイドチャネル攻撃とも呼ばれる) の研究が活発になってきている。実装攻撃は、暗号アルゴリズムを実装した時に、その実装環境において物理的デバイスから漏洩したり、不正な行為を行って得られた物理量 (処理時間や消費電流等) やエラーメッセージを利用して攻撃対象の秘密情報を得る攻撃である。2000−2002 年度の CRYPTREC は主にアルゴリズムの評価活動を実施しており、実装攻撃に関してはこれまでの研究の状況と最新の研究動向について調査した。

■評価項目 アルゴリズムの性質と特徴をより良く理解して、十分な安全性、処理性能、実装性能を有するかどうかを効率良くかつ迅速に判断するために、評価は段階的にかつ平行して進められた。

(C1) スクリーニング評価
仕様が公知であるかどうかを確認し、設計思想、設計指針、安全性、実装性能に問題がないかを提出書類から判断した。また、第三者が実装を行うに足る十分な情報が提出書類に含まれているかどうかを検査した。

(C2) 詳細評価
既知の攻撃が適用可能かどうかの検査、既知の攻撃法が成功するために必要とされるコストの見積もり、証明可能安全性の妥当性、パラメータや鍵の設定方法の妥当性、補助関数の選択やスキームの中での補助関数の利用方法、提案暗号技術を実際に運用する際に起こり得る問題点などを調べた。さらに評価者独自の視点による攻撃を試みて、実際に攻撃が可能かどうかを調べた。関連する暗号技術と比較したり、対象の暗号技術特有の利点や欠点を抽出した。

(C3) ソフトウェア実装評価
ソフトウェア実装評価においては、(1) 一般的と思われる PC 環境、(2) 最も普及していると思われるサーバ環境、(3) 高性能を実現しているハイエンド環境、の 3 つの環境において計算機のリソースとの整合性や使用環境との整合性、提出書類に記載通りに動作するかどうかの確認等を実施した。ブロック暗号についてはすべての応募暗号に対して、PC 環境下での評価を実施し、サーバ環境とハイエンド環境下については応募者の選択とした上で、評価を実施した。8 ビット CPU 等のロースペック環境 (IC カード環境) については、128 ビッ

トブロック暗号の一部に対して、Z80 シミュレータ上での評価を実施した。ストリーム暗号については MUGI と MULTI-S01 に対して PC 環境下での評価を実施した。ハッシュ関数と擬似乱数生成系に対してはソフトウェア評価を実施しなかった。公開鍵暗号については PC 環境下で提案者から提出されたテストベクトルを使用し、提出書類に記載通りに動作するかどうかを確認した。

(C4) ハードウェア実装評価

「応募書類 (アルゴリズム仕様書およびテストベクトル)」だけに基づき、第三者によるハードウェア設計が可能かどうかを検証した。評価対象としては、共通鍵暗号技術に限定したが、それでも評価対象暗号技術が 12 種類と多数あることから、実装評価の容易性の観点から FPGA をターゲットデバイスとした実装評価を試みた。その結果、全ての共通鍵ブロック暗号 (10 種類) およびストリーム暗号 (2 種類) は応募書類のみから実装可能であることが確認できた。

(C5) 関連調査

近年、実装攻撃に対しての安全性の研究が急速に発展している。電子政府システムの調達者および実装者に注意を喚起するため実装攻撃について調査を行った。

また、公開鍵暗号技術の安全性の基礎となる素因数分解問題についてはセミナーを開催して、計算機実験を行う準備をすすめている。素因数分解問題解法用のハードウェア回路を実現する研究も進みつつあり、これについても現状について調査した。

さらに、SSL3.0/TLS1.0 の利用に関する調査を行った。

■評価基準 暗号評価にあたって、各カテゴリごとに以下の評価基準が設定された。

(D1) 公開鍵暗号技術 (詳細は 2 章を参照)

- 比較的長い期間にわたる使用実績・評価実績があり、インターオペラビリティの観点から仕様の変更を簡単には行えない公開鍵暗号技術については、多くの研究者から十分な評価研究を受けているが、実運用における安全性に関する問題点が指摘されていないことを求めた。
- 使用実績が少ない新しい公開鍵暗号技術については、既存暗号技術とは独立に仕様を定めることができることから、最低限の条件として証明可能安全性が示されていることを必須とした。それに加えて、プリミティブが依存する数論的な問題の困難性や、推奨パラメータの選択方法や補助関数のスキームの中での利用方法を含めて総合的に安全性を評価した。

(D2) 共通鍵暗号技術 (詳細は 3 章を参照)

以下の条件のいずれかを満たすことを求めた。

- 現時点の最良のアルゴリズム解読技術を適用しても、秘密鍵の総当たりである 2^{128} 以上の計算量が必要と判断されるもの。特に、差分攻撃や線形攻撃などの代表的な攻撃法について、安全性が確認されているもの。
- 世界的に広く使用され、かつ多くの研究者から十分な評価研究を受けているが、実運用における安全性上の大きな問題点が指摘されておらず、現時点で安全と判断される

もの。この場合、2^{100} 以上の解読計算量を目安とした。

(D3) ハッシュ関数 (詳細は 4 章を参照)

以下の条件のいずれかを満たすことを求めた。

- 特定の出力値に対する入力値を発見する計算量が、最良のアルゴリズム解読技術を適用しても、総当りに相当する計算量から著しく低下しないと判断され、かつ出力値が一致するような異なる入力値を発見する計算量が最良のアルゴリズム解読技術を適用しても 2^{128} 以上であること。
- 世界的に広く使用され、かつ実運用における安全性上の問題点が指摘されていないこと。なお、全ハッシュ値の長さが少なくとも 160 ビット以上であること。

(D4) 擬似乱数生成系 (詳細は 5 章を参照)

以下の条件をすべて満たすことを求めた。

- 統計的性質が真性乱数に近く、かつ、既知の出力ビット履歴から未来又は過去の未知出力ビットが予測困難であること。
- 擬似乱数生成系を使用するシステムの総当たり攻撃に対し、十分に耐性をもつ程に、実質的入力空間が大きいこと。
- 統計的性質は、例えば SP800-22 等の代表的な擬似乱数検定テストに合格することが期待されること。

■暗号技術仕様の安定性 暗号理論の発展に応じて、仕様の変更はさまざまな暗号に対して行われている。それはよりよい技術への移行を意味する。一方で、暗号技術の調達者・利用者の観点からは、仕様が度々変更されたり、同じ暗号技術の名称に対してむやみに多い仕様が並存することは、システムの再構成や暗号技術の交換を迫られたり、調達時に判断を難しくするなどの障害を引き起こす。仕様が度々変更されることに起因する利用者の不利益を極力減らすということは調達者・利用者からの切実な要請である。仕様変更が、利用者への配慮なくして、開発者の意向で行われることは利用者に対する不利益を生み出す。利用者に対する配慮を無視して仕様が再三変更される技術は不安定であると考えることも出来る。仕様が変更されることは、暗号理論の進展から余儀なくされる場合もあるが、一方で開発者が十分な安全性評価を実施していないことの証拠とも解釈出来る。暗号理論の進展とは無関係に暗号技術の利用者は調達した暗号技術を利用しつづけることになるわけであり、十分に調達者・利用者に配慮を行う姿勢が望まれる。現実のシステムに採用するためには、最新の暗号理論を盛り込む技術と同様に、暗号技術の維持管理 (仕様がすぐに変更されるようなことがなく) や調達後のサポートに信頼がもてる暗号技術が望ましい。電子政府推奨暗号には技術的な視点からも仕様の安定性が求められた。

■提案暗号の仕様の変更について 応募技術に関しては応募後の仕様の変更を原則として認めなかった。AES や NESSIE では仕様の変更を認めているが、CRYPTREC とは状況がいくつか違った。仕様の変更を認めなかった理由が主に三つある。

一つ目の理由は、2003 年度に開始される電子政府システム構築において、システム構築のスケジュールに支障を来たさないようにするためである。暗号技術の調達作業は 2003 年度に開始されることが予想され、調達が 2003 年度に可能であることが絶対条件であった。大きな仕様の

修正を認めた場合に、評価期間が延長されて 2003 年度開始予定の電子政府システム構築に遅延をもたらす結果になることが予想されるため、仕様の変更は認めなかった。公募に際しても、「2002 年度末に調達が可能」であることが条件であることを明確に提示した。

二つ目は暗号評価の資源の平等な分配が望まれたことである。公平かつ中立な評価を実現するため、評価資源 (予算、人材、時間) はすべての暗号技術に可能な限り平等に配分されるべきであり、ある特定の暗号技術に過剰な資源をつぎ込むことはできないと考えた。大きな仕様の変更を許すことは、暗号技術評価委員会が修正案を作成することや、修正された仕様の再評価を要請することになり、結果として特定の暗号技術への偏った資源の配分を引き起こすことにつながる。公共事業である本プロジェクトにおいては、限りある評価資源を公平にかつ効率的に配分することが望まれ、特定の暗号技術への偏った評価は避けなければならないと判断した。従って、仕様の変更を許し仕様変更後の新仕様の暗号技術を評価することは行わない方針となった。

三つ目は、応募時点で暗号技術に高い完成度を求めたことである。電子政府システムに利用され、かつ今後長期間に渡り利用されることを考慮すれば、安全性と共に仕様が安定していることが必要である。電子政府システムに採用した後ですぐに仕様の変更が予想される暗号技術は利用者の立場を考慮すれば推奨できない。政府調達の暗号技術として推奨されるものであるから、十分な安全性と同様に仕様の安定性も求められる。

■電子政府推奨暗号リスト素案の作成に関する要請項目 2002 年度暗号技術検討会において、電子政府システム構築のために十分に安全性が確認された暗号技術のリストを作成することが重要であると報告された。これを受け、電子政府システムにおける認証、鍵共有、守秘、署名の機能を実現する暗号技術について、2001 年度の電子政府暗号候補を評価し、以下の 3 点に留意して電子政府推奨暗号リスト素案を作成することが暗号技術検討会から暗号技術評価委員会に対し要請された。

(E1) 電子政府利用に資すると判断されたレベルの (10 年間安心して利用できる) ものをいたずらに多くならないように複数個選ぶ。
(E2) 一般に使われる商用ソフトに予め入っているか、入る可能性の高いものが各カテゴリに 1 つは選択される。
(E3) 電子政府推奨暗号に選定されたものと同一仕様の暗号を確実に調達出来るようにするという観点から、各電子政府推奨暗号の仕様を確認する。

また、これら 3 点に加えて、電子政府推奨暗号リスト素案に掲載される暗号技術は、2003 年度開始の電子政府システム構築において利用されることを想定しているため、2003 年の時点で仕様が特定され、技術仕様が明確に規定される仕様書が存在して調達が現実に可能であり、ライセンス方針が明確であることも重要な要件とした。

1.6　電子政府推奨暗号リスト素案

3 年間の CRYPTREC プロジェクトの集大成として暗号技術評価委員会で作成された「電子政府推奨暗号リスト素案」は暗号技術検討会に提出され、暗号技術検討会での審議ならびに (総務省・経済産業省による) パブリックコメント募集を経て「電子政府推奨暗号リスト」とされた。そして、「各府省の情報システム調達における暗号の利用方針 (平成 15 年 2 月 28 日、行政情報システム関係課長連絡会議了承)」により、各府省における暗号技術の利用方針として合意された。

表 1.3 に電子政府推奨暗号リスト素案を示す。電子政府システムで利用する際に注意が必要な暗号技術については、注釈にその旨を記載した。

表 1.3: 電子政府推奨暗号リスト素案 (2002 年 11 月作成)

技術分類		名称
公開鍵暗号	署名	DSA
		ECDSA
		RSASSA-PKCS1-v1_5
		RSA-PSS
	守秘	RSA-OAEP
		RSAES-PKCS1-v1_5(注 1)
	鍵共有	DH
		ECDH
		PSEC-KEM(注 2)
共通鍵暗号	64 ビットブロック暗号 (注 3)	CIPHERUNICORN-E
		Hierocrypt-L1
		MISTY1
		3-key Triple DES(注 4)
	128 ビットブロック暗号	AES
		Camellia
		CIPHERUNICORN-A
		Hierocrypt-3
		SC2000
	ストリーム暗号	MUGI
		MULTI-S01
		128-bit RC4(注 5)
その他	ハッシュ関数	RIPEMD-160(注 6)
		SHA-1(注 6)
		SHA-256
		SHA-384
		SHA-512
	擬似乱数生成系 (注 7)	PRNG based on SHA-1 in ANSI X9.42-2001 Annex C.1
		PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) Appendix 3.1
		PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) revised Appendix 3.1

注釈:

(注 1) SSL3.0/TLS1.0 で使用実績があることから当面の使用を認める。

(注 2) KEM(Key Encapsulation Mechanism)-DEM(Data Encapsulation Mechanism) 構成における利用を前提とする。

(注 3) 新たな電子政府用システムを構築する場合、より長いブロック長の暗号が使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 4) 3-key Triple DES は、以下の条件を考慮し、当面の使用を認める。
1) FIPS 46-3 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 5) 128-bit RC4 は、SSL3.0/TLS1.0 以上に限定 して利用することを想定している。なお、リストに掲載されている別の暗号が利用できるのであれば、そちらを使用することが望ましい。

(注 6) 新たな電子政府用システムを構築する場合、より長いハッシュ値のものが使用できるのであれば、256 ビット以上のハッシュ関数を選択することが望ましい。ただし、公開鍵暗号での仕様上、利用すべきハッシュ関数が指定されている場合には、この限りではない。

(注 7) 擬似乱数生成系は、その利用特性上、インタオペラビリティを確保する必要性がないため、暗号学的に安全な擬似乱数生成アルゴリズムであれば、どれを利用しても基本的に問題が生じない。したがって、ここに掲載する擬似乱数生成アルゴリズムは「例示」である。

次に電子政府推奨暗号リスト素案作成において行われたいくつかの例外的な措置について説明する。

(F1) SSL/TLS において利用される暗号技術

一般に使われる商用ソフトに入っているか、または入る可能性の高いものが 1 つは選択されなければならない要請 (E2) を考慮して、特定評価対象暗号技術 (B2) で挙げられる SSL/TLS において利用される暗号技術を評価した。その結果、RSA (RSAES-PKCS1-v1_5, RSASSA-PKCS1-v1_5), RC4 (128 ビット鍵) を電子政府推奨暗号リスト素案に加えた。RSASSA-PKCS1-v1_5 は経験的な安全性があり安全と判断して SSL3.0/TLS1.0 以外の利用も推奨する。一方、RSAES-PKCS1-v1_5 及び RC4 (128 ビット鍵) については、無条件に推奨するわけでなく、(注 1) 及び (注 5) を付す。また SSL3.0/TLS1.0 は常に最新の修正プログラムを施すなどして実装に関する攻撃に細心の注意を払うべきである。なお、DES (40, 56 ビット鍵), RC2 (40, 128 ビット鍵), RC4 (40 ビット鍵) は安全ではないと判断した。これらのアルゴリズムについては、SSL/TLS に限らずいかなる場合の利用であっても薦めることは出来ない。

(F2) 電子政府推奨暗号技術において利用される楕円曲線パラメータ

暗号技術評価委員会は、楕円曲線パラメータ選択方法として、既知の攻撃法が成立しないことを確認でき、十分に選択の自由度がある仕様として SECG で示される楕円曲線生成方法を推奨することにした。ECDSA (ANSI X9.62) と ECDSA (SEC1) についてはスキームとしては同一であるが、楕円曲線パラメータの選択等に違いがある。要件 (E3) を配慮し CRYPTREC では ECDSA の仕様として SEC1 に記載されているものを指定した。

(F3) DH の仕様

DH に関しては評価対象の仕様が複数あるので、要件 (E3) を配慮し、電子政府推奨暗号として同一仕様の暗号を確実に調達出来るように ANSI X9.42-2001, “Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography” を DH の仕様として指定した。

(F4) 認証技術

公募はされたが、電子政府推奨暗号リスト素案に暗号技術が掲載されていないカテゴリとして公開鍵暗号技術の認証がある。十分安全であると判断される応募暗号がなかったため、推奨する暗号技術はない。ただし、電子政府システム構築における相手認証に関しては、多くの場合、公開鍵暗号技術等のその他の技術を用いて実現することが出来る。被認証者が特定の情報に署名を施し、検証者がそれを署名検証することで被認証者の同一性が判断できるので、特定の手続きと共に用いれば、電子政府推奨暗号リスト素案に記載されたすべての署名方式は相手認証への応用が基本的に可能である。たとえば、JIS X5056-3:2002 (ISO/IEC 9798-3:1998)[*22]に詳しい手続きの記述がある。

(F5) 擬似乱数生成系の例示

擬似乱数生成系については、応募暗号の中に適切な性能が認められる暗号技術がなかった。しかし、擬似乱数生成系は電子政府システム構築において必要な技術であり、他の暗号技

[*22] 日本規格協会 (`http://www.jsa.or.jp`) から入手可

術とは異なりインターオペラビリティを確保する必要性がないと考えられることから、公開鍵暗号技術の補助関数として利用されるもので、擬似乱数の特性に着目して特に実用上の問題点が指摘されていない 3 つの擬似乱数生成技術を例示として電子政府推奨暗号リスト素案に加えることとした。

例示された 3 つの暗号技術は推奨暗号技術ではなく、それ以外の擬似乱数生成系であっても、技術仕様が公開され安全性の評価が第三者により十分に行われていることを利用者が確認の上利用することは可能である。

1.7 その他の成果

■電子署名法の指針改正 2001 年度の CRYPTREC 評価結果を検討して、電子署名法の指針が改正された。以下に電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針について、平成 14 年総務省 法務省 経済産業省告示第 13 号 (平成 14 年 11 月 21 日官報第 3492 号) の一部改正の改定前と改定後の記述を示し、その理由を説明する。

まず 2001 年 4 月に出された電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針においては以下のように記述されている。

平成 13 年総務省 法務省 経済産業省告示第 2 号 (平成 13 年 4 月 27 日官報号外第 86 号)

(特定認証業務に係る電子署名の基準)
第三条 規則第二条の基準を満たす電子署名の方式は、
次の各号のいずれかとする。
一 RSA 方式 (オブジェクト識別子 1 2 840 113549 1 1 5 又は 1 2 840 113549 1 1 4) であって、モジュラスとなる合成数が 1024 ビット以上のもの
二 ECDSA 方式 (オブジェクト識別子 1 2 840 10045 4 1) であって、楕円曲線の定義体及び位数が 160 ビット以上のもの
三 DSA 方式 (オブジェクト識別子 1 2 840 10040 4 3) であって、モジュラスとなる素数が 1024 ビットのもの
四 ESIGN 方式 (オブジェクト識別子 0 2 440 5 5 3 4 又は 0 2 440 5 5 3 3) であって、モジュラスとなる合成数が 1024 ビット以上、検証に利用されるベキ指数が 8 以上のもの

2002 年 11 月に以下のように改定された。

平成 14 年総務省 法務省 経済産業省告示第 13 号 (平成 14 年 11 月 21 日官報第 3492 号) のよる一部改正後

(特定認証業務に係る電子署名の基準)
第三条 規則第二条の基準を満たす電子署名の方式は、
次の各号のいずれかとする。
一 RSA 方式 (オブジェクト識別子 1 2 840 113549 1 1 5) 又は RSA-PSS 方式 (オブジェクト識別子 1 2 840 113549 1 1 10) であって、モジュラスとなる合成数が 1024 ビット以上のもの
二 ECDSA 方式 (オブジェクト識別子 1 2 840 10045 4 1) であって、楕円曲線の定義体及び位数が 160 ビット以上のもの
三 DSA 方式 (オブジェクト識別子 1 2 840 10040 4 3) であって、モジュラスとなる素数が 1024 ビットのもの

変更点は以下の 3 点である。

(G1) RSA 方式 (オブジェクト識別子 1 2 840 113549 1 1 4) においてはハッシュ関数として MD5 が利用されている。しかし、MD5 は、そのハッシュ値の長さが 128 ビットと十分に大きいわけではなく長期にわたって安全に利用できるとは言えないうえ、すでに MD5 よりも安全なハッシュ関数である SHA-1 が広く普及しているため、RSA 方式 (オブジェクト識別子

1 2 840 113549 1 1 4) の記載が削除された。

(G2) ESIGN 方式には、推奨パラメータの一部の利用に際して、無視できない確率で偽造が可能であり、署名技術として安全ではないため、ESIGN 方式 (オブジェクト識別子 0 2 440 5 5 3 4 又は 0 2 440 5 5 3 3) の記載が削除された。

(G3) RSA-PSS 方式 (オブジェクト識別子 1 2 840 113549 1 1 10) は証明可能安全性を持つ署名技術であり、RSA-PSS 方式 (オブジェクト識別子 1 2 840 113549 1 1 10) の記載が追加された。

次に、電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針 (2003 年 3 月時点) に記載されている署名技術と電子政府推奨暗号リスト素案に掲載されている署名技術の関係について表 1.4 にまとめる。

表 1.4: 電子署名法に係る指針と電子政府推奨暗号リスト素案との対応

電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針第 3 条に記載された電子署名の方式 (平成 14 年総務省 法務省 経済産業省告示第 13 号 (平成 14 年 11 月 21 日) による一部改正後)	電子政府推奨暗号リスト素案に掲載された公開鍵暗号技術の暗号アルゴリズム名 (括弧内はその仕様書名)	対応関係
RSA 方式 (オブジェクト識別子 1 2 840 113549 1 1 5)	RSASSA-PKCS1-v1_5 (PKCS#1 v2.1)	同一 (ただし、ハッシュ関数として SHA-1 を使用した場合)
RSA-PSS 方式 (オブジェクト識別子 1 2 840 113549 1 1 10)	RSA-PSS (PKCS#1 v2.1)	同一
ECDSA 方式 (オブジェクト識別子 1 2 840 10045 4 1)	ECDSA (SEC 1, Version 1.0)	同一 (だたし、楕円曲線パラメータの違いを除く)
DSA 方式 (オブジェクト識別子 1 2 840 10040 4 3)	DSA (ANSI X9.30:1-1997)	同一

■SSL/TLS の調査報告 2001 年度に暗号技術検討会から暗号技術評価委員会に SSL/TLS において利用される暗号技術の評価依頼があった。CRYPTREC で実施した RSA (RSAES-PKCS1-v1_5, RSASSA-PKCS1-v1_5), DES/Triple DES (40, 56, 168 ビット鍵), RC2 (40, 128 ビット鍵), RC4 (40, 128 ビット鍵) の評価について簡単に報告する。

RSASSA-PKCS1-v1_5 は経験的な安全性があると判断し電子政府推奨暗号リスト素案に掲載する。RSAES-PKCS1-v1_5 (法のサイズが 1024 ビット) は当面の使用を認めて電子政府推奨暗号リスト素案に掲載するが、証明可能安全性が示される見通しはなく、実際に能動的攻撃が成立する可能性も無視できないので、SSL3.0/TLS1.0 にける利用においても最新の修正プログラムを施すなどして、実運用環境における攻撃に対する対策を行い、細心の注意を払う必要がある。

鍵長 40 ビットの DES, RC2, RC4、および鍵長 56 ビットの DES は、鍵の全数探索攻撃により現実的な時間で解読可能であるか、その可能性が極めて高くなってきている。このため、これらの暗号アルゴリズムは、電子政府推奨暗号リスト素案に掲載しないことはもとより、使用そのものをできる限り止めるべきであると勧告する。鍵長が 128 ビットの RC2 は鍵の全数探索よりも効率的な攻撃が可能であるうえ、SSL3.0 以上では利用できなくなっているので、電子政府推奨暗号リスト素案には掲載しない。

Triple DES(168 ビット鍵)[*23]に関しては、現時点で実運用における安全性上の大きな問題が

[*23] 電子政府推奨暗号リスト素案では 3-key Triple DES と表記。

指摘されていないことに加え、インターオペラビリティ確保などの要件を総合的に判断した結果、当面の使用を認めて電子政府推奨暗号リスト素案に掲載する。RC4 (128 ビット鍵)[*24]は、SSL3.0/TLS1.0 での利用方法にしたがって使用する場合には安全であることが確認されたため、SSL3.0/TLS1.0 以上に限定して利用することを想定して電子政府推奨暗号リスト素案に掲載する。今後、電子政府推奨暗号リスト素案に掲載された、RC4 (128 ビット鍵) や Triple DES (168 ビット鍵) 以外の暗号技術が選択できるようになれば、それらを利用するほうが望ましい。

CRYPTREC Report 2001 には SSL3.0/TLS1.0 の暗号プロトコルとしての評価として

> SSL/TLS については、暗号方式、セキュリティプロトコルとして、既存の攻撃に耐性のある方式であるといえる。利用・運用の際には、既存バクに対処済みの最新ソフトウェアを用い、適切なパラメータで運用するなど若干の留意が必要であるが、適切な運用を行うことで、実用上、十分な安全性を有していると考えられる。TLS については、機能追加を目的として拡張作業が行われている。これらの拡張作業の結果、新たなセキュリティホールが発生する可能性もあるため、今後とも、TLS の動向に注目し、その安全性について継続的に調査・検討を実施していく必要がある

と総括されている。

詳しい評価結果はそれぞれの暗号技術が記載されている節を参照されたい。SSL/TLS の利用についての注意点は CRYPTREC Report 2001 にまとめられている。SSL/TLS にはいくつかのバージョンがあるが最新版である SSL3.0/TLS1.0 を利用することを推奨する。詳しい調査報告が CRYPTREC Report 2001 に付随しているので、そちらを参照してほしい。`http://www.shiba.tao.go.jp/kenkyu/CRYPTREC/PDF/c01.pdf` 及び `http://www.ipa.go.jp/security/fy13/report/cryptrec/c01.pdf` から入手できる。

■外部評価報告書の公開 公平性と透明性の観点から評価対象暗号の評価結果は公にして安全性評価の信頼性を高めることも重要な任務と考えられる。CRYPTREC における評価活動における外部評価報告書は原則として CRYPTREC のウェブサイト[*25] において公開する。外部評価報告書を公開することで、暗号技術の調達者・利用者から CRYPTREC における評価の信頼性を得て、また暗号研究者の今後の研究に役立てられることを期待する。公開される外部評価報告書の著者の氏名・所属の公表については、著者の意思を尊重して、公表していないものもある。

■技術的な問い合わせ 本評価報告書に対する、技術的意見や問合せは、CRYPTREC 事務局 (問い合わせ先 e-mail: `info@cryptrec.org`) までご連絡していただけると幸いである。

[*24] 電子政府推奨暗号リスト素案では 128-bit RC4 と表記。

[*25] `http://www.shiba.tao.go.jp/kenkyu/CRYPTREC/index.html` 及び `http://www.ipa.go.jp/security/enc/CRYPTREC/index.html`

1.8　謝辞

暗号技術評価委員会、公開鍵暗号評価小委員会、共通鍵暗号評価小委員会は 3 年間にそれぞれ、25 回、37 回、34 回開催され、各委員会の委員の方々には評価作業、報告書作成に粉骨砕身してご尽力いただいた。また、オブザーバの方々も毎回、委員会に参加していただき、貴重なご意見をいただいた。

国内外の多くの暗号研究者には外部評価者として協力していただいた。彼らの専門的知識と暗号理論に対する真摯な気持ちがなければ CRYPTREC 活動は成し得なかっただろう。また素因数分解セミナーを開催し、運営していただいた皆様のご協力も貴重であった。

また公募は暗号技術評価活動においてもっとも重要な側面であり、提案者の方々には厳しいスケジュールの中で多数の提出物を作成して多くの技術を提案していただいた。暗号技術の応募は CRYPTREC 活動において必要不可欠なものであり、活動を国際的にも認められるプロジェクトになることに大きな役割を果たした。

暗号技術説明会、ワークショップ等への参加、コメントの投稿を通して CRYPTREC 活動に参加していただいた方々の参加もプロジェクトの支えとなった。

ここに多くの方々のご協力に対して感謝の意を表したい。

第 2 章

公開鍵暗号技術の評価

この章では公開鍵暗号技術の評価について報告する。まず、公開鍵暗号技術の評価の概要として、評価方針、評価対象技術、評価方法、評価結果の全体像について説明する。次に 2002 年度の詳細評価対象の各公開鍵暗号技術のそれぞれに対して評価結果を報告する。

2.1　評価の概要

2.1.1　評価方針

電子政府で用いる暗号技術に求められる基本的な性質として、パラメータ指定の仕方を含み具体的に規定された暗号が、現時点において安全であり、少なくとも直ちに安全でなくなる危険性が小さく、10 年程度の使用には耐えられるであろうと、広くコンセンサスを得られるものであることがあげられよう。豊富な使用実績があり現時点までに安全性の上で特段の問題点が指摘されていないという経験的な知識もそのようなコンセンサスの形成に役立つ。さらに安全性を評価する上で曖昧な部分をなるべく絞りこむ方法として、証明可能安全性という概念を用いることが有効である。

安全性の評価は、次の方針によって行った。

1. 比較的長い期間にわたる使用実績・評価実績があり、インターオペラビリティの観点から仕様の変更を簡単には求められない公開鍵暗号技術については、経験的な安全性を求めるのみで、必ずしも証明可能安全性が示されていることを要件とはしない。
2. 使用実績が少ない新しい公開鍵暗号技術については、既存暗号技術とは独立に仕様を定めることができることから、少なくとも証明可能安全性が示されていることを必須とする。
3. それに加えて、プリミティブが依存する数論的な問題の困難性や、推奨パラメータの選択方法や補助関数のスキームの中での利用方法を含めて総合的に安全性を評価する。

なお、ここで参照する経験的安全性とは、該当する暗号技術が比較的長い期間にわたる使用実績があり、多くの研究者による研究にも係わらず具体的な攻撃アルゴリズムが指摘されていなく、実運用においても脆弱性が指摘されていないことを指す。しかしながら、そのことが攻撃ア

ルゴリズムおよび脆弱性の非存在性を主張できるものではないことに注意するべきである。

また、ここで参照する証明可能安全性とは、暗号が安全であることが証明されているということを示すものではないことに注意が必要である。本章では、「ある仮定の下での証明可能安全性を有する」という表現を用いて次の状況を示す。すなわち、ある公開鍵暗号が証明可能安全性を有するとは、その暗号またはその暗号の理想化暗号に対して、その暗号で守りたい安全性を脅かす攻撃方法があれば、それを使って、別の数学的問題を低い計算量で解く方法が導けることを、何らかの前提のもとで、厳密に証明できることを指すことにする。ただし、ある暗号の理想化暗号とは、その暗号スキームが用いる補助関数 (ハッシュ関数など) を仮想的なもの (ランダム関数など) に置き換えた以外はもとの暗号と全く同じである仮想的な暗号のことを指す。表現「ある仮定の下での」は、その暗号自身についてであるか仮想暗号についてであるかの違い、数学的問題の種類や計算量的困難性の違い、問題とする安全性の種類の違い、攻撃方法の種類の違い、前提の違い、などがあり、これら次第でその暗号の安全性に対して与えられる信頼感には多様性があることを伝えるために用いている。署名技術に関しては、適応的選択文書攻撃に対して存在的偽造不可、守秘技術には適応的選択暗号文攻撃に対して強秘匿であることを求めた。

証明可能安全性の証明自体が誤りでない限り、ある暗号が証明可能安全性を有すること自体が時間経過によって覆ることはない。しかし、数学的問題の計算量的困難性の見積もりは、理論の進歩や技術環境の変化によって変動するものであるから、ある仮定の下での証明可能安全性を有していて、その仮定が現時点においては満たされていると判断される暗号であっても、安全とはいえない暗号に将来変わることがありえる。さらに、安全性において理想化暗号とのギャップが著しいことが将来判明することもありえる。なお、ある暗号が証明可能安全性を有することが現時点で示されていないことが、その暗号が安全でないことを意味するわけではない。利用実績があり現時点で特段の安全性上の問題点が発見されていないが、安全性を証明可能安全性という形で示すことが現時点の証明技術ではできていないという場合もある。

また、近年研究が活発になっている実装攻撃に関しては、その安全性がアルゴリズムの実装方法に大きく依存するため、実装攻撃への耐性は公開鍵暗号技術のアルゴリズムの評価においては高い優先度を与えなかった。現実に研究動向も活発で今後の研究により攻撃と対策のいずれにおいてもさらなる進展が予想されるため、実運用に際しては最新の実装攻撃の研究動向に十分に注意を払い実装を行う必要がある。たとえ証明可能性が示されている暗号技術であっても不注意な実装を行えば、実運用環境においては実装攻撃により攻撃可能となる可能性もあることに注意すべきである。実装攻撃については 6 章にサーベイが与えられている。

2.1.2 評価対象暗号技術

2002 年度評価対象となった暗号技術は、

1. 応募暗号技術 (詳細評価対象)
   ESIGN[*1], ECDSA(SEC 1), RSA-PSS, RSA-OAEP, HIME(R), ECIES, ECDH(SEC 1),

[*1] 平成 13 年 総務省 法務省 経済産業省 告示第 2 号 (平成 13 年 4 月 27 日 官報 号外第 86 号) の電子署名及び認証業

PSEC-KEM

2. その他評価が必要とされた暗号技術
   RSAES-PKCS1-v1_5, TSH-ESIGN, RSA-OAEP, RSA-PSS, DH
3. 特定評価対象技術
   DSA, ECDSA(ANSI X9.62), RSASSA-PKCS1-v1_5, ESIGN

の 3 つに分類される。ESIGN は応募暗号技術でもあり、特定評価対象でもある。RSA-OAEP, RSA-PSS は応募暗号でもあり、「その他評価が必要」と判断されたものでもある。

2.1.3 評価の実施方法

2002 年度においては詳細評価および関連調査のみを実施した。国内外の暗号理論の専門家にも評価を委託し、その評価結果を含めて公開鍵暗号評価小委員会にて検討し結論をまとめた。暗号技術評価活動の最終年度である 2002 年度は 2001 年度に把握し切れなかった部分を詳細に調査し、また SSL/TLS において使用実績のある RSAES-PKCS1-v1_5 の評価を暗号技術検討会からの要請で「その他評価が必要とされた暗号技術」と判断して評価した。

2.1.3.1 詳細評価

比較的長い期間にわたる使用実績・評価実績がある公開鍵暗号技術については問題点が指摘されていないことを確認した。使用実績が少ない新しい公開鍵暗号技術については証明可能安全性の議論、パラメータの選択方法、補助関数の利用方法で問題点の有無を調査した。安全性評価においては、公開鍵暗号評価小委員会における評価活動だけでなく、国内外の暗号研究者にも評価を委託して、公開鍵暗号評価小委員会で結論をまとめた (表 2.1 を参照)。

■詳細評価項目 各評価対象暗号技術に関して、安全性が依存している数論的問題の困難さとスキームに関して安全性の評価を実施した。特に、以下の点について重点的に評価を行った。

- 数論的問題の困難さに関する安全性評価項目
  - i) 素因数分解問題
    - 既知の解法アルゴリズムの調査とそれらの効率の比較
    - pq 型と $p^d q$ 型 ($d \geq 2$) の比較
    - 数体ふるい法をハードウェア回路上で実現する研究の妥当性と実現可能性
  - ii) 離散対数問題
    - 既知の解法アルゴリズムの調査とそれらの効率の比較
  - iii) 楕円曲線上の離散対数問題
    - 既知の解法アルゴリズムの調査とそれらの効率の比較
    - 限定された曲線 (Koblitz 曲線等) の場合の問題点の調査

務に関する法律に基づく特定認証業務の認定に係る指針に記載されたもの。平成 14 年 総務省 法務省 経済産業省告示第 13 号 (平成 14 年 11 月 21 日 官報 第 3492 号) の一部改正により削除されたところ。

- パラメータ選択とその安全性
  - SEC 1 と ANSI における楕円曲線パラメータの相違と安全性
  - RSA で利用されるパラメータの選択方法
- 各暗号スキームに関する安全性評価項目

  i) DSA
  - プリミティブ及びスキームの安全性評価
  - FIPS186-2 Appendix 3 で与えられている乱数生成法の不備

  ii) ECDSA
  - generic group model における存在的偽造不可の証明可能安全性の妥当性と意義
  - 還元関数に関係する脆弱性と DSKS 特性について
  - Koblitz 曲線の安全性評価

  iii) ESIGN, TSH-ESIGN
  - 推奨パラメータのサイズの妥当性
  - e 乗根近似問題と p^2q 型の素因数分解問題
  - SO-CMA モデルにおける証明可能安全性

  iv) RSA
  - RSASSA-PKCS1-v1_5, RSAES-PKCS1-v1_5 の安全性評価
  - RSA-PSS, RSA-OAEP の証明可能安全性及びその帰着効率

  v) ECIES
  - MAC および KDF 関数に関連する脆弱性の調査

  vi) HIME(R)
  - 証明可能安全性を含む総合的な安全性の検査
  - p^2q 型の素因数分解問題

  vii) DH
  - スキーム (ANSI X9.42-2001) の安全性評価

  viii) ECDH
  - スキーム (SEC 1) の安全性評価

  ix) PSEC-KEM
  - KEM-DEM 構成にもとめられる KEM の証明可能安全性
  - KEM-DEM 構成法による (ハイブリッド型) 公開鍵暗号の安全性
  - KEM 以外の利用方法における安全性

2.1.3.2　ソフトウェア実装評価

暗号技術のパフォーマンスの評価についてはソフトウェア実装評価を実施して、運用上の問題点が存在しないことを確認した。公開鍵暗号技術においては、処理速度になんらかの基準を設けることはしなかった。公開鍵暗号技術についてのソフトウェア実装評価は 2000 年度のみ実施された。2001 年度および 2002 年度は詳細評価対象暗号の多くが 2000 年度に計測済みかあるいは多くの使用実績を有して運用上に問題点がないと考えれるため実装評価は実施しなかった。

表 2.1: 電子政府暗号候補および 2002 年度詳細評価対象暗号技術に対する詳細評価依頼数

評価対象		2000 年度	2001 年度	2002 年度	合計
スキーム	DSA	0(1)	3(2)	-	3(3)
	ECDSA	2(1)	3(1)	0(1)	5(3)
	ESIGN	-	3(1)	-	3(1)
	RSA-OAEP, RSA-PSS	0(1)	2(2)	-	2(3)
	PKCS#1v1.5 署名等	-	-	2(1)	2(1)
	ECIES	2(1)	-	2(0)	4(1)
	HIME(R)	-	-	3(0)	3(0)
	DH	0(1)	-	-	0(1)
	ECDH	2(1)	-	-	2(1)
	PSEC-KEM	-	1(2)	0(2)	1(4)
数論的問題の困難さ	素因数分解問題 (調査)	0(1)	0(1)	0(1)	0(3)
	素因数分解問題 (実験)	-	0(1)	0(1)	0(2)
	特殊な型の素因数分解問題	-	3(1)	-	3(1)
	離散対数問題	0(1)	2(1)	-	2(2)
	楕円曲線上の離散対数問題	-	2(0)	1(1)	3(1)

[海外評価数 (国内評価数)]

2.1.3.3 関連調査

2001 年度には SSL/TLS の調査を実施した。2002 年度には実装攻撃の調査を行った。調査報告が 6 章に与えられている。実装攻撃については今後も研究が進むことが予想されるので、暗号技術を実装する際には最新情報を把握して注意して実装を行う必要がある。

2001 年度に開始した素因数分解問題の計算機実験は、素因数分解アルゴリズムの実装法の研究とともに、計算機環境の構築などの準備を行っている。

2.2 評価結果

2.2.1 評価結果の全体像

2002 年度に評価を行った公開鍵暗号技術は、機能と関連する数論的問題とから表 2.2 のように分類できる。

第 2.1.1 節の方針に従って評価を行い、以下の結論 (1), (2), (3), (4), (5), (6), (7) を得た。表 2.2 中には暗号名にその暗号に対する結論を付記した。

(1) DSA、ECDSA(ANSI X9.62, SEC 1)、RSASSA-PKCS1-v1_5、RSA-PSS、RSA-OAEP、DH、ECDH は経験的に安全であり、電子政府での使用には問題がないと考えられる。さらに RSA-PSS と RSA-OAEP は証明可能安全性を有する。もちろん、使用に際しては適切なパラメータ群を選択することが必要である。

表 2.2: 公開鍵暗号技術の評価結果

	素因数分解問題	離散対数問題	楕円曲線上の離散対数問題
署名	ESIGN(4) TSH-ESIGN(5) RSA-PSS(1) RSASSA-PKCS1-v1_5(1)	DSA(1)	ECDSA(ANSI X9.62, SEC 1)(1)
守秘	HIME(R)(7) RSA-OAEP(1) RSAES-PKCS1-v1_5(3)	-	ECIES(6)
鍵共有	-	DH(1)	ECDH(1) PSEC-KEM(2)

(2) PSEC-KEM の運用は KEM-DEM 構成における利用を前提とする。また楕円曲線パラメータは SEC 1 で規定されるものを利用することを推奨する。

(3) RSAES-PKCS1-v1_5 は、SSL3.0/TLS1.0 での使用実績があることから当面の使用を認める。ただし、経験的安全性を持つが、証明可能安全性が示される見通しはなく、実際に能動的攻撃が成立する可能性も無視できないので、実運用環境における攻撃に対する対策を十分に施し細心の注意を払う必要がある。

(4) ESIGN には、提案者が推奨するパラメータのなかに偽造を可能にするものが含まれていることがわかった。証明可能安全性を持たない。

(5) TSH-ESIGN は ESIGN の参考のために評価された。望まれる証明可能安全性を持たない。

(6) ECIES には、KDF 関数への入力及び MAC の取り扱い方法に問題がある。適応的選択暗号文攻撃が可能であり十分な安全性を持たない。望まれる証明可能安全性を持たない。

(7) HIME(R) は、自己評価書における証明は正しくないと判断された。正確な議論をすることにより証明可能安全性が成立する可能性もあるが、2002 年 9 月の時点では確認されていないと判断された。仕様書における記述に不備が存在する。

2.2.2 個別暗号技術の総評

1. DSA (署名)

   米国 NIST によって提案、標準化された電子署名方式であり、電子署名法に係る指針 に記載されている (オブジェクト識別子 1 2 840 10040 4 3)。NIST ではパラメータ p の大きさを、1024、2048、3072、7680、15360 ビットから選択可能とし、ハッシュ関数についても、SHA-1 だけでなく、SHA-256、SHA-384、SHA-512 が利用可能とする DSA の FIPS ドラフトを 2003 年に公開する予定である。

   安全性は有限体上の離散対数問題の困難性に依存している。証明可能安全性は示されていないが経験的に安全である。安全性の観点からパラメータ p のサイズは 1024 ビットを選択することを強く推奨する。DSA の仕様書で規定される疑似乱数生成方法である FIPS 186-2 Appendix 3 の擬似乱数生成系に問題があるので、2001 年 10 月に NIST が FIPS PUB 186-2 (+ change notice 1) に提示した擬似乱数生成系の修正に従う必要がある。さら

に今後も DSA で利用される擬似乱数生成系の動向に注意すべきである。

2. ECDSA (署名)

ECDSA (ANSI X9.62)(特定評価対象暗号) と ECDSA (SEC 1) (応募暗号) を評価した。ECDSA(ANSI X9.62) は電子署名法に係る指針に記載されている署名方式 (オブジェクト識別子 1 2 840 10045 4 1) である。ECDSA (ANSI X9.62) と ECDSA (SEC 1) はスキームとして同一であり、規定される楕円曲線パラメータについてもほぼ同じである。ECDSA(SEC 1) は、パラメータの値が 160 ビット以上の場合において、電子署名法に係る指針に適合する。

安全性は楕円曲線上の離散対数問題の困難性に依存している。証明可能安全性は示されていないが経験的に安全であり、安全性について重大な問題点は指摘されていない。鍵生成プロセスに十分に留意すれば安全性に問題はないと考えられる。擬似乱数生成に関して FIPS186-2 に記載された手法が規定されているが、ECDSA の原型である DSA において問題点が指摘されており、NIST が FIPS PUB 186-2 (+ change notice 1) に提示した擬似乱数生成系の動向に注意すべきである。

■ECDSA (ANSI X9.62) 電子署名法に係る指針に記載されている署名方式の 1 つである。

■ECDSA (SEC 1) 楕円曲線パラメータ選択法については SEC 1 に示されていて、SEC 2 には推奨される具体的な楕円曲線が与えられている。安全性の観点から群位数が 160 ビット以上の素因子をもつようなパラメータを選択することを強く推奨する。SEC 2 で具体的に示されている楕円曲線については、どの曲線も既知の効率的攻撃法は適用できないことが保証されている。なお、高速処理可能で使用実績があるため SEC 2 に含まれている Koblitz 曲線とよばれる曲線は、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。

3. ESIGN (署名)

ESIGN 署名にはいくつかの仕様が存在する。CRYPTREC では ESIGN(応募暗号) と TSH-ESIGN を評価した。TSH-ESIGN は ESIGN(応募暗号) の評価の参考にするために評価が行われた。電子署名法に係る指針の一部改正 (平成 14 年 総務省 法務省 経済産業省 告示第 13 号 (平成 14 年 11 月 21 日 官報 第 3492 号)) により ESIGN の記載は削除されている。

プリミティブの安全性は、RSA における $n = pq$ 型素因数分解問題の困難性と異なり、$n = p^2q$ 型素因数分解問題と e 乗根近似問題の困難性に依存している。ESIGN の署名生成速度は RSA 署名と比べて高速であるが、RSA プリミティブと同程度の安全性を ESIGN のプリミティブにおいて得るためには RSA のパラメータ (法) サイズよりも少し大きなパラメータ (法) を利用しなければならない。

■ESIGN(応募暗号) 新提案技術に必須とされた証明可能安全性を持たない。実際に、一部のパラメータ (例えば SHA-1 を用いた場合、$|n| = 2048$ かつ $e = 8$ など) の利用に際して、無視できない確率で署名の偽造が可能である。

■TSH-ESIGN TSH-ESIGN が有する証明可能安全性は SO-CMA [*2]に対しての存在的偽造不可にすぎなく、新提案技術に必須とされた証明可能安全性 (CMA[*3]に対して存在的偽造不可) は示されていない。

4. RSA (署名、守秘)

RSA プリミティブを利用した署名にはいくつかの仕様が存在する。CRYPTREC では RSASSA-PKCS1-v1_5(特定評価、その他評価が必要な暗号技術) と RSA-PSS(応募暗号) を評価した。また RSA プリミティブを利用した守秘技術にはいくつかの仕様が存在する。CRYPTREC では RSAES-PKCS1-v1_5(特定評価) と RSA-OAEP(応募暗号) を評価した。RSASSA-PKCS1-v1_5 と RSA-PSS はそれぞれ電子署名法に係る指針に記載されている署名方式 (オブジェクト識別子 1 2 840 113549 1 1 5、オブジェクト識別子 1 2 840 113549 1 1 10) である。ハッシュ関数 MD5 が十分に安全ではないため、MD5 を利用した RSA 方式 (オブジェクト識別子 1 2 840 113549 1 1 4) の記載は電子署名法に係る指針の一部改正 (平成 14 年 総務省 法務省 経済産業省 告示第 13 号 (平成 14 年 11 月 21 日 官報 第 3492 号)) により削除された。

プリミティブの安全性は、$n = pq$ 型素因数分解問題の困難性に依存している。RSA は長期間広く使われている実績、広範な観点から評価が行われてきていて経験的に安全である。安全性の観点から、パラメータ (法)$n = pq$ のサイズは 1024 ビット以上のものを利用することを強く推奨する。

■RSASSA-PKCS1-v1_5 証明可能安全性は示されていないが、経験的に安全である。多くの署名法のエンコーディング手法について署名の偽造が提案されていることにより、本方式で採用されているエンコーディング手法の安全性について監視していくことが必要である。

■RSA-PSS 新提案技術に必須とされた証明可能安全性 (適応的選択文書攻撃に対して存在的偽造不可) がランダムオラクルモデルのもとで RSA 問題の困難性に帰着されるように示されている。安全性が証明されているほかの署名法 (全域ハッシュ法等) に比べて緊密な帰着関係を証明できる特徴を有している。CRYPTREC への提案方式と論文で証明が与えられている方式に若干の相違があるため、対応するパラメータの関係を把握して設計パラメータを選択する必要がある。

■RSAES-PKCS1-v1_5 SSL3.0/TLS1.0 での使用実績があることから当面の使用を認める。ただし、経験的安全性を持つが、証明可能安全性が示される見通しはなく、実際に能動的攻撃が成立する可能性も無視できないので、実運用環境における攻撃に対する対策を十分に施し細心の注意を払う必要がある。

■RSA-OAEP 新提案技術に必須とされた証明可能安全性 (適応的選択暗号文攻撃に対して強秘匿) がランダムオラクルモデルのもとで RSA 問題の困難性に帰着されるように示

[*2] Single-Occurrence Chosen Message Attack の略。1 つのメッセージにつき、署名オラクルへの問い合わせは 1 回しか許されない (1 つのメッセージに対する署名は 1 つしか入手できない) という選択メッセージ攻撃モデルのこと。

[*3] Chosen Message Attack の略。選択メッセージ攻撃モデルのこと。

されている。2000 年に Shoup による指摘を皮きりにして、RSA-OAEP の安全性が議論されたが、安全性の帰着の効率は低下するものの安全であることは示されている。

5. ECIES (守秘)

   ECIES にはいくつかの仕様が存在する。CRYPTREC では応募書類における、SEC 1 に記載された仕様を基に評価した。これは証明可能安全性を有することが確認されている Abdalla, Bellare, Rogaway による ECIES の仕様とは異なるものである。

   安全性は楕円曲線上の離散対数問題の困難性に依存している。応募暗号技術である ECIES におけるスキームの仕様には KDF 関数への入力及び MAC の取り扱い方法の不備が原因で、脆弱性が存在し、新提案技術に必須とされた証明可能安全性 (適応的選択暗号文攻撃に対して強秘匿) を持たない。

6. HIME(R) (守秘)

   2000 年度応募された HIME1 と HIME2 の改良版として 2001 年度に応募された技術である。

   プリミティブの安全性は、RSA における $n = pq$ 型素因数分解問題の困難性と異なり、$n = p^2q$ 型素因数分解問題の困難性に依存している。RSA プリミティブと同程度の安全性を HIME(R) のプリミティブにおいて得るためには RSA のパラメータ (法) サイズよりもすこし大きなパラメータ (法) を利用しなければならない。HIME(R) にはその仕様書に不備・曖昧さがあり、2002 年 9 月時点で信頼に足る HIME(R) の仕様書が公に手に入る状況になっていないと判断された。このため、HIME(R) の第三者による実装性や相互接続性が担保されない。HIME(R) の仕様の曖昧さを埋めてその仕様を合理的に定めたとしても、自己評価書に記載されている証明可能安全性の証明中いくつかの箇所に問題があり不完全である。ある外部評価者が示した証明可能安全性の証明の候補があるが、その妥当性が多くの研究者により精査されていない。2002 年 9 月時点で、新提案技術に必須とされた証明可能安全性 (適応的選択暗号文攻撃に対して強秘匿) を有すると断定できないと判断された。

7. ECDH (鍵共有)

   ECDH は、2000 年度には ECDHS として CRYPTREC に応募されていたが、2001 年度では暗号技術名を ECDH に変更して応募されている。

   安全性は楕円曲線上の離散対数問題の困難性に依存している。CRYPTREC では SEC 1 に記載された仕様を基に評価した。証明可能安全性は示されていないが経験的に安全である。基本的スキームに関して、受動的攻撃 (鍵共有のために通信されるデータに攻撃者が影響を与えることがない場合) に対しては問題点は指摘されていないが、能動的攻撃 (鍵共有のために通信されるデータに攻撃者が影響を与える可能性がある場合) に対して、以下の 2 点に注意を払う必要がある。(1) 公開鍵と Entity との結びつきを保証する手段を確保する。(2) (更新を前提とする) セッション鍵共有方式として使用する場合は、交換する公開鍵は一時的なものとする。楕円曲線パラメータ選択法については SEC 1 に示されていて、SEC 2 には推奨される具体的な楕円曲線が与えられている。安全性の観点から群位数が 160 ビット以上の素因子をもつようなパラメータを選択することを強く推奨する。SEC 2 で具体的に示されている楕円曲線については、どの曲線も既知の効率的攻撃法は適用できないことが保証されている。なお、高速処理可能で使用実績があるため SEC 2 に

含まれている Koblitz 曲線とよばれる曲線は、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。

8. DH (鍵共有)

   DH には、プロトコルにいくつかの仕様が存在する (参考:実使用されているプロトコルの例:RFC2631、ISO/IEC 11770-3、Oakley、PGP)。CRYPTREC では ANSI X9.42-2001 を評価対象の仕様とした。

   安全性は有限体上の離散対数問題の困難性に依存している。証明可能安全性は示されていないが経験的に安全である。基本的スキームに関して、受動的攻撃 (鍵共有のために通信されるデータに攻撃者が影響を与えることがない場合) に対しては問題点は指摘されていないが、能動的攻撃 (鍵共有のために通信されるデータに攻撃者が影響を与える可能性がある場合) に対して、以下の 2 点に注意を払う必要がある。(1) 公開鍵と Entity との結びつきを保証する手段を確保する。(2) (更新を前提とする) セッション鍵共有方式として使用する場合は、交換する公開鍵は一時的なものとする。安全性の観点からパラメータ p のサイズは 1024 ビットを選択することを強く推奨する。

9. PSEC-KEM (鍵共有)

   2000 年度に応募された PSEC を ISO/IEC 18033-2 において審議されている KEM 技術に適合するように変更を加えたもので、2001 年度に応募された。

   KEM 技術に関する証明可能安全性がランダムオラクルモデルのもとで楕円曲線 DH 計算問題に帰着されるように示されている。したがって、KEM(Key Encapsulation Mechanism)-DEM(Data Encapsulation Mechanism) 構成に利用することは安全であると判断した。しかし、それ以外の目的の利用について安全性の研究は十分ではないので、今後の研究に注意すべきである。楕円曲線ドメインパラメータの選択方法については仕様書に明確な記載がない。CRYPTREC としては SEC 1 で規定される曲線の利用を推奨する。SEC 2 には SEC 1 に従って作成された具体的な楕円曲線が与えられている。安全性の観点から群位数が 160 ビット以上の素因子をもつようなパラメータを選択することを強く推奨する。SEC 2 で具体的に示されている楕円曲線については、どの曲線も既知の効率的攻撃法は適用できないことが保証されている。なお、高速処理可能で使用実績があるため SEC 2 に含まれている Koblitz 曲線とよばれる曲線は、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。

2.2.3 数論的問題の困難さに関する総評

2.2.3.1 素因数分解問題

合成数 n の素因数分解問題に関しては、2002 年時点で、$n = pq$ については $|p| = |q|$ かつ $|n| \geq 1024$ で、$n = p^2q$ については $|p| = |q|$ かつ $|n| \geq 1024$ で、それぞれ安全と考えられる。ハードウェア上で数体ふるい法を実現する研究があるが、現実の脅威にはなっていない。素因数分解問題の安全性についての今後の見通しについては、2.4.1.3 節に詳述してある。

2.2.3.2 離散対数問題

素体 $\mathbb{F}_p$ の部分群 (位数 q) の離散対数問題に関しては、2002 年時点で、$|p| \geq 1024$ かつ $|q| \geq 160$ で安全と考えられる。離散対数問題の安全性についての今後の見通しについては、2.4.2.3 節に詳述してある。

2.2.3.3 楕円曲線上の離散対数問題

楕円曲線上の離散対数問題に関しては、2002 年時点で、例外的な楕円曲線を除けば、群位数 (より正確には、ベースポイントの位数) が 160 ビット以上の素因子をもてば安全と考えられる。楕円曲線上の離散対数問題の安全性についての今後の見通しについては、2.4.3.4 節に詳述してある。

2.2.4 電子政府推奨暗号リスト素案の作成

2002 年度の CRYPTREC 活動においては暗号技術の技術的な評価の結果として電子政府推奨暗号を決定した。暗号技術検討会からの 1.5 節の (E2)「一般に使われる商用ソフトに予め入っているか、入る可能性の高いものが各カテゴリーに 1 つは選択される」と (E3)「電子政府推奨暗号に選定されたものと同一仕様の暗号を確実に調達出来るようにするという観点から、各電子政府推奨暗号の仕様を確認する」という要請を考慮して、既存のシステムへの影響やインターオペラビリティといったユーザーの利益も考慮して意思決定を行った。

(1) 仕様の一意化

(E3) を考慮して、同一のスキームを利用した暗号技術についていくつかの仕様書が存在する場合に電子政府システム構築の際に利用するべき暗号技術の仕様書を特定するために仕様の一意化を行った。

1 ECDSA(SEC 1) 及び ECDSA(ANSI X9.62) はスキームとしては同一であるが楕円曲線の選択方法等に差がある。CRYPTREC においては楕円曲線パラメータの選択法について SEC 1 を利用することに決定し、ECDSA の仕様書は SEC 1 のものを利用することに決定した。楕円曲線パラメータの選択において議論されたことは、既知の攻撃法への対処が十分に考慮されているかどうかである。SEC 1 において選択できる楕円曲線パラメータ群よりも ANSI で選択できるパラメータ群のほうが自由度が大きいが、SEC 1 における選択方法はランダムに作成する方法も含まれていて、十分に広範な曲線を選択できる。

2 DH の評価においては Diffie-Hellman の論文で規定される暗号スキームを 2000 年度、2001 年度では評価した。DH に関連する仕様は多様である。2002 年度は DH の仕様として ANSI X9.42-2001 を評価した。DH は応募暗号ではないので、2000 年度、2001 年度に特別に指定した仕様が存在しなかった。

(2) SSL/TLS において利用される暗号技術

(E2) を考慮して、SSL/TLS の利用が電子政府において不可欠であるという判断から、選定されたものが RSAES-PKCS1-v1_5 である。SSL/TLS における利用実績があるので、条件付

きで、当面の使用は認める。

表 2.3 および表 2.4 は、電子政府推奨暗号リストに記載予定の暗号技術に関する安全性評価結果を一覧にしたものである。

利用形態			署名			
暗号アルゴリズム名			**DSA**	**ECDSA**	**RSASSA-PKCS1-v1_5**	**RSA-PSS**
参照すべき仕様書			ANSI X9.30:1-1997 (http://www.x9.org/)	SEC1 (Version 1.0) (http://www.secg.org/)	PKCS#1v2.1 (http://www.rsasecurity.com/rsalabs/pkcs/)	PKCS#1v2.1 (http://www.rsasecurity.com/rsalabs/pkcs/)
リストに載せた根拠	経験的安全性		○	○	○	○
	証明可能安全性の有無		－	－	－	○
		仮定するモデル	－	－	－	ランダムオラクルモデル
		帰着される問題	－	－	－	RSA問題の困難性
		安全性の到達度	－	－	－	適応的選択文書攻撃に対して存在的偽造不可
プリミティブの安全性の根拠			有限体上の離散対数問題	楕円曲線上の離散対数問題	素因数分解問題	素因数分解問題
主なパラメータ・補助関数に関する要件	パラメータの範囲		・「参照すべき仕様書」に記載の各暗号技術の仕様書で指定されたパラメータのうち、CRYPTREC Report 2002で指定された条件を満たすものを使用すること			
	ハッシュ関数		・電子政府推奨暗号リストに記載されたものを使用すること			
	擬似乱数生成器		・電子政府推奨暗号リストの注釈7を参照のこと			
国際標準等への採用状況 (2003年1-3月期)	仕様策定	電子署名法に係る指針	○	○	○	○
		ANSI(注1)	・X9.30:1-1997	・X9.62-1998	－	－
		IEEE(注2)	・P1363-2000	・P1363-2000	・P1363a(Draft)	・P1363a(Draft)
		ISO/IEC(注3)	・14888-3	・14888-3 ・15946-2	－	－
		NESSIE(注4)	－	○	－	○
		NIST(注5)	・FIPS PUB 186-2(+Change Notice1)	・FIPS PUB 186-2(+Change Notice1)	－	－
	利用実績	IETF(注6)	・RFC2246(Proposed Standard) ・RFC3275(Draft Standard) ・RFC3279(Proposed Standard) ・RFC3370(Proposed Standard)	・RFC3278(Informational) ・RFC3279(Proposed Standard) ・ipsec(Internet-Draft) ・tls(Internet-Draft)	・RFC2246(Proposed Standard) ・RFC3275(Draft Standard) ・RFC3279(Proposed Standard) ・RFC3370(Proposed Standard) ・RFC3447(Informational)	・RFC3447(Informational) ・pkix(Internet-Draft) ・smime(Internet-Draft)
		SET(注7)	－	－	○	－
		SSL3.0(注8)	○	－	○	－
		WAP/WTLS(注9)	－	○	○	－
		W3C(注10)	・XML-Signature Syntax and Processing(12 February 2002)	－	・XML-Signature Syntax and Processing(12 February 2002)	－
特記事項			・より大きなサイズのパラメータを選択可能とするために仕様の変更が検討されていることに注意を払う必要がある。	・電子署名法の指針にも(一部のパラメータを除いて)適合している。 ・Koblitz曲線と呼ばれる曲線は、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。	－	－

注1: Americal National Standards Institute (http://www.ansi.org)
注2: Institute of Electrical and Electronics Engineers, Inc. (http://www.ieee.org)
注3: International Organization for Standardization (http://www.iso.org)
International Electrotechnical Commission (http://www.iec.ch)
注4: New European Schemes for Signatures, Integrity, and Encryption (http://www.cryptonessie.org)
注5: National Institute of Standards and Technology (http://www.nist.gov)
注6: Internet Engineering Task Force (http://www.ietf.org)
注7: Secure Electronic Transaction (http://www.setco.org)
注8: Secure Sockets Layer protocol (http://wp.netscape.com/eng/ssl3)
注9: Wireless Application Protocol (http://www.wapforum.org)
注10: World Wide Web Consortium (http://www.w3.org)

表 2.3: 電子政府推奨暗号リスト素案に記載された署名方式に関する安全性のまとめ

利用形態			守秘		鍵共有		
暗号アルゴリズム名			**RSA-OAEP**	**RSAES-PKCS1-v1_5**	**DH**	**ECDH**	**PSEC-KEM**
参照すべき仕様書			PKCS#1v2.1 (http://www.rsasecurity.com/rsalabs/pkcs/)	PKCS#1v2.1 (http://www.rsasecurity.com/rsalabs/pkcs/)	ANSI X9.42-2001 (http://www.x9.org/)	SEC1 (Version 1.0) (http://www.secg.org/)	PSEC-KEM仕様書(2002 年5 月14 日) (http://info.isl.ntt.co.jp/psec/CRYPTREC/index-j.html)
リストに載せた根拠	経験的安全性		○	○	○	○	-
	証明可能安全性の有無		○	-	-	-	○
		仮定するモデル	ランダムオラクルモデル	-	-	-	ランダムオラクルモデル
		帰着される問題	RSA問題の困難性	-	-	-	楕円曲線上のDH計算問題
		安全性の到達度	適応的選択暗号文攻撃に対して強秘匿	-	-	-	鍵カプセル化メカニズムとして適応的選択暗号文攻撃に対して強秘匿
プリミティブの安全性の根拠			素因数分解問題	素因数分解問題	有限体上の離散対数問題	楕円曲線上の離散対数問題	楕円曲線上の離散対数問題
主なパラメータ・補助関数に関する要件	パラメータの範囲		・「参照すべき仕様書」に記載の各暗号技術の仕様書で指定されたパラメータのうち、CRYPTREC Report 2002で指定された条件を満たすものを使用すること				
	ハッシュ関数		・電子政府推奨暗号リストに記載されたものを使用すること	-	・電子政府推奨暗号リストに記載されたものを使用すること		
	擬似乱数生成器		・電子政府推奨暗号リストの注釈7を参照のこと				
国際標準等への採用状況 (2003年1-3月期)	仕様策定	電子署名法に係る指針	-	-	-	-	-
		ANSI(注1)	-	・X9.44(Draft)	・X9.42-2001	・X9.63-2001	-
		IEEE(注2)	・P1363-2000	-	・P1363-2000	・P1363-2000	
		ISO/IEC(注3)	-	-	・11770-3	・11770-3 ・15946-3	・18033-2(Committee Draft)
		NESSIE(注4)	-	-	-	-	○ (Public-key Encryptionとして)
		NIST(注5)	-	-	-	-	-
	利用実績	IETF(注6)	・RFC3447(Informational) ・pkix(Internet-Draft) ・smime(Internet-Draft)	・RFC2246(Proposed Standard) ・RFC3370(Proposed Standard) ・RFC3447(Informational)	・RFC2246(Proposed Standard) ・RFC2409(Proposed Standard) ・RFC2631(Proposed Standard) ・RFC3370(Proposed Standard) ・RFC3279(Proposed Standard)	・RFC3278(Informational) ・RFC3279(Proposed Standard) ・ipsec(Internet-Draft) ・tls(Internet-Draft)	-
		SET(注7)	-	-	-	-	-
		SSL3.0(注8)	-	○	○	-	-
		WAP/WTLS(注9)	-	○	○	○	-
		W3C(注10)	・XML Encryption Syntax and Processing(10 December 2002)	・XML Encryption Syntax and Processing(10 December 2002)	・XML Encryption Syntax and Processing(10 December 2002)	-	-
特記事項			-	・SSL3.0/TLS1.0で使用実績があり当面の使用を認める。	・鍵とエンティティとの結び付きを保証する手段を備え、またセッション鍵として使用する場合、交換する公開鍵は一時的なものとするべきである。	・鍵とエンティティとの結び付きを保証する手段を備え、またセッション鍵として使用する場合、交換する公開鍵は一時的なものとするべきである。 ・Koblitz曲線と呼ばれる曲線は、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。	・KEM(Key Encapsulation Mechanism)-DEM(Data Encapsulation Mechanism)構成における利用を前提とする。 ・Koblitz曲線と呼ばれる曲線は、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。

注1: Americal National Standards Institute (http://www.ansi.org)
注2: Institute of Electrical and Electronics Engineers, Inc. (http://www.ieee.org)
注3: International Organization for Standardization (http://www.iso.org)
International Electrotechnical Commission (http://www.iec.ch)
注4: New European Schemes for Signatures, Integrity, and Encryption (http://www.cryptonessie.org)
注5: National Institute of Standards and Technology (http://www.nist.gov)
注6: Internet Engineering Task Force (http://www.ietf.org)
注7: Secure Electronic Transaction (http://www.setco.org)
注8: Secure Sockets Layer protocol (http://wp.netscape.com/eng/ssl3)
注9: Wireless Application Protocol (http://www.wapforum.org)
注10: World Wide Web Consortium (http://www.w3.org)

表 2.4: 電子政府推奨暗号リスト素案に記載された守秘・鍵共有に関する安全性のまとめ

2.3　個別暗号技術の評価

2.3.1　DSA

2.3.1.1　評価対象暗号技術

- ANSI X9.30 Public Key Cryptography for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA),
  American National Standard for Financial Services, 1997.

2.3.1.2　技術概要

DSA(Digital Signature Algorithm) は、米国 NIST(National Institute of Standards and Technology) によって提案・標準化された署名方式である [1, 2]。また、DSA は電子署名法に係る指針で記載された署名方式の一つでもある。

DSA では有限体上の離散対数問題が困難であることを安全性の根拠としている。

2.3.1.3　技術仕様

鍵生成処理

DSA における鍵生成処理は以下の手順で行う。

1. $2^{511+64j} < p < 2^{512+64j}$ ($j \in \{0, 1, \ldots, 8\}$) を満たす素数 p を選ぶ。
2. $p-1$ を割り切る 160 ビットの素数 q ($2^{159} < q < 2^{160}$) を選ぶ。
3. $g = h^{(p-1)/q} \bmod p$ となる g を算出する。ただし、h は $1 < h < p-1$ を満たす整数。
4. $0 < x < q$ を満たす乱数 x を生成する。
5. $y = g^x \bmod p$ となる y を算出する。

上記手順によって生成された (p, q, g, y) が公開鍵 (public key)、x が秘密鍵 (private key) となる。

署名生成処理

DSA において、平文 M に対する署名生成処理は以下の手順で行う。

1. $0 < k < q$ を満たす乱数 k を生成する。
2. $r = (g^k \bmod p) \bmod q$ となる r を算出する。
3. $s = (k^{-1}(\text{SHA-1}(M) + xr)) \bmod p$ となる s を算出する。ただし、SHA-1(M) は、平文 M を FIPS 180-1 で規定された Secure Hash Algorithm によって変換した結果。

上記手順によって生成された (M, r, s) が署名文となる。

署名検証処理

DSA において、署名文 (M', r', s') に対する署名検証処理は以下の手順で行う。

1. $w = (s')^{-1} \bmod q$ となる w を算出する。
2. $u1 = ((\text{SHA-1}(M'))w) \bmod q$ となる $u1$ を算出する。
3. $u2 = ((r')w) \bmod q$ となる $u2$ を算出する。
4. $v = (((g)^{u1}(y)^{u2} \bmod p) \bmod q$ となる v を算出する。
5. $v = r'$ であるかどうかを確認する。

上記手順 5 で、v と r' とが等しい場合にのみ、受け取った署名文が正しいものと判断する。

2.3.1.4 安全性評価

FIPS 186-2 Appendix 3 の乱数生成について

FIPS 186-2 の Appendix 3 では、$(0, 2^{160})$ の出力域を持つ擬似乱数生成器 G を用いてパラメータ k を以下のように生成することとしている。

$$k = G(t, \text{KVAL}) \bmod q \qquad \text{ただし、} t, \text{KVAL} \text{ は乱数種}$$

本来、パラメータ k は、$(0, q-1)$ の範囲で同じ確率で生成されるべきであるが、上記方式では、mod q の折り返し効果によって、k が $(0, 2^{160} - q - 1)$ に入る確率が $(2^{160} - q, q - 1)$ に入る確率の 2 倍になってしまう。

D. Bleichenbacher は、このような問題点を利用した攻撃法を指摘している [3]。Bleichenbacher の攻撃は、パラメータ k のバイアスを利用するものであるが、本人はその詳細を広く公にはしていない。ある評価者は、外部評価レポートにおいて、Bleichenbacher の攻撃法を以下のように説明している。

パラメータ k のバイアスは、

$$bias(k) = E(e^{\frac{2i\pi k}{q}})$$

で示されるが、上記生成確率の偏りを考慮すると、

$$\begin{aligned}
bias(k) &= \sum_{r=0}^{N-1} \frac{1}{N} e^{\frac{2i\pi(r \bmod q)}{q}} \\
&= \sum_{r=0}^{N-1} \frac{1}{N} e^{\frac{2i\pi r}{q}} \\
&= \frac{1}{N} \times \frac{e^{\frac{2i\pi N}{q}} - 1}{e^{\frac{2i\pi}{q}} - 1} \\
&= \frac{e^{i\pi\frac{N-1}{q}}}{N} \times \frac{\sin(\frac{\pi N}{q})}{\sin(\frac{\pi}{q})} \\
&\approx \frac{q e^{i\pi\frac{N-1}{q}}}{\pi N} \times \sin(\frac{\pi N}{q})
\end{aligned}$$

となる ($N = 2^{160}$)。$q \approx N$ であるため、$bias(k)$ は、$\frac{\pi N}{q}$ に大きく依存することになる。この性質を利用すれば秘密鍵 x が導出できる可能性がある。

Bleichenbacher の指摘を受け、2001 年 10 月に、NIST は、FIPS 186-2 に Change Notice を追加して乱数生成の手順を以下のように修正した。修正された手順によって生成した乱数を使用すれば上記問題点を回避できることから、FIPS186-2 change notice に示された手順に従うことが望ましい。

$$z_1 = G(t, \mathrm{KVAL}_1) \bmod q$$
$$z_2 = G(t, \mathrm{KVAL}_2) \bmod q$$
$$k = (z_1 \times 2^{160} + z_2) \bmod q$$

また、上記擬似乱数生成器 G に関して、FIPS 186-2 の Appendix 3 では SHA-1 と DES を利用した方式がそれぞれ規定されているが、DES を利用した方式には特殊な性質があるので SHA-1 を利用したほうがよいという報告もある。ただし、DES を利用した場合においても指摘されている性質が DSA の安全性に影響を及ぼすことはないものと考えられる。

パラメータの選択について

上記技術仕様にも記載したように、DSA の元々の仕様では、パラメータ p の大きさを 512 ビットから 1024 ビットまで 64 ビット単位で選択可能である。これに対し、電子署名法に係る指針や、FIPS 186-2 Change Notice では、パラメータ p の大きさを 1024 ビットに限定している。現時点で安全なパラメータサイズがどれくらいであるかということに関しては様々な意見があり、正確な値を示すことは困難であるが、現在の計算機の能力でも 512 ビット程度では安全性を保つことが難しいというのは、ある程度一致した見解だと考えられる。したがって、我々は、パラメータ p の大きさとして、現在の仕様において最も安全なパラメータサイズ、すなわち 1024 ビットを選択することを強く推奨する。

一方、NIST は、パラメータの大きさや利用可能なハッシュ関数などについてすでに仕様の改定を検討している [4]。具体的には、パラメータ p の大きさは、1024、2048、3072、7680、15360 ビットから選択可能となる。また、ハッシュ関数についても、SHA-1 だけでなく、SHA-256、SHA-384、および SHA-512 を利用可能となる予定である[*4]。

このような仕様変更も含めた今後の世の中の動向にも十分注意を払うことが必要だと考える。なお、安全なパラメータサイズの問題に関しては、「2.4.2 離散対数問題」を参照されたい。

また、署名生成時に使用する乱数 k に関して、複数の平文を同じ乱数 k を用いて署名すると秘密鍵 x が算出されてしまう。例えば、2 つの平文 M_1, M_2 に対する署名文をそれぞれ $(r, s_1), (r, s_2)$ とするとき、次式より秘密鍵 x を算出することができる。したがって、署名生成毎に異なる乱数 k を選択することが必要である。

$$x = -\frac{s_2 \mathrm{SHA\text{-}1}(M_1) - s_1 \mathrm{SHA\text{-}1}(M_2)}{r(s_2 - s_1)} \bmod q$$

[*4] NIST と CRYPTREC との私的な情報交換に基づく。それによると、日程的には 2003 年 5 月くらいに public review を行った後、2003 年末あるいは 2004 年初頭に正式な FIPS(FIPS186-3) となるとのことである。具体的には、`http://csrc.nist.gov/CryptoToolkit/tkdigsigs.html` を参照のこと

さらに、乱数 k に関して、その一部のビットがわかってしまうと秘密鍵が算出されてしまう恐れがあるという報告がある。文献 [5] の攻撃法は、lattice reduction technique に基づいたものであり、70 個の署名文 (パラメータ p の大きさは 512 ビット) が与えられた場合、乱数 k の中の 5 ビットがわかれば 100% の確率で、4 ビットがわかれば 90% の確率で秘密鍵を算出できるという実験結果が示されている。文献 [5] の著者らはより少ないビットでの攻撃可能性についても指摘しており、今後の研究の進展に注意を払うことが必要である。

上記以外にも、いくつかの特殊なパラメータ (例えば、$g = 0, g = y^{\alpha} \bmod p$) における攻撃法が報告されており、DSA の使用に際してはパラメータの選択に留意する必要がある。

証明可能安全性について

DSA に若干の変更を加えた場合には、ランダムオラクルモデルで、離散対数問題の困難性と同等の証明可能安全性を示すことができるが、DSA 自体に関しては、何らかの妥当なモデルや仮定のもとでの証明可能安全性は、現在までのところ報告されていない。

しかしながら、これまで広く使用されてきているという実績をも考慮すれば、現時点で安全性に大きな影響を与えるような問題点があるとは思われない。

参考文献

[1] ANSI X9.30 Public Key Cryptography for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA), American National Standard for Financial Services, 1997.

[2] FIPS PUB(Federal Information Processing Standards publication) 186-2: DIGITAL SIGNATURE STANDARD (DSS), U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology, 2000.

[3] Lucent Technologies, Press releases: Scientist discovers significant flaw that would have threatened the integrity of on-line transactions,
Available at `http://www.lucent.com/press/0201/010205.bla.html`

[4] NIST Second Key Management Workshop: Key Management Guideline (Draft),
Available at `http://csrc.nist.gov/CryptoToolkit/tkkeymgmt.html`

[5] P. Q. Nguyen and I. E. Shparlinski, The insecurity of the Digital Signature Algorithm with partially known nonces, Journal of cryptology, Volume 15 - Number 3, 2002 .

2.3.2 ECDSA

2.3.2.1 評価対象暗号技術

- ECDSA in SEC1
  Standards for Efficient Cryptography,
  “SEC1:Elliptic Curve Cryptography”,
  Certicom Research, Ver.1.0, September 2000,
  http://www.secg.org/secg_docs.htm
- ECDSA(ANSI X9.62)
  ANSI X9.62-1998,
  “Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digial Signature Algorithm (ECDSA)”,
  American National Standard for Financial Services, 1998.

2.3.2.2 技術概要

ECDSA は楕円曲線上の離散対数問題に基づく署名方式である。

ECDSA には複数の仕様があるが、CRYPTREC では 2000 年度暗号技術評価委員会に応募された ECDSA in SEC1[3] を主に評価した。ECDSA in SEC1 は、楕円曲線暗号の標準仕様策定のためのコンソーシアムである SECG(Standards for Efficient Cryptography Group) により、SEC1:Elliptic Curve Cryptography(Version 1.0)[3] として、2000 年に策定された仕様である。この他の仕様として、電子署名法に係る指針に記載されている ANSI X9.62-1998[1] がある。

2.3.2.3 技術仕様

ECDSA による署名手順・検証手順の概要

ECDSA による署名生成、署名検証は以下のように行われる。

楕円曲線パラメータを T とする。T には位数が素数 n であるベースポイント G が含まれる。

鍵生成: 秘密鍵 $d \in [0, n-1]$ をランダムに選択し、$Q = dG$ を公開鍵とする。

署名生成: 平文 M に対する署名生成を以下の手順で行う。

1 ランダムな整数 $k \in [0, n-1]$ を選択。
2 $R = kG = (x_1, y_1)$ を計算。
3 $r = x_1 \bmod n$ を計算。
4 $e = h(M)$ を計算。ここで h はハッシュ関数 SHA-1 である。
5 $s = k^{-1}(e + dr) \bmod n$ を計算。
6 (r, s) を平文 M の署名とする。

署名検証: 平文 M、署名 (r, s)、公開鍵 Q に対して以下の検証を行う。

1 $R' = (x_2, y_2) = (s^{-1}h(M))G + (s^{-1}r)Q$ を計算。
2 $x_2 \bmod n = r$ が成り立つことを検証。

ECDSA in SEC1 と ANSI X9.62 の差異

ECDSA in SEC1 と ANSI X9.62 の間で、ECDSA の署名スキームは細かな部分を除いて同一である。2 つの仕様の差は主に以下の点にある。

- 楕円曲線パラメータの選択範囲
- 署名におけるメッセージのハッシュ処理
- 擬似乱数生成器

楕円曲線パラメータに関して、ECDSA in SEC1 ではその選択範囲に制約がある。定義体が $\mathbb{F}_p$ と $\mathbb{F}_{2^m}$ のそれぞれに対し、定義体のビットサイズが 8 種類づつに限定されている。この中には定義体サイズが 160 ビット未満のものが 2 種類づつ含まれている。さらに、推奨される具体的な楕円曲線パラメータが SEC2 ドキュメントに挙げられている。SEC2 ドキュメントには、標数 p と標数 2 の楕円曲線に関し、複数の定義体サイズのもとで、検証可能な形式でランダムに選択された曲線と Koblitz 曲線とが推奨されている。検証可能な形式でランダムに曲線を選択する手法は ANSI X9.62 に記載されたものである。一方、ANSI X9.62 では、ベースポイントの位数 n に $n > 2^{160}$ と制約がある以外は定義体のサイズに制約はない。また、ANSI X9.62 では、付録にサンプルとして楕円曲線パラメータが挙げられているのみで、推奨される具体的な曲線パラメータはない。ただし、楕円曲線パラメータの選択手順が付録に示されており、検証可能な形式でランダムに選択する手法と、その他の手法 (Weil 法と CM 法の名前が挙げられている) が記載されている。

署名におけるメッセージのハッシュ処理に関しては、ECDSA in SEC1 ではベースポイントの位数 n が $\lceil \log_2 n \rceil < 160$ となるような楕円曲線パラメータの場合、$h(M)$ の最左 $\lceil \log_2 n \rceil$ ビットを利用すると規定されている。これ以外の場合には、ECDSA in SEC1 と ANSI X9.62 のハッシュ処理は同一である。

擬似乱数生成器に関しては、ECDSA in SEC1 には具体的なアルゴリズムの記載がない。一方、ANSI X9.62 では、FIPS186-2 に記載された手法が規定されている。

2.3.2.4 安全性評価

プリミティブの安全性

ECDSA のプリミティブは楕円曲線上の離散対数問題に安全性の根拠を置いている。この問題の評価結果は「第 2.4.3 節 楕円曲線上の離散対数問題」を参照されたい。

なお、ECDSA in SEC1[3] で指定されているパラメータには、定義体のサイズが 160 ビット未満のものが含まれているが、安全性の観点から 160 ビット以上のパラメータを使用すべきである。

証明可能安全性

ECDSA ではランダムオラクルモデルでの証明可能安全性は確認されていない。一方、ランダムオラクルモデルとは異なる generic group model(generic model とも呼ばれるが、ここでは generic group model で統一する) での安全性証明が Brown によって与えられている [2]。generic group model とは群要素の表現がランダムに与えられると仮定した仮想的なモデルである。すなわち、加法群 $\mathbb{Z}_n$ から群要素の表現を定めるビット列集合 $S \in \{0,1\}^*$ への全単射 σ をランダムに定める generic group オラクルを仮定し、群要素の演算は generic group オラクルへの問い合わせにより実行するものとしたモデルである。Brown はこのモデルの下で、ハッシュ関数の衝突困難性を前提とすると ECDSA は適応的選択平文攻撃に対して存在的偽造不可であると主張している。Brown の定理に対して別の証明を与えた結果が文献 [5] に示されている。

しかしながら、generic group model での安全性証明が ECDSA の安全性に関して意義を与えるかどうかについては否定的な見方が強い [5]。これは、ECDSA には 'malleable' である性質が発見されているのに対し、generic group model の下ではハッシュ関数の衝突困難性を前提とすれば 'non-malleability' が証明できるという一見矛盾した状況が生じるためである。ここでの 'malleable' とは、文書 m に対する署名 (r, s) から同一の文書 m に対する別の署名 (r', s') が第三者により構成できる性質をいう。ECDSA では、署名 (r, s) に対して $(r, -s)$ も正当な署名となり、上記の性質が確認できる。このようなずれの生じる原因は、ECDSA では群要素の表現を定める関数 σ がランダムとはみなせず、代数的な構造をもっていることによる。

以上から、ECDSA の証明可能安全性について generic group model での安全性証明が与える意義は少ないと考えられる。

還元関数に由来する脆弱性

楕円曲線上の点 $R = (x, y)$ を $\mathbb{Z}/n\mathbb{Z}$ の元 r に写像する ECDSA の還元関数 $f(R)$ には $f(R) = f(-R)$ なる性質がある。具体的には $f(R) = x \bmod n$ である。この性質を利用して、次の 2 つの脆弱性が指摘されている [5]。

- duplicate signature
- malleability

malleability は上の証明可能安全性の節に記述した性質である。この性質には注意すべきであるが、実運用時にはほとんど問題にならないと考えられる。以降では、duplicate signature について説明する。ECDSA では duplicate signatures を発行する不正ができる。duplicate signatures とは、(m_1, r, s),(m_2, r, s), $m_1 \neq m_2$ なる 2 組の署名をいう。不正者は、ECDSA の鍵生成フェーズも含めて不正をはたらく。ECDSA の検査式 $sR = h(m)G + r(dG)$ が、$R = kG, r = f(R), e_1 = h(m_1)$ の場合と $R' = -R = -kG, r = f(R') = f(-R), e_2 = h(m_2)$ の場合とで共に成立するように秘密鍵 d を定める。連立式を d について解くと $d = -(e_1 + e_2)/(2r) \bmod n$ となり、このように定めた d はメッセージ (m_1, m_2) に対する duplicate signature を発行可能な鍵となる。ただし、一度 duplicate

signature が発行されると秘密鍵 d は第三者にも分かるため、不正者にもリスクがある。

この脆弱性は、署名の基本機能である否認防止に関わる不正につながる可能性があり、運用によっては問題になるケースが考えられる。問題となる運用例は、署名の受領者が署名部 (r, s) だけを保存し、メッセージ m は保存しないケースである。この場合、後で署名者 (不正者) がメッセージ m を duplicate signature となる m' に置き換える可能性がある。この不正に対しては、署名部だけでなくメッセージ m も必ず一緒に保存することが対策になる。メッセージのハッシュ結果 $e = h(m)$ を署名部と一緒に保存することでも良い。このような運用上の注意により、duplicate signature の問題は回避可能である。信頼できる第三者機関がユーザ鍵を生成して配布することも防止策になる。

なお、鍵生成フェーズを利用した不正は、DSA にも例がある [6]。DSA において、$h(m_1) \equiv h(m_2) \bmod q$ となるよう (m_1, m_2) を選び、その差が指定サイズの素数 q になるかどうかを検査することを繰り返して、(m_1, m_2, q) の組を定めると、m_1 に対する署名は m_2 に対する署名にもなる。DSA の場合には、公開鍵の生成に不正をはたらく攻撃であるため、検証可能な鍵生成手順の利用が解決策になる。ECDSA の duplicate signature の不正は、ユーザ秘密鍵の生成手順も含んだ不正行為であるため、DSA での対策をそのままでは利用できない点に注意が必要である。

DSKS(duplicate-signature key selection) 特性

ECDSA では、ベースポイントの変更を許すと、不正な鍵生成を行うことにより、同一の署名文に対する正当な署名者を追加することができる。

攻撃者は与えられたメッセージ m とその楕円曲線パラメータ T(ベースポイント G を含む) および公開鍵 Q に対する署名 (r, s) に対して、1 以上 $n-1$ 以下の乱数 c を生成し、$t = s^{-1}(h(m)+rc) \bmod n (\neq 0)$, $X = s^{-1}h(m)G + s^{-1}rQ$, $G' = t^{-1}X$, $Q' = cG'$ を計算する。すると、(r, s) はメッセージ m に対する、楕円曲線パラメータ T(ただしベースポイントのみ G' に変更) および公開鍵 Q' に対する正しい署名となる。実際、$X' = s^{-1}(h(m)G' + rQ') = s^{-1}(h(m)G' + rcG') = tG' = X$ である。

上記の不正を防ぐ方法として、ベースポイント G をシステムで共通とする、あるいは信頼できる第三者機関がベースポイントの生成を行う、などがある。

楕円曲線パラメータの検証

ECDSA で利用されるシステムパラメータである楕円曲線パラメータにはトラップドアの無いことを検証可能にすべきという意見がある。

これに関連して、ECDSA in SEC1 では、楕円曲線上の離散対数問題の攻撃法に対する回避条件をチェックする形式で楕円曲線パラメータの有効性を確認する手法が記載されている。現時点では示されている回避条件に不足はないが、今後新たな攻撃法が発見される可能性もあり、それがトラップドアとして利用される潜在的な脅威もあるため、攻撃法の動向には注意が必要である。

擬似乱数生成器

ANSI X9.62[1] では FIPS186-2(DSS) に記載されている擬似乱数生成器が規定されている。この擬似乱数生成器によって生成された $k = rand \bmod n$ は $[1, n-1]$ で一様に分布しないため、この性質を利用した DSA に対する攻撃が Bleichenbacher によって指摘されている。NIST はこの攻撃の対策として擬似乱数生成器の改訂を FIPS186-2 change notice に示している。具体的には、2 つの乱数 $rand, rand'$ を連結した 2 倍のサイズの乱数データを用いて $k = (rand||rand') \bmod n$ とするものである。現状では、擬似乱数生成器の仕様変更は ECDSA に対しては提唱されていないが、ECDSA での擬似乱数生成器にも同様の攻撃が適用される可能性があり、動向に注意すべきである。この件に関しては「第 2.3.1 節 DSA」も参照されたい。

参考文献

[1] ANSI X9.62, “Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digial Signature Algorithm (ECDSA)”, American National Standard for Financial Services, 1998.

[2] D.Brown, “The exact security of ECDSA”, Technical Report CORR 200-34, Dept. of C&O, University of Waterloo, 2000. available at `http://www.cacr.math.uwaterloo.ca`

[3] Standards for Efficient Cryptography, “SEC1:Elliptic Curve Cryptography”, Certicom Research, Ver.1.0, September 2000, available at `http://www.secg.org/secg_docs.htm`

[4] Standards for Efficient Cryptography, “SEC2:Recommended Elliptic Curve Domain Parameters”, Certicom Research, Ver.1.0, September 2000, available at `http://www.secg.org/secg_docs.htm`

[5] J.Stern, D.Pointcheval, J.M.-Lee, and N.P.Smart, “Flaws in Applying Proof Methodologies to Signature Schemes”, Advanced in Cryptology – CRYPTO2002, Lecture Notes in Computer Science **2442**, Springer-Verlag, pp.93–110, 2002.

[6] S.Vaudenay, “Hidden Collisions on DSS”, Advances in Cryptology – CRYPTO’96, Lecture Notes in Computer Science **1109**, Springer-Verlag, pp.83–88, 1996.

2.3.3 ESIGN 署名

2.3.3.1 評価対象暗号技術

- “ESIGN 仕様書”,
  2001 年度 CRYPTREC 応募書類, 日本電信電話株式会社
  NTT 情報流通プラットフォーム研究所, (2001),
  http://info.isl.ntt.co.jp/esign/CRYPTREC/index-j.html
- “TSH-ESIGN:Efficient Digital Signature Scheme Using Trisection Size Hash” (T. Okamoto, E. Fujisaki, H. Morita), Submission to P1363a, (1998),
  http://grouper.ieee.org/groups/1363/StudyGroup/submissions.html

2.3.3.2 技術概要

ESIGN 署名は、署名を目的とした暗号アルゴリズムである。ESIGN 署名の仕様は複数存在する。複数ある ESIGN 署名を大きく 2 つに分類すると、1 つは、証明可能安全性が示されていない仕様である。もう 1 つは、e 乗根近似問題の困難性の仮定とランダムオラクルモデルのもとで、Single-Occuerence 適応的選択文書攻撃に対して存在的偽造不可であるという証明可能安全性を有するとされる仕様である。

前者を ESIGN[10] [*5] と呼び、後者を TSH-ESIGN[12, 6] と呼ぶ。本章の以下の節において、特に断わりのない場合は、2 つの仕様の ESIGN 署名に対する共通の評価であると考えていただきたい。

2.3.3.3 技術仕様

ESIGN 署名は、何度かの仕様の変更が行われており、CRYPTREC 投稿版以外にも様々なバージョンがある [10]。それらの仕様の違いを表 2.5 にまとめる。年数経過とともに、提案者の推奨するパラメータはより大きくなっており、また安全性理論研究の進展によって、署名方式における最強の意味での安全性を (ある仮定のもとで) 証明できるスキームに変更されている。

[*5] 平成 13 年 総務省 法務省 経済産業省 告示第 2 号 (平成 13 年 4 月 27 日 官報 号外第 86 号) の電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針の第 3 条第 4 号に記載された方式。平成 14 年 総務省 法務省 経済産業省 告示第 13 号 (平成 14 年 11 月 21 日 官報 第 3492 号) の一部改正により削除されたところ。

表 2.5: 各種 ESIGN 署名

	提案者の推奨するパラメータ	証明可能安全性
電子署名法に係る指針に記載されていた方式	$\|n\| \geq 1024, e \geq 8$	無
CRYPTREC 2001	$\|n\| = 1152, e = 1024$	
CRYPTREC 2000	$\|n\| \geq 960, e \geq 8$	有 ($n = p^2q$ 型素因数分解仮定、e 乗根近似仮定とランダムオラクルモデルのもとで、Single-Occurrence 適応的選択文書攻撃に対して存在的偽造不可)
IEEE P1363a	—	
TSH-ESIGN (NESSIE 投稿版)†	—	有 (SO-CMA に対する存在的偽造不可)

†NESSIE[8] においては TSH-ESIGN を含め、ESIGN-D および ESIGN-R[4, 5] も評価されたが、いずれも推奨されていない。

ESIGN と TSH-ESIGN のプリミティブ部分の仕様は共通であり、その概要は以下の通りである。

鍵生成

入力: k セキュリティパラメータ (正整数)
e 8 以上の指数 (正整数)
出力: PK 公開鍵 (n, k, e)
SK 秘密鍵 (p, q)
Step 1 k ビットの 2 つの素数 p、q を選ぶ
Step 2 $n = p^2q$ を計算する
Step 3 $PK = (n, k, e)$、$SK = (p, q)$ を出力する

署名生成プリミティブ:SP-ESIGN

入力: SK 秘密鍵 (p, q)
PK 公開鍵 (n, k, e)
f メッセージ、$0 \leq f < 2^{k-1}$ である整数
出力: s 署名、$0 \leq s < n$ である整数
Step 1 $\mathrm{GCD}(r, n) = 1$ を満たす $r \in \{1, 2, \cdots, pq - 1\}$ をランダムに選ぶ
Step 2 $z = f \cdot 2^{2k}$ を計算する
Step 3 $\alpha = (z - r^e) \bmod n$ とする
Step 4 $w_0 = \left\lceil \frac{\alpha}{pq} \right\rceil$ を計算する
Step 5 $t = \frac{w_0}{e \cdot r^{e-1}} \bmod p$ とし、$s = r + tpq$ を計算する
Step 6 s を出力する

署名検証プリミティブ:VP-ESIGN

入力: PK 公開鍵 (n, k, e)
s 署名、$0 \leq s < n$ である整数
出力: f 検証データ、$0 \leq f < 2^{k-1}$ である整数
Step 1 $T = s^2 \bmod n$ を計算する
Step 2 $f = \left\lfloor \frac{T}{2^{2k}} \right\rfloor$ を計算する
Step 3 f が $0 \leq f < 2^{k-1}$ でなかったら、"invalid"と出力して終了
Step 4 f を出力する

注意: ESIGN 署名が安全であるためには、2 つの素数 p、q は異なるものであることが必要である。しかし、ESIGN 仕様書 [10] には明示的にこのことが記述されていない。我々は、p と q は異なるものとして評価を行っている。

2.3.3.4 プリミティブの安全性

ESIGN 署名のプリミティブは、

- e 乗根近似問題
- $n = p^2 q$ 型素因数分解問題

の 2 つに安全性の根拠をおいている。この 2 つの問題のいずれかが解ければ、ESIGN 署名の秘密鍵が第三者に露呈するか、あるいは署名の偽造に成功する。我々はこの 2 つの問題について評価を行った。

■e 乗根近似問題 ESIGN 署名が安全性の根拠とする e 乗根近似問題とは、次のような問題である [11]。

> **定義 1 (AER 問題)** $\mathcal{G}$ を ESIGN の鍵生成とする。e 乗根近似問題 (AER 問題) とは、$pk := \{n, e\} \leftarrow \mathcal{G}(1^k)$ と $y \leftarrow_R \{0,1\}^{k-1}$ が与えられたとき、$0 \| y = [x^e \bmod n]^k$ となるような $x \in (\mathbb{Z}/n\mathbb{Z}) \setminus p\mathbb{Z}$ を見つける問題である。

また、AER 問題が難しいという仮定は次のように定義される [11]。

> **定義 2 (AER 仮定)** どのような確率的多項式時間アルゴリズム Adv に対しても、全ての定数 c、十分大きな値 k に対して、
>
> $$Pr[Adv(k, n, e, y) \rightarrow x] < 1/k^c$$
>
> が成立するとき、e 乗根近似問題は難しいという。ここで、$0 \| y = [x^e \bmod n]^k$ であり、確率は $\mathcal{G}$ と Adv の確率空間上で取られる。e 乗根近似問題が難しいという仮定は、e 乗根近似仮定 (AER 仮定) と呼ばれる。

- **$e = 2$ と $e = 3$ の場合**
  $e = 2$ の場合は、Brickell と DeLaurentis の方法 [1] により署名の偽造に成功する。その方法の概要は次の通りである。

x を $n^{1/2}$ に近い整数とする。この時、$x^2 \bmod n$ は $O(n^{1/2})$ であり、メッセージ $m = 0$ の場合の ESIGN 署名の検証式を満足する。この原理を任意の m に対して適用できるように、連分数展開を用いて平方根の近似値を求めるようにした方法である。
Brickell と DeLaurentis の方法は、$e = 3$ の場合にも容易に拡張できる。
また、$e = 2$ の場合に、Vallée と Girault と Toffin は LLL アルゴリズムのような格子基底縮小アルゴリズムを利用した ESIGN 署名に対する署名偽造方法を発表した [16, 17]。格子基底縮小アルゴリズムを用いて有限体上の多変数多項式を解くことに関しては、Coppersmith による改良 [2, 3] が知られている。

- **$e \geq 4$ の場合**
  $e \geq 4$ の場合、法 n を素因数分解すること以上に効率的な解法は、現在のところ知られていない。

■$n = p^2q$ 型素因数分解問題　法 n の素因数分解が与えられれば e 乗根近似問題を解くことができる。ESIGN 署名の法は、RSA 暗号で用いられる $n = pq$(p と q は同じ大きさ) という型とは異なり、$n = p^2q$(p と q は同じ大きさ) という型をしている。この型の素因数分解問題の困難さを考察することが必要である。素因数分解問題の困難さに関しては、2.4.1 節を参照のこと。

2.3.3.5　スキームの安全性

前節で述べたように、ESIGN 署名には複数の仕様が存在する。証明可能安全性を今のところ有しない ESIGN と、$n = p^2q$ 型素因数分解仮定と e 乗根近似仮定と、ランダムオラクルモデルのもとで、適応的選択文書攻撃に対して存在的偽造不可であるという証明可能安全性を有するとされる TSH-ESIGN の 2 つの仕様に分類される。それぞれの場合についての安全性評価を以下に述べる。

■署名の安全性について　安全性評価は、署名方式への攻撃の種類を分類し、その上で、

1. 利用される数学的問題 (ESIGN 署名の場合、e 乗根近似問題や素因数分解問題) の安全性評価
2. 利用される数学の問題と署名方式との関連の評価

の 2 つを行う。前節では 1 の評価を行った。本節では 2 の評価を行う。まず、署名方式に対する攻撃の種類の分類と、偽造の種類の分類を表 2.6 と 2.7 に示す [15, 14]。署名方式における最強の安全性は、

> “適応的選択文書攻撃 (CMA) に対して存在的偽造不可”

ということである。

表 2.6: 署名方式への攻撃の種類

攻撃方法		内容
受動的攻撃	直接攻撃	公開鍵のみを利用して行う攻撃
	既知文書攻撃	いくつかのランダムなデータに対応する署名を入手できる場合の攻撃
能動的攻撃	選択文書攻撃	攻撃者があらかじめ指定したいくつかの署名に対応する署名対象データを入手できる場合の攻撃。ただし、署名者に署名させるデータを攻撃に先立って全て選択しなければならない。
	Single-Occurrence 適応的選択文書攻撃	選択文書攻撃における署名の選択を、それまでに入手した署名とそれに対応する署名対象データに関する情報を参考にしながら決定することができる場合の攻撃。ただし 1 つの文書に対する署名は 1 つしか入手できない。
	適応的選択文書攻撃	選択文書攻撃における署名の選択を、それまでに入手した署名とそれに対応する署名対象データに関する情報を参考にしながら決定することができる場合の攻撃。

表 2.7: 署名の偽造の種類

偽造の種類	内容
一般的偽造	任意のデータに対して署名を偽造できる
選択的偽造	攻撃者があらかじめ選んだいくつかのデータに対して署名を偽造できる
存在的偽造	少なくともある特定のデータに対して署名を偽造できる

署名スキームが決定論的でない場合、1 つのメッセージに対して正当な署名が複数存在することになる。この場合、CMA では 1 つのメッセージにつき署名オラクルへの問い合わせを複数回できる (複数の署名を入手できる)。これに対し、Stern は、

“Single-Occurrence 適応的選択文書攻撃 (SO-CMA)”

という攻撃モデルを定義している [14]。SO-CMA は、一般の適応的選択文書攻撃よりも、より攻撃能力が制限されたモデルで、SO-CMA は、1 つのメッセージにつき、署名オラクルへの問い合わせは 1 回しか許されない (署名は 1 つしか入手できない) というものである。

■TSH-ESIGN　TSH-ESIGN のエンコーディングでは、出力長が $k-1$ ビットであるハッシュ関数 H を用いてメッセージ m のハッシュ値を計算し、m は次のようにエンコーディングされる。

$$0||H(m)||0^{2k}$$

Stern は TSH-ESIGN 署名に対し、ランダムオラクルモデルのもとで次の安全性を証明した [14]。

定理 3 (Stern[14]) TSH-ESIGN 署名スキームに対して存在的偽造を生成する SO-CMA 攻撃者を $\mathcal{A}$ とする。$\mathcal{A}$ の攻撃成功確率を ε、攻撃時間を τ、ハッシュ関数への問い合わせ回数を q_H、署名オラクルへの問い合わせ回数を q_s とすると、e 乗根近似問題は次式を満たす確率 ε'、時間 τ' で解ける。

$$\varepsilon' \geq \frac{\varepsilon}{q_H} - (q_H + q_s) \times \left(\frac{3}{4}\right)^k - \frac{1}{2^{k-1}}$$
$$\tau' \leq \tau + k(q_s + q_H) \cdot T_{exp}(k)$$

ここで、$T_{exp}(k)$ は、$3k$ ビットの法でのべき乗剰余演算の計算時間を表す。

この定理の導出のアプローチおよび証明は、Shoup が OAEP に関して議論したこと [13] に関連している。Shoup 流の Game を使った方法である。この定理が証明していることは、一般の適応的選択文書攻撃 (CMA) に対する存在的偽造不可ではなく、SO-CMA に対する存在的偽造不可であることに注意。Stern はまた、次の 2 つも指摘している。

- 提案者による証明 [12] は、攻撃モデルとして SO-CMA を暗黙に仮定している。
- TSH-ESIGN の証明可能安全性を一般の CMA に対する証明可能安全性へ拡張する方法は現在のところ知られていない。

TSH-ESIGN に対する具体的攻撃法や署名偽造法が発見されたわけではないが、通常の CMA に対してではなく、SO-CMA に対してしか証明可能安全性を有しないという状況は望ましくない。

なお、TSH-ESIGN を通常の CMA に対して証明可能安全性を有するように改良した ESIGN 署名が Granboulan によりすでに提案されている [4, 5]。[4, 5] では確定的スキーム ESIGN-D と、ランダム化スキーム ESIGN-R の 2 つのスキームが示されている。ESIGN-D と TSH-ESIGN の仕様上の違いは、乱数生成部分を少し変更しただけである。なお、TSH-ESIGN と ESIGN-D は互換性を持つことに注意。すなわち、TSH-ESIGN で生成した署名を ESIGN-D は正しく検証し、ESIGN-D で生成した署名を TSH-ESIGN は正しく検証する。このことは、アルゴリズムを見れば明らかである。NESSIE においては、TSH-ESIGN が応募暗号であるが、ESIGN-D および ESIGN-R も評価されたが、いずれも推奨されていない [9]。

■ESIGN　電子署名法に係る指針に記載されていた ESIGN 署名、すなわち 2001 度の CRYPTREC へ応募された ESIGN では、EMSA と呼ばれるメッセージエンコーディング手法が適用されている [10]。ESIGN は、無視できない確率で署名の偽造に成功することが Stern により報告された [14]。この概要を以下に述べる。

EMSA エンコーディング手法の概要は次の通りである。この変換においては、まず長さ $hLen \leq k - 16$ ビットのストリングを出力するハッシュ関数 H を用いて、メッセージ m のハッシュ値を計算し、次に長さ $k - hLen$ ビットのストリングをハッシュ値に付加する。そのフォーマットは、16 進数表記で表すと次の通りである。

$$00 \| PS \| \mathrm{FF} \| H(m)$$

ここで、*PS* は FF 以外のバイト列である。このようにしてメッセージ m は k ビットのストリングに変換される。

このパディングストリング *PS* が安全性に悪影響を与えている。安全性は e 乗根近似問題には帰着せず、次のような変形版 e 乗根近似問題に関連する。

定義 4 (変形版 e 乗根近似問題 [14]) 与えられた $3k$ ビットの n と $hLen$ ビットの v に対して、$x^e \bmod n$ を 2 進表現した時に、ビット位置 $2k+1, \cdots, 2k+hLen$ に v が現れるような x を求めよ。

この変形版 e 乗根近似問題は、e が小さい時は容易に解ける。次の条件を満たす時に、署名の偽造が可能となる [14]。

$$2k \geq e(hLen + \log 2 + 8)$$

電子署名法に係る指針では、平成 14 年 11 月まで、ハッシュ関数として SHA-1 と MD5 が指定されていたが、SHA-1($hLen = 160$) の場合、上式は、

$$\frac{|n|}{253.04} > e$$

となる。したがって SHA-1 を用いた時、例えば次のような場合に署名の偽造に成功する。

- $|n| = 1024$ かつ $e \leq 4$
- $|n| = 2048$ かつ $e \leq 8$

また MD5($hLen = 128$) の場合、

$$\frac{|n|}{205.04} > e$$

の時、署名の偽造が可能となる。例えば次のような場合に署名の偽造に成功する。

- $|n| = 1024$ かつ $e \leq 4$
- $|n| = 2048$ かつ $e \leq 9$

これらのパラメータは、$|n| = 2048$ かつ $e = 8$ の場合など、これまでの電子署名法に係る指針に記載されていたパラメータを含んでいる。

2.3.3.6 補助関数について

ESIGN 署名では、補助関数としてハッシュ関数を用いている。ESIGN には、ハッシュ関数として MD5 の使用を指定した方式と、SHA-1 の使用を指定した方式とがある。このうち、MD5 の使用を指定した方式は推奨できない。ハッシュ関数の安全性に関しては、4 章および文献 [7] を参照のこと。

2.3.3.7 実装性

提案者による実装 [11] では、Celeron 800MHz において、法 n が 1152 ビット、安全性パラメータ e が 1024 の時、鍵生成 610ms、署名生成 1.04ms、署名検証 0.70ms である。

RSA 署名や ECDSA 署名は、様々な研究者により様々なプラットフォームで実装され、速度計測が行われている。しかし ESIGN 署名の実装は、提案者によるもの以外ほとんど知られていない。したがって、どの程度の高速化ができるかは未知である。ただし、べき乗剰余演算等、RSA 暗号の高速化に用いられている高速化技術の一部は、ESIGN 署名にも適用することが可能である。

ESIGN 署名の署名生成速度は、RSA 署名と比べて高速であると言える。

2.3.3.8 ESIGN 署名のまとめ

- ESIGN: 電子署名法に係る指針に記載されていた安全性パラメータの一部 (例えば SHA-1 を用い、かつ n が 2048 ビットの時、e が 8 以下の場合など) は、無視できない確率で署名の偽造に成功する。このため、平成 14 年 11 月、電子署名法に係る指針の一部改正により、ESIGN は削除された。
- TSH-ESIGN: 証明される安全性は、通常の適応的選択文書攻撃に対する存在的偽造不可ではなく、より攻撃能力が制限された Single-Occurrence 適応的選択文書攻撃に対する存在的偽造不可である。

参考文献

[1] E. Brickell, J. DeLaurentis, "An Attack on a Signature Scheme proposed by Okamoto and Shiraishi," *Advances in Cryptology – CRYPTO'85*, LNCS, **218** (1986), Springer-Verlag, 28–32.

[2] D. Coppersmith, "Finding a Small Root of a Univeriate Modular Equation," *Advances in Cryptology – EUROCRYPT'96*, LNCS, **1070** (1996), Springer-Verlag, 155–165.

[3] D. Coppersmith, "Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known," *Advances in Cryptology – EUROCRYPT'96*, LNCS, **1070** (1996), Springer-Verlag, 178–189.

[4] L. Granboulan, "How to Repair ESIGN," Cryptology ePrint Archive, Report 2002/074, (2002),
available at `http://eprint.iacr.org`

[5] L. Granboulan, "How to Repair ESIGN," Proceedings of Third NESSIE Workshop, (2002)

[6] "IEEE P1363a Draft Version 9 Standard Specifications for Public Key Cryptography: Additional Techniques," *IEEE* (2001),
available at `http://grouper.ieee.org/groups/1363/StudyGroup/submissions.html`

[7] 情報処理振興事業協会, “暗号技術評価報告書 CRYPTREC Report 2000,” (2000).

[8] NESSIE, New European Schemes for Signatures, Integrity, and Encryption, https://www.cosic.esat.kuleuven.ac.be/nessie/

[9] NESSIE, “Portfolio of recommended cryptographic primitives,” available at https://www.cosic.esat.kuleuven.ac.be/nessie/

[10] NTT 情報流通プラットフォーム研究所, “ESIGN 仕様書,” (2001), available at http://info.isl.ntt.co.jp/esign/CRYPTREC/index-j.html

[11] NTT 情報流通プラットフォーム研究所, “ESIGN 自己評価書,” 2001 年度 CRYPTREC 応募書類, (2001), available at http://info.isl.ntt.co.jp/esign/CRYPTREC/index-j.html

[12] T. Okamoto, E. Fujisaki, H. Morita, “TSH-ESIGN:Efficient Digital Signature Scheme Using Trisection Size Hash,” Submission to P1363a, (1998), available at http://grouper.ieee.org/groups/1363/StudyGroup/submissions.html

[13] V. Shoup, “OAEP Reconsidered,” *Advances in Cryptology – CRYPTO 2001*, LNCS, **2139** (2001), Springer-Verlag, 239–259.

[14] J. Stern, D. Pointcheval, J.M. Lee, N.P. Smart, “Flaws in Applying Proof Methodologies to Signature Schemes,” *Advances in Cryptology – CRYPTO 2002*, LNCS, **2442** (2002), Springer-Verlag, 93–110.

[15] 宇根正志, 岡本龍明, “公開鍵暗号の理論研究における最近の動向,” IMES Discussion Paper Series 98-J-28, (1998)

[16] B. Vallée, M. Girault, P. Toffin, “How to Break Okamoto’s Cryptosystem by Reducing lattice Bases,” *Advances in Cryptology – EUROCRYPT’88*, LNCS, **330** (1988), Springer-Verlag, 281–291.

[17] B. Vallée, M. Girault, P. Toffin, “How to Guess *l*th Roots Modulo *n* by Reducing Lattice Bases,” *AAECC-6*, LNCS, **357** (1988), Springer-Verlag, 427–442.

2.3.4 RSA (RSA-PSS, RSASSA-PKCS1-v1_5, RSA-OAEP, RSAES-PKCS1-v1_5)

2.3.4.1 評価対象暗号技術

- PKCS #1 v2.1: RSA Cryptography Standard,
  RSA Laboratories, June 14, 2002,
  http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html
  ただし、モジュラスはほぼ同じサイズの 2 つの異なる素数の積によるものとする。なお、RSA-PSS および RSA-OAEP は、PKCS #1 v2.1 において、それぞれ RSASSA-PSS および RSAES-OAEP として定義されている方式と同一である。

2.3.4.2 技術概要

CRYPTREC では RSA プリミティブ を利用した暗号技術として、 RSA-OAEP, RSAES-PKCS1-v1_5(RSA 守秘), RSA-PSS, および RSASSA-PKCS1-v1_5(RSA 署名) を評価している。RSA-OAEP(Optimal Asymmetric Encryption Padding) と RSA 守秘は情報の秘匿を目的とした暗号アルゴリズムであり、RSA-PSS (Probabilistic Signature Scheme) と RSA 署名はデジタル署名を目的とした暗号アルゴリズムである。

RSA を利用した守秘に関しては、1) RSA-PKCS #1 v1.5 に記載された方式. (以降では RSAES-PKCS1-v1_5(RSA 守秘) と記す)、2) RSA-OAEP (PKCS#1 v2.1 に記載された方式. 2001 年度に CRYPTREC に応募された方式) を対象とした.

RSA-OAEP については 2001 年度から評価を継続しており、RSAES-PKCS1-v1_5(RSA 守秘) については 2002 年度に暗号技術評価委員会より「使用実績のある暗号技術」ということで採用が要請され評価を行なった。

規格書 PKCS #1 v1.5 で標準化された RSA 守秘が、1998 年に Bleichenbacher [5] によってある種の攻撃 (暗号文が予め定められた条件をみたさないという情報を利用した攻撃) で解読できることが指摘されて以来、公開鍵暗号の暗号アルゴリズムが適応的選択暗号文攻撃に対して頑強性 (IND-CCA2) を みたすことが必須となった [3, 1]。

RSA-OAEP は RSA 法のプリミティブが部分一方向性をみたす[*6]とき、最強の安全性 (IND-CCA2) をみたすという証明可能安全が示された暗号方式である。

一方、RSA を利用した署名に関しては、1) いわゆる「教科書的」RSA 署名、2) ANSI X9.31、3) RSASSA-PKCS1-v1_5(RSA 署名)(電子署名法に係る指針に記載された方式 4) RSA-FDH (Full-Domain Hash Schems: FDH) 、5) RSA-PSS (Bellare-Rogaway の論文版)、6) RSA-PSS (IEEE P1363a 版) など、多数の仕様が存在する。

[*6] 文献 [15] によって、この条件は、「RSA 問題の 困難性」(RSA プリミティブの一方向性) と同値であることが示されている.

RSA-PSS は RSA 法のプリミティブが一方向性をみたすとき、最強の安全性 (適応的選択文書攻撃に対して存在的偽造を許さないこと) をみたすという証明可能安全が示された署名方式である。

署名法の詳細評価にあたっては、電子署名法に係る指針に記載されている RSASSA-PKCS1-v1_5(RSA 署名) および 2001 年度 CRYPTREC に応募された RSA-PSS (IEEE P1363a 版) を対象とした。

2.3.4.3 技術仕様

RSA プリミティブ

公開鍵を (N, e)、秘密鍵を (N, d) とする。ここで、e は 3 以上の奇数で $\mathrm{GCD}\{e, (p-1)(q-1)\} = 1$ をみたし、d は $de \equiv 1(\mathrm{LCM}\{p-1, q-1\})$ をみたす。

RSA 暗号化プリミティブ RSAEP/ RSA 署名検証プリミティブ RSAVP を

$$\mathrm{RSAEP}((n, e), x) = \mathrm{RSAVP}((n, e), x) = x^e \bmod N \tag{2.1}$$

で、復号プリミティブ RSADP/ 署名生成プリミティブ RSASP を

$$\mathrm{RSADP}((n, d), y) = \mathrm{RSASP}((n, d), y) = y^d \bmod N \tag{2.2}$$

で定義する。x と y は $\{0, 1, \ldots, N-1\} = Z_N$ の整数である。N のオクテット数を k (以降では $|N| = k$ と記す) とする。

RSA-PSS

RSA-PSS の構成は以下の通り。

EMSA-PSS-Encode (M, emBits)

1. M のオクテット長がハッシュ関数の入力制限 (SHA-1 の場合には $2^{61} - 1$ オクテット) よりも長いなら、“message too long” を出力して停止する。
2. 長さ hLen オクテットの列、$mHash = Hash(M)$ を生成する。
3. $\mathsf{emBits} < 8\mathsf{hLen} + 8\mathsf{sLen} + 8t + 1$ なら、“encoding error” を出力して停止する。(t は Trailer field のオプションであり、値として 1 または 2 をとる。)
4. 長さ sLen のランダムな列 *salt* を生成する。sLen = 0 ならば、*salt* は空列となる。
5. $m' = 00\,00\,00\,00\,00\,00\,00\,00 \,\|\, mHash \,\|\, salt$。$m'$ は長さ $(8 + \mathsf{hLen} + \mathsf{sLen})$ オクテットの列であり、先頭に 8 個の zero octet ($P = 00\,00\,00\,00\,00\,00\,00\,00$ とおく。) を含む。
6. 長さ hLen オクテットの列、$H = Hash(m')$ を生成する。
7. $(\mathsf{emLen} - \mathsf{sLen} - \mathsf{hLen} - t - 1)$ 個の zero octet を含んだデータ列 PS を生成する。$|PS| = 0$ のこともある。
8. $DB = PS \,\|\, 01 \,\|\, salt$ とおく。
9. $dbMask = \mathsf{MGF}(H, \mathsf{emLen} - \mathsf{hLen} - t)$ とする。
10. $MaskedDB = DB \oplus dbMask$ とする。

11. *MaskedDB* の最左 octet 中の左から (8emLen – emBits) ビットを zero に設定する。
12. $t = 1$ かつ $TF = bc$ とする[*7]。
13. $EM = MaskedDB \| H \| TF$ とする。
14. *EM* を出力する。

EMSA-PSS-Decode (*M*, *EM*, emBits)

1. *M* のオクテット長がハッシュ関数の入力制限 (SHA-1 の場合には $2^{61} - 1$ オクテット) よりも長いなら、“inconsistent” を出力して停止する。
2. hLen オクテットの列を $mHash = Hash(M)$ とする。
3. emBits < 8hLen + 8sLen + 8t + 1 なら、“dencoding error” を出力して停止する。
4. $t = 1$ かつ $TF = bc$ とする。
5. *EM* の一番右の t オクテットが TF に一致しなければ、“inconsistent” を出力して停止する。
6. $EM = MaskedDB \| H \| TF$ とみなす。ここで $|MaskedDB|$ = emLen – hLen – t、$|H|$ = hLen。
7. *MaskedDB* の最左オクテット中の左から (8emLen – emBits) ビットが zero に一致しなければ、“inconsistent” を出力して停止する。
8. $dbMask$ = MGF(H, emLen – hLen – t) とする。
9. $DB = MaskedDB \oplus dbMask$ とする。
10. *DB* の左から (8emLen – emBits) ビットを zero に設定する。
11. *DB* の右から (emLen – hLen – sLen – t – 1) オクテットが zero に一致しない、あるいは右から第 (emLen – hLen – sLen – t) オクテット目が 01 に一致しなければ、“inconsistent” を出力して停止する。
12. *DB* の最後の sLen オクテットを *salt* とする。
13. $m' = 00\,00\,00\,00\,00\,00\,00\,00 \| mHash \| salt$。$m'$ は先頭に 8 個の zero octet を含む (8 + hLen + sLen) オクテットの列。
14. hLen オクテットの列を $mHash' = Hash(m')$ とする。
15. $H = H'$ ならば “consistent” を出力する。他の場合は “inconsistent” を出力する。

RSA-PSS の署名作成は文書 *M* に対して以下の処理をする。RSA-PSS の署名検証は署名文 *S* に対して以下の処理をする。

RSA-PSS-Sign ((*n*, *d*), *M*)

1. *EM* = EMSA-PSS-Encode (*M*, modBits – 1). encoding 操作が “message too long” を出力するとき、“message too long” を出力して停止する。
2. *S* = RSASP((*n*, *d*), *EM*)。

[*7] 仕様書 [20] では TF に $Hash_ID \| cc$ の設定も規定されていた。仕様書 [21] では TF に *bc* (SHA-1) の設定のみが許されている。

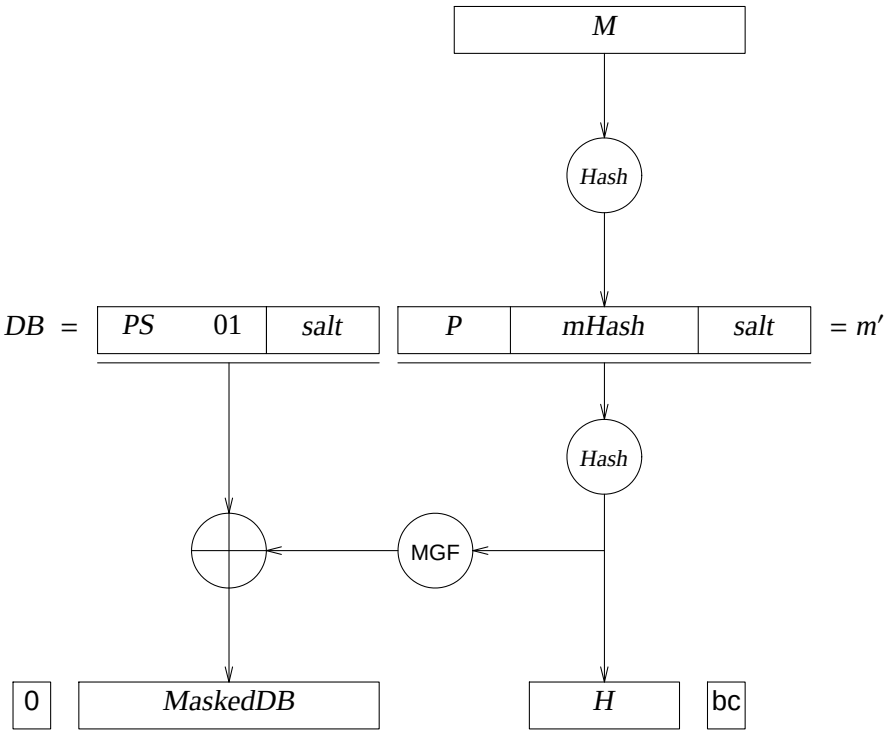

図 2.1: RSA–PSS

3. 署名 S を出力する。

RSA-PSS-Verify $((n, e), M, S)$

1. $EM = \mathsf{RSAVP}((n, e), S)$。
2. $Result$ = EMSA-PSS-Decode (M, EM, emBits)。ここで、$\mathsf{emLen} = \lceil(\mathsf{modBits} - 1)/8\rceil$ オクテット、modBits は法 n のビット長。encoding 操作が “consistent” を出力するなら、“valid” を出力する。その他の場合には、”signature invalid” を出力する。

注意 5 なお、ある与えられた鍵対について、Trailer field オプション、ハッシュ関数、および MGF 関数は一定でなければならないことが提案者より指摘されている。

RSASSA-PKCS1-v1_5

RSASSA-PKCS1-v1_5(RSA 署名)(電子署名法に係る指針に記載された方式) は、規格書 PKCS#1 v1.5[19] で規定され、規格書 PKCS#1 v2.1[20, 21] にも引き継がれている。

RSA 社から出版されたこの三つの文書における署名法に関する記述は以下のように整理できる。

1. 規格書 v1.5 には、ハッシュ関数 MD5 についての記述 (OID) はあるが、ハッシュ関数 SHA-1 についての記述 (OID) はない (文献 [19] の 11 章)。

00	01	$PS(FF\cdots FF)$	00	T(先頭に hash_Id を含む)

図 2.2: EMSA-PKCS1-v1_5 の出力形式

2. 規格書 v2.0 には、EMSA-PKCS1-v1_5 エンコーディング手法として、新たに SHA-1 が利用できるように記述されている (OID がある)(文献 [20] の 10.1 節)。
3. 規格書 v2.1 には、EMSA-PKCS1-v1_5 エンコーディング手法として、新たに SHA-256, 384, 512 が利用できるように記述されている (OID がある)(文献 [21] の 9.2 節)。

EMSA-PKCS1-v1_5 エンコーディング手法として規定されたフォーマットは図 2.2 のとおり。ここで、T は Distingushed Encoding Rule (DER)[*8] によって規定されており、先頭のフィールドはハッシュ関数を特定し、次のフィールドはハッシュ値を含んでいる。また、PS は、値 FF (16 進表示) から構成された 8 オクテット以上のオクテット列である。

RSA-OAEP

RSA-OAEP の構成は以下の通り。

EME-OAEP-Encode (M, P, emLen)

1. P のオクテット長がハッシュ関数の入力制限 (SHA-1 の場合には $2^{61}-1$ オクテット) よりも長いなら、“parameter string too long” を出力して停止する。
2. mLen > emLen – 2hLen – 2 なら、“message too long” を出力して停止する。
3. (emLen – mLen – 2hLen – 2) 個の zero octet を含んだデータ列 PS を生成する。$|PS| = 0$ でもよい。
4. 長さ hLen オクテットの列、 $pHash = Hash(P)$ を生成する。
5. $DB = pHash \,\|\, PS \,\|\, 01 \,\|\, M$ とおく。
6. 長さ hLen オクテットのランダムな列、$seed$ を生成する。
7. $dbMask = \mathrm{MGF}(seed, \mathrm{emLen} - \mathrm{hLen} - 1)$ とする。
8. $MaskedDB = DB \oplus dbMask$ とする。
9. $seedMask = \mathrm{MGF}(MaskedDB, \mathrm{hLen})$ とする。
10. $MaskedSeed = seed \oplus seedMask$ とする。
11. $EM = 00 \,\|\, MaskedSeed \,\|\, MaskedDB$ とする。
12. EM を出力する。

EME-OAEP-Decode (EM, P)

1. P のオクテット長がハッシュ関数の入力制限 (SHA-1 の場合には $2^{61}-1$ オクテット) よりも長いなら、“decoding error” を出力して停止する。

[*8] v1.5 では BER encoding(DER encoding を含む) とだけ指定されているので、互換性には注意を要する。

2. emLen < 2hLen + 2 なら、“decoding error” を出力して停止する。
3. $EM = X \parallel MaskedSeed \parallel MaskedDB$ とおく。ここで $|X| = 1, |MaskedSeed| = \mathsf{hLen}, |MaskedDB| = \mathsf{emLen} - \mathsf{hLen} - 1$。
4. $seedMask = \mathsf{MGF}(MaskedDB, \mathsf{hLen})$ とおく。
5. $seed = MaskedSeed \oplus seedMask$ とおく。
6. $dbMask = \mathsf{MGF}(seed, \mathsf{emLen} - \mathsf{hLen} - 1)$ とおく。
7. $DB = MaskedDB \oplus dbMask$ とおく。
8. hLen オクテットの列を $pHash = Hash(P)$ とする。
9. $DB = pHash' \parallel M'$。ここで、$\left|pHash'\right| = \mathsf{hLen}$。
10. $M' = \alpha \parallel T \parallel M$。$T$ は M' 中の zero でない最左 octet とする。$|T| = 1$。
11. もし $pHash' \neq pHash$, $X \neq 00$ あるいは $T \neq 01$ ならば、“decoding error” を出力する[*9]。
12. M' から T と α(すべてが zero からなる) を除いて M を生成する。
13. M を出力する。

RSA-OAEP の暗号化は通信文 M に対して以下の処理をする。RSA-OAEP の復号は暗号文 C に対して以下の処理をする。

RSAES-OAEP-Encrypt $((n, e), M, P)$

1. EM = EME-OAEP-Encode (M, P, k)。encoding 操作が “message too long” を出力するとき、“message too long” を出力して停止する。
2. $C = \mathsf{RSAEP}((n, e), EM)$。
3. C を出力する。

RSAES-OAEP-Decrypt $((n, d), C, P)$

1. $EM = \mathsf{RSADP}((n, d), C)$。
2. M = EME-OAEP-Decode (EM, P)。dencoding 操作が “decoding error” を出力するとき、“decryption error” を出力して停止する。
3. M を出力する。

注意 6 なお、ある与えられた鍵対について、*seed* の長さ、ハッシュ関数、MGF 関数、およびデータ列 *PS* の値は一定でなければならないことが提案者より指摘されている。

RSAES-PKCS1-v1_5

RSAES-PKCS1-v1_5(RSA 守秘) の構成は以下の通り。

[*9] このステップで、エラーの理由にかかわらず、同時に同一のエラー通知で出力することとした。文献 [5, 17] で指摘された攻撃に対する対策として導入された実装法である。

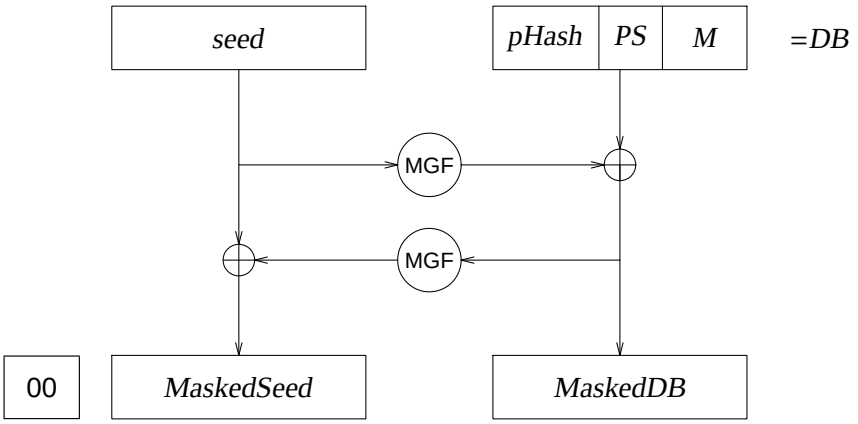

図 2.3: RSA–OAEP–Encode

00	02	*PS* (非 zero octet)	00	*M*(暗号化対象の平文)

図 2.4: EME-PKCS1-v1_5 の形式

RSAES-PKCS1-x1_5-Encrypt $((n, e), M)$

M は暗号化対象のメッセージであり, mLen オクテットの列とする.

1. mLen > k – 11 なら、“message too long” を出力して停止する。
2. 長さ k – mLen – 3 オクテットのランダムに生成された、非 zero octet を含んだデータ列 PS を生成する。PS の長さは少なくとも 8 オクテットとする。
3. PS, M, および他のオクテットを連結して次のようにエンコードされたメッセージ EM を作る: $EM = 00 \| 02 \| PS \| 00 \| M$.
4. C = RSAEP$((n, e), EM)$。
5. C を出力する。

RSA-PKCS1-v1_5-Decrypt $((n, d), C)$

1. C が k オクテットでない、あるいは $k < 11$ ならば、“decryption error” を出力して停止する。
2. EM = RSADP$((n, d), C)$。
3. EM を以下のように、非 zero octet を含んだデータ列 PS とメッセージ M に分離する: $EM = 00 \| 02 \| PS \| 00M$. もし EM の第一オクテットがヘキサ表示の 00 でなければ、もし EM の第二オクテットがヘキサ表示の 02 でなければ、もし M から PS を分離するオクテット 00 が存在しなければ、あるいは PS の長さが 8 オクテットより短いなら、“decoding error” を出力する[*10]。

[*10] このステップで、エラーの理由にかかわらず、同時に同一のエラー通知で出力することとした。文献 [5, 17] で指摘された攻撃に対する対策として導入された実装法である。

4. M を出力する。

2.3.4.4 安全性評価

プリミティブの安全性

RSA 法のプリミティブは、

- $n = pq$ 型素因数分解問題

に安全性の根拠をおいている。素因数分解問題についての評価結果は第 2.4.1 節に報告する。RSA 法の証明可能安全性については、上記の型の素因数分解の困難性との同値性は示せていないが、経験的に安全であると信じられている。長期間広く使われている実績、広範な観点からの安全性評価が行われてきた。すべての評価者から、RSA プリミティブに関する自己評価書の記述に問題なしと報告されている。すでに指摘されている使用制約としては、例えば、暗号・署名に共通の事項として、1) 法の値の共有、2) 秘密鍵 d が小さい場合の脅威、3) 鍵の部分情報から全体の情報の導出の脅威などが報告されていた。また、暗号として使用する場合の事項として、1) 公開鍵 e が小さいときの脅威 (Coppersmith の攻撃) 2) 同報通信環境での脅威 (Håstad, Coppersmith の攻撃) などが報告されている。

よって、RSA パラメータの選択方法には注意を要する。詳しくは、2.5.1.1 節を参照のこと。

RSA-PSS の安全性

確率的署名 (Probabilistic Signature Schemes, PSS) は Bellare と Rogaway によって提案されたデジタル署名のエンコーディング手法である。署名対象の文書に乱数成分を付加することで、RSA 署名のように確定的な署名であっても、毎回異なった署名が生成される。確定的な署名法を確率的に変更するだけでなく、RSA-PSS はランダムオラクルモデルのもとで安全性が証明できる [4]。

安全性の証明された署名法として、全域ハッシュ法 (Full Domain Hash Schems:FDH) [2] や、本人確認法からデジタル署名を構成する変換法 [14] が提案されているが、PSS は これらの方式に比べてより緊密 (tight) な帰着関係を証明できる特徴を有している。

Coron は FDH について帰着関係を緊密にする技法 (Coron の技法) [8] を提案している。Jonsson は、PSS の安全性を Coron の技法を適用して再評価している [16]。論文 [16] では、帰着関係を緊密 (tight) に評価したことに加えて、以下を示した。

1. salt 長を可変にした場合にも安全性の証明を与える。
2. 関数 *Hash*、MGF が相関がある場合も含めて、帰着関係の効率を評価した。

Jonsson の証明は、Hash-ID については検討の対象外としているが、規格書 [21] では Hash-ID は *bc* に固定となったので、検討対象外としたことによる問題は生じないと考えられる。

*salt*のサイズに関しては文献 [11] によって、従来の証明技法では 180 ビットの *salt*のサイズが必要だったものが、同じ安全性を保証するために 30 ビットで十分であることが示された。

RSASSA-PKCS1-v1_5 の安全性

RSASSA-PKCS1-v1_5(RSA 署名) に関しては外部評価者 (2 名) に安全性の評価を依頼した。評価者#1 からは: 「For PKCS#1 v1.5 (and ANSI x9.31), we have seen that the attack of [10] does not apply. To our knowledge, no attack better than factoring the modulus or finding a collision in the hash function, is known for PKCS#1 v1.5 (and ANSI x9.31). 」、評価者#2 からは: 「The attacks are not a threat to the practical security of schemes described in the (ANSI X9.31 and) PKCS#1 v1.5 standards. 」との報告があった。

現在時点 (2002 年 9 月末) までに発見されている RSASSA-PKCS1-v1_5(RSA 署名) に対する偽造法の複雑度は、選択文書攻撃を許したとしても素因数分解の複雑度を下回ることはないことを確認したので、特に安全性の問題はないと考える。

なお、文献 [12] において RSASSA-PKCS1-v1_5(RSA 署名) の安全性に関連する次の定理が証明された。

定理 7 Let $\mathcal{S}$ be the Rabin-Williams partial-domain hash signature scheme with constant γ and hash size k_0 bits. Assume that there is no algorithm which factors a RSA modulus with probability greater than ε within time t. Then the success probability of a forger against $\mathcal{S}$ making at most q_{hash} hash queries and q_{sig} signature queries within time t' is upper bounded by ε', where:

$$\varepsilon' = 8 \cdot q_{sig} \cdot \varepsilon + 32 \cdot (q_{hash} + q_{sig} + 1) \cdot k_1 \cdot \gamma \cdot 2^{-\frac{3}{13} \cdot k_1} \tag{2.3}$$

$$t' = t - k_1 \cdot \gamma \cdot (q_{hash} + q_{sig} + 1) \cdot O(k^3) \tag{2.4}$$

and $k_1 = k_0 - \frac{2}{3}k$.

RSASSA-PKCS1-v1_5(RSA 署名) で規定された署名方式の署名生成プリミティブは Rabin 法 ($e = 2$) でないので、上記の定理は直接には適用できない。しかし、ハッシュ関数の出力サイズが法のサイズの 2/3 以上ならば証明可能安全性が保証できることが示されたことより、この署名技法がさらに改良されれば、将来、RSASSA-PKCS1-v1_5(RSA 署名) で規定された署名方式の安全性が保証される余地が残っているとも解釈できる。

RSA-OAEP の安全性

2000 年末に Shoup によって Bellare-Rogaway によるオリジナル論文 [3] の安全性の主張に誤りが指摘されたことで、安全性についての議論が学会で議論された [18]。

その結果、安全性の証明帰着の効率が低下するものの、RSA-OAEP としての証明可能安全性は RSA 問題の困難性を前提として成り立つことが示された [15] 。また、2001 年に OAEP 実装上の問題点が指摘されたが [17]、現在の仕様では対処済みであることが確認されている。

RSAES-PKCS1-v1_5 の安全性

文献 [9] などにより、本方式に対する攻撃法の進展状況を調査した。

RSAES-PKCS1-v1_5(RSA 守秘) に実効上有効な攻撃法は発見されなかったものの、選択平文攻撃 (Coppersmith の攻撃法 [6, 7], Coron-Joye-Naccache-Paillier の攻撃法）が理論的に可能となりうることには注意を要する。また、選択暗号文攻撃として Bleichenbacher による実装上の弱点をついた攻撃があることにも注意を要する [5]。

RSA-PSS-ES の安全性

2002 年に Coron らによって Messege 回復型-PSS (PSS-R) を暗号方式のパディングに用いても、OAEP と同様の安全性証明が可能なことが指摘された [13]。すなわち、利用者が、暗号文の復号と文書の署名生成に同一の秘密鍵を用いても、暗号系については選択暗号文攻撃に加えて選択文書攻撃を許したとしても暗号系としての安全性を保証でき、署名系については選択文書攻撃に加えて選択暗号文攻撃を許したとしても署名系としての安全性を保証できることが、証明された。

これによって、理論的には、一つの秘密鍵を暗号系の復号鍵と署名系の署名生成鍵に併用しても問題ないことが保証されたことになる。しかし、例えば Manger によるような実装上の問題点も予想されるので、暗号用と署名用に異なる鍵を使用するのが、安全サイドの選択と考えられる。

2.3.4.5 まとめ

RSA-PSS

証明可能安全性の証明はランダムオラクルモデルのもとで信頼できる。ただし、CRYPTREC への提案方式と論文で証明が与えられている方式に若干の相違があるため、対応するパラメータの関係を把握して設計パラメータを選択する必要がある。

2002 年度に実施された電子署名法に係わる指針の一部改正において、RSA-PSS は追加された。

参考情報として、RSA-PSS と学会で議論されている [4] に従った仕様の違い (記法の対応関係) を表にまとめる。

PSS[4] $\longleftrightarrow$ EMESA-PSS
$m \longleftrightarrow P \,\|\, mHash$
$r \longleftrightarrow salt$
$k_0 \longleftrightarrow 8 \times \mathsf{sLen}$
$0^{k_2} \longleftrightarrow PS$
$k_2 \longleftrightarrow \mathsf{emBits} - 8 \times \mathsf{hLen} - 8 \times \mathsf{sLen} - 8$
$k \longleftrightarrow \mathsf{emBits}$
$k_1 \longleftrightarrow 8 \times \mathsf{hLen}$
$w \longleftrightarrow H$
$s \,\|\, t \longleftrightarrow$ *MaskedDB* で先頭 (8emLen – emBits) ビットに zero を設定
$\mathtt{G}(.) \longleftrightarrow \mathsf{MGF}(., \mathsf{emLen} - \mathsf{hLen} - 1)$ で先頭 (8emLen – emBits) ビットに zero を設定
$\mathtt{H}(.) \longleftrightarrow$ Hash function

注意 8 文献 [4] ではビット単位の表記が使われており、RSA-PSS ではバイト単位の表記が使われていることに注意を要する。

電子署名法に係る指針に記載された RSASSA-PKCS1-v1_5(RSA 署名) の使用について

現時点で特に安全性の問題は存在しないが、図 2.2 で示したエンコーディング手法の安全性の検討を継続する必要がある。電子署名法に係る指針では、図 2.2 中の T を生成する方法として MD5 と SHA-1 がハッシュ関数として指定されているが、CRYPTREC Report 2000 で指摘されているように、MD5 の使用は推奨できない。

RSA-OAEP

証明可能安全性の証明はランダムオラクルモデルのもとで信頼できる。ただし、CRYPTREC への提案方式と論文で証明が与えられている方式に若干の相違があるため、対応するパラメータの関係を把握して設計パラメータを選択する必要がある。

参考情報として、RSA-OAEP と学会で議論されている [3] に従った仕様の違い (記法の対応関係) を表にまとめる。

OAEP[3] $\longleftrightarrow$	EME-OAEP
m	$PS \,\|\|\, M$
n	$8 \times$ mLen
0^{k_1}	*pHash*
k_0	$8 \times$ hLen
k_1	$8 \times$ hLen
$G(.)$	MGF(., emLen – hLen – 1)
$H(.)$	MGF(., hLen)
r	*seed*
k	$8 \times$ emLen
s	*MaskedDB*
t	*MaskedSeed*

注意 9 文献 [3] ではビット単位の表記が使われており、RSA-OAEP ではバイト単位の表記が使われていることに注意を要する。

帰着の効率の観点から、RSA-OAEP の代わりに RSA-OAEP+ を推奨する意見もあったことも付記しておく。

> RSA-OAEP Encryption Scheme. RSA-OAEP has been proven to be semantically secure against adaptive chosen-ciphertext attacks in the random oracle model under the RSA assumption. However, the reduction is not tight, and thus it is not clear what security assurances the proof provides. We recommend that RSA-OAEP be modified to RSA-OAEP+ which has a tighter security reduction, and furthermore can be easily modified to allow encryption of arbitrarily-long messages (see [18]).

RSAES-PKCS1-v1_5(RSA 守秘) の使用について

現在時点 (2002 年 9 月末) までに発見されている RSAES-PKCS1-v1_5(RSA 守秘) に対する攻撃法の複雑度は、CJNP 攻撃 [9] については選択平文攻撃を許したとしても素因数分解の複雑度を下回ることはないことを確認したので、特に安全性の問題はないと考える。

ただし、RSA パラメータの選択方法には注意を要する。詳しくは、2.5.1.1 節を参照のこと。

選択暗号文攻撃では、Bleichenbacher, Manger による実装上の弱点をついた攻撃 [5, 17] があることにも注意を要するが、最新の仕様書ではその対策が施されているので問題ないと考える。

参考文献

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO'98*, pages 26–45. Springer, 1998. Lecture Notes in Computer Science No. 1462.

[2] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols,. In *Proc. of the First ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[3] M. Bellare and P. Rogaway. Optimal asymetric encryption — how to encrypt with RSA. In A.D. Santis, editor, *Advances in Cryptology — EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Berlin, Heidelberg, New York, 1995. Springer-Verlag.

[4] M. Bellare and P. Rogaway. The exact security of digital signatures –how to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Berlin, Heidelberg, New York, 1996. Springer-Verlag.

[5] D. Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS#1. In H. Krawczyk, editor *Advances in Cryptology — CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12, Berlin, Heidelberg, New York, 1998. Springer-Verlag. Springer-Verlag,

[6] D. Coppersmith. Finding a small root of a univariate modular equation. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 155–165, Berlin, Heidelberg, New York, 1996. Springer-Verlag.

[7] D. Coppersmith, M. K. Franklin, J. Patarin, M. K. Reiter. Low-exponent RSA with related messages. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 1–9, Berlin, Heidelberg, New York, 1996. Springer-Verlag.

[8] J. S. Coron. On the exact security of full domain hash. In M. Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235, Berlin, Heidelberg, New York, 2000. Springer-Verlag.

[9] J. S. Coron, M. Joye, D. Naccache and P. Paillier. New Attacks on PKCS# 1 v1.5 Encryption. In P. Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 369–379, Berlin, Heidelberg, New York, 2000. Springer-Verlag.

[10] J. S. Coron, D. Naccache, and T. P. Stern. On the Security of RSA Padding. In Michael J. Wiener, editor, *Advances in Cryptology — CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 1–18, Berlin, Heidelberg, New York, 1999. Springer-Verlag.

[11] J. S. Coron. Optimal Security Proofs for PSS and other signarure Schemes. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287, Berlin, Heidelberg, New York, 2002. Springer-Verlag.

[12] J. S. Coron. Security proof for partial-domain hash signature schemes. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 613–626, Berlin, Heidelberg, New York, 2002. Springer-Verlag.

[13] J. S. Coron and M. Joye and D. Naccache and P. Paillier, Universal Padding Schemes for RSA" In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 226–241, Berlin, Heidelberg, New York, 2002. Springer-Verlag.

[14] A. Fiat and A. Shamir. How to prove yourself. In A. M. Odlyzko, editor, *Advances in Cryptology — CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–208, Berlin, Heidelberg, New York, 1986. Springer-Verlag.

[15] E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern. RSA-OAEP is chosen-ciphertext secure under the RSA assumption. In J. Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274, Berlin, Heidelberg, New York, 2001. Springer-Verlag.

[16] Jakob Jonsson. Security proofs for the RSA-PSS signature schemes and its variants –draft 1.1. 2001. Available at `http://eprint.iacr.org/2001/053/`.

[17] J. Manger. A chosen ciphertext attack on rsa optimal asymmetric encryption padding (OAEP) as standardized in PKCS# v2.0. In J. Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 230–238, Berlin, Heidelberg, New York, 2001. Springer-Verlag.

[18] V. Shoup. OAEP reconsidered. In J. Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259, Berlin, Heidelberg, New York, 2001. Springer-Verlag.

[19] PKCS #1: RSA Encryption Standard, An RSA Laboratories Technical Note Version 1.5, Revised November 1, 1993, Available at `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html`

[20] PKCS #1 v2.0: RSA Cryptography Standard, RSA Laboratories, October 1, 1998, Available at `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html`

[21] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002, Available at `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html`

2.3.5 ECIES

2.3.5.1 評価対象暗号技術

- Certicom Research, Standards for Efficient Cryptography Group (SECG), September 20, 2000. Version 1.0.
  http://www.secg.org/

2.3.5.2 技術概要

ECIES は SECG (Standards for Efficient Crytography Group) によって策定された公開鍵暗号技術であり、楕円曲線を用いた暗号化方式である。 本暗号技術は、2000 年度は ECAES として CRYPTREC に応募されていたが、2001 年度には暗号技術名を ECIES に変更して応募されている。ただし、両者は同一の方式である。ECIES[5] は CRYPTREC 応募版であるが、証明可能安全性を有するものとして [1]、[2] および [3] に記載された方式がある。

2.3.5.3 技術仕様

ECIES の仕様 (概要) は以下の通りである。詳細については、仕様書を参照されたい。

楕円曲線パラメータ (i) (p, a, b, G, n, h) または (ii)$(m, f(x), a, b, G, n, h)$:
(i) 素体 $\mathbb{F}_p$ 上の楕円曲線 $E : y^2 = x^3 + ax + b(a, b \in \mathbb{F}_p)$、$E$ 上の素数位数 n の有理点 $G \in E(\mathbb{F}_p)$ 及び $h = \#E(\mathbb{F}_p)/n$ からなるパラメータ。
(ii) $\mathbb{F}_2$ 上の m 次既約多項式 $f(x)$ によって表される体 $\mathbb{F}_{2^m}$, $\mathbb{F}_{2^m}$ 上の楕円曲線 $E : y^2 + xy = x^3 + ax^2 + b(a, b \in \mathbb{F}_{2^m})$、$E$ 上の素数位数 n の有理点 $G \in E(\mathbb{F}_{2^m})$ 及び $h = \#E(\mathbb{F}_{2^m})/n$ からなるパラメータ
KDF: 鍵導出関数 (Key Derivation Function)
MAC: メッセージ認証子 (Message Authentication Code)
ENC: 共通鍵暗号 (Symmetric Key Encryption) もしくは排他的論理和 XOR
秘密鍵: 整数 d $(\in [1, n-1])$
公開鍵: G の d 倍点 Q

暗号化: 平文 M の入力に対して、暗号文 C の出力を以下の手順で行う。

1. ランダムに整数 $k \in [1, n-1]$ を選び、$R = kG$ を計算する
2. kQ の x 座標 $Z = x(kQ)$ を計算する
3. K = KDF(Z) を計算し、ENC で用いる鍵 EK 及び MAC で用いる鍵 MK を $K = (EK||MK)$ により計算する
4. 平文 M を鍵 EK を用いた ENC によって、暗号化:$EM = ENC(EK, M)$ (XOR のときは、$EM = EK \oplus M$) を行う
5. EM を鍵 MK を用いた MAC によって、認証子 $D = MAC(MK, EM)$ を計算する
6. 暗号文 $C = (R||EM||D)$ を出力する

復号: 暗号文 C の入力に対して、平文 M あるいは “invalid” を以下の手順で出力する。

1. $C = (R'||EM'||D')$ により、R', EM', D' を計算する
2. dR' の x 座標 $Z' = x(dR')$ を計算する
3. $K' = \mathrm{KDF}(Z')$ を計算し、ENC で用いる鍵 EK' 及び MAC で用いる鍵 MK' を $K' = (EK'||MK')$ により計算する
4. $D' = MAC(MK', EM')$ が成り立つかどうか検証する。もし成り立たなければ、“invalid” を出力する
5. EM' を鍵 EK' を用いた ENC によって復号:$M = ENC(EK', EM')$ (XOR のときは、$M = EK' \oplus EM'$) を行う
6. 平文 M を出力する

2.3.5.4 安全性評価

- **楕円曲線離散対数問題**

  ECIES は楕円曲線上の離散対数問題の困難性に依存した公開鍵暗号である。

  現在、楕円曲線上の離散対数問題に対して種々の攻撃法が知られているが、ECIES においては、実用上有効な各種ビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC 2 ドキュメントに具体的に示されている (ただし、これらは推奨パラメータであって他の楕円曲線の使用を禁じるものではない)。それらの楕円曲線は検証可能な形でランダムに選定された楕円曲線と Koblitz 曲線と呼ばれる楕円曲線 (2.3.2.4 節の「Koblitz 曲線の安全性」を参照) からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC 2 に含まれているが、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある (2.3.2.4 節の「Koblitz 曲線の安全性」を参照)。

- **ECIES[5] と証明可能安全性を有する ECIES[1, 2, 3] との違い**

  ECIES[5] の仕様は、その基になっている論文 [1]、[2] および [3] の記述にある方式と異なる部分が見受けられる。ECIES[5] と論文 [1]、[2] および [3] における方式との違いとしては、(i) KDF への入力データの違い、及び (ii) MK(MAC に用いる鍵) の取り方の違いが挙げられる。より正確には、次のように記述できる。(ただし、以下では楕円曲線上の有理点 R および Q から求められる Diffie-Hellman 共有鍵を $DH(R, Q)$ であらわすことにする。)

  ECIES[5] と ECIES [1, 2, 3] との違い

  (i) KDF への入力データ:

  KDF への入力データとして、ECIES[5] では $x(DH(R, Q))$ ($DH(R, Q)$ の x 座標) が用いられているのに対し、[1] では R と $DH(R, Q)$ が用いられている。また、[2] および [3] では $DH(R, Q)$ が用いられている。

  (ii) MK(MAC に用いる鍵) の取り方:

  EK および MK のとり方として、ECIES[5] では $K = (EK||MK)$ を用いているが、[1]、[2] および [3] では $K = (MK||EK)$ を用いている。

論文 [1]、[2] および [3] にある方式は、スタンダードモデル (standard model) のもとで、オラクル Diffie-Hellman 仮定 (Oracle Diffie-Hellman Assumptions) が正しく、また共通鍵暗号 ENC とメッセージ認証子 MAC が安全ならば、適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であることが証明されている。

一方、2002 年度詳細評価者 [6] によって、KDF の部分をランダム関数と仮定するランダムオラクルモデル (random oracle model) のもとで、Gap-Diffie-Hellman 仮定が正しく、また共通鍵暗号 ENC とメッセージ認証子 MAC が安全ならば、適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であることが証明されている。ただし、ここで証明可能安全性が示されている ECIES では、KDF への入力データとしては R と $x(DH(R, Q))$ が用いられており、EK および MK のとり方としては $K = (MK||EK)$ を用いている。

さらに、[9] によって、ジェネリックグループモデル (generic group model) のもとで、共通鍵暗号 ENC とメッセージ認証子 MAC が安全ならば、適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であることが証明されている。ただし、ここで証明可能安全性が示されている ECIES では、KDF への入力データとしては $DH(R, Q)$ が用いられており、EK および MK のとり方としては $K = (EK||MK)$ を用いている。しかしながら、[9] にも述べられているように、ECIES に対してジェネリックグループモデル (generic group model) で議論する妥当性については、注意が必要であると思われる。

- **ECIES の安全性**

  先に述べたように、ECIES[5] と、論文 [1][2][3] における方式において、(i) KDF への入力データ、及び (ii) MK(MAC に用いる鍵) の取り方に関しての相違点が存在する。この相違点は、[8] において指摘されているように、ECIES[5] の安全性に対して以下のような脆弱性をもたらしている。

  (i) KDF への入力データの相違点から生ずる脆弱性:

  KDF の入力に、Diffie-Hellman 共有鍵である楕円曲線上の有理点の x 座標のデータのみが用いられている。このため、適応的選択暗号文攻撃に対しての強秘匿 (IND-CCA2) の証明可能安全性が成り立たず、実際に適応的選択暗号文攻撃が可能になる (詳しくは、[8] を参照)。

  (ii) MK(MAC に用いる鍵) の取り方の相違点から生ずる脆弱性:

  共通鍵暗号として排他的論理和 (XOR) を用い、可変長の平文の使用を許すとき、MK(MAC の鍵) が平文の長さにも依存してしまう。このため、適応的選択暗号文攻撃に対しての強秘匿 (IND-CCA2) の証明可能安全性が成り立たず、実際に適応的選択暗号文攻撃が可能になる (詳しくは、[8] を参照)。

参考文献

[1] M. Abdalla, M. Bellare and P. Rogaway, “DHAES: An encryption scheme based on the Diffie-Hellman Problem”, submission to IEEE P1363a, September, 1998.

[2] M. Abdalla, M. Bellare and P. Rogaway, “The oracle Diffie-Hellman assumptions and an analysis of DHIES”, Topics in Cryptology, - CT-RSA 2001, LNCS 2020, pages 143-158, Springer-

Verlag, 2001.

[3] M. Abdalla, M. Bellare and P. Rogaway, “DHIES: An encryption scheme based on the Diffie-Hellman Problem”, full version of [4] [2], September, 2001. Available at `http://www-cse.ucsd.edu/users/mihir/`

[4] M. Bellare and P. Rogaway, “Minimizing the use of random oracles in authenticated encryption schemes”, Information and Communications Security, LNCS 1334, pages 1-16, Springer-Verlag, 1997.

[5] Certicom Research, Standards for Efficient Cryptography Group (SECG), September 20, 2000. Version 1.0. Available at `http://www.secg.org/`

[6] CRYPTREC Evaluation Reports, CRYPTREC, 2002.

[7] IEEE P1363a/D9 - standard specifications for public key cryptography: Additional techniques, June 2001. Draft version 9.

[8] V. Shoup, “A proposal for an ISO Standard for public key encryption”, Version 2.0, September 17, 2001. Version 2.1, December 20, 2001. Available at `http://shoup.net/papers/`

[9] N. P. Smart, “The exact security of ECIES in the generic group model”, Cryptography and Coding 2001, 2001.

2.3.6　HIME(R)

2.3.6.1　評価対象暗号技術

- “暗号技術仕様書　HIME(R) 暗号”, 2002 年度 CRYPTREC 応募書類.
  (株) 日立製作所,
  http://www.sdl.hitachi.co.jp/crypto/hime/index-j.html

2.3.6.2　技術概要

HIME(R) は、守秘を目的とした公開鍵暗号方式である。法として $N = p^d q$ を用いたモジュラー平方関数 (Rabin 暗号) をプリミティブとして、OAEP パディング [5] を利用してスキームを構成している。HIME(R) の設計方針は、素因数分解問題の計算量的困難性を仮定してランダムオラクルモデルでの証明可能安全性を備えること、暗号化および復号処理のスピードが速いこと、RSA-OAEP と同等以上の平文空間を持つことであると提案者は述べている。

2.3.6.3　技術仕様

HIME(R) の仕様 (概要) は以下のとおりである。

秘密鍵:　素数 p,q。但し、$p \equiv 3 \pmod 4, q \equiv 3 \pmod 4$。

公開鍵:　$N = p^d q$。但し、$d > 1$ なる整数で、2 もしくは 3 が推奨されている。$|N| = k$ とする。

補助関数:　ハッシュ関数 G, H。$G : \{0,1\}^{k_0} \to \{0,1\}^{k-k_0-1}, H : \{0,1\}^{k-k_0-1} \to \{0,1\}^{k_0}$。$|k_0| \geq 128$ を推奨。

暗号化:　メッセージ m を次のようにして暗号化する。

1. メッセージ $m \in \{0,1\}^n$ に対して、乱数 $r \in \{0,1\}^{k_0}$ を選び、

$$x = (m0^{k_1} \oplus G(r)) \| (r \oplus H(m0^{k_1} \oplus G(r)))$$

   を計算。但し、$n + k_1 = k - k_0 - 1, 2k_0 < k$。$|k_1| \geq 128$ を推奨。
2. $y = x^2 \bmod N$ を計算。y を暗号文とする。

復号:　暗号文 y から次のようにしてメッセージ m を復号する。

1. y が Z_N 上で平方剰余であることを検査する。平方剰余でない場合には y をリジェクトする。
2. $y = x^2 \bmod N$ を満たす 4 つの x_1, x_2, x_3, x_4 を求める。法 N が $p^d q$ 型であることと $p \equiv q \equiv 3 \pmod 4$ であることを利用した独自の手順が仕様書 [1] に規定されている。この手順は文献 [9] の内容と関連している。
3. $x_i (1 \leq i \leq 4)$ に対して $x_i \in \{0,1\}^{k-1}$ のとき、$x_i = s_i \| t_i, s_i \in \{0,1\}^{n+k_1}, t_i \in \{0,1\}^{k_0}$ なる (s_i, t_i) を求める。
4. $r_i = H(s_i) \oplus t_i, w_i = s_i \oplus G(r_i)$ なる $w_i (1 \leq i \leq 4)$ を求める。

5. $w_i = m_i||z_i, m_i \in \{0,1\}^n, z_i \in \{0,1\}^{k_1} (1 \leq i \leq 4)$ を求め、$z_i = 0^{k_1}$ が成立する m_i を復号文とする。そのような $m_i (1 \leq i \leq 4)$ が存在しない場合には、リジェクトを出力する。

2.3.6.4 仕様記述

HIME(R) の仕様書には不備・曖昧さがあり、現時点で信頼に足る HIME(R) の仕様書が公に手に入る状況になっていない。このため、HIME(R) の第三者による実装性や相互接続性が担保されない。

HIME(R) 仕様書 [1] の不備・不明点は以下のとおりである。

- 復号手順において定義されていない記号 x_0 がある。
- 復号手順において複数の復号文候補が得られる可能性がある。このようになる確率は無視できるが、仕様書ではこのことに触れるべきである。
- OAEP パディング後のメッセージは上位 1 ビットを 0 に固定する。仕様書の 3 章に記述されているパラメータ $k = 1344, k_0 = 128, k_1 = 128$ での実装では、このことが考慮されておらず、$n = 1087$ となるべきところが $n = 1088$ とされている。また、3 章の復号手順において、暗号文の平方根を求めた結果の最上位ビットが 0 であことのチェックをすべきである。
- ハッシュ関数 G と H の具体的な仕様が仕様書の 3.2 節に与えられているが、最も単純な構成である入力とカウンタ値を連結してハッシュする手法とは異なるものが採用されている。そのように決定したセキュリティ上の根拠を説明すべきである。また、定数 $C_i (1 \leq i \leq 10)$ において ‘128’ は下付き (LSB 側からとる) ではなく上付き (MSB 側からとる) の間違いである。同様に定数 C の ‘64’ は下付きの間違いである。
- 仕様書の 3.4 節と 3.6 節では OAEP パディング用の乱数 r の生成法として、最初に 192 ビットの乱数 R を定め、ハッシュ結果 SHA-1(R) の上位 128 ビットをとることで乱数 r を定めている。このように定めている根拠が不明である。
- 法 $N (= p^d q)$ を ‘強い素数 (strong prime)’ 条件を満たす素数 p, q から構成している。この条件を課すことでセキュリティ上の問題をもたらすことはないが、この条件を課す必要性もないという意見がある。

2.3.6.5 安全性評価

証明可能安全性

HIME(R) 自己評価書 [2] には、HIME(R) はランダムオラクルモデルの下で適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であることの安全性証明が記載されている。しかし、この安全性証明の幾つかの箇所に問題があり、不完全である。このことは複数の外部評価者も指摘している。

自己評価書における安全性証明の問題点は以下のとおりである。

- 自己評価書の補題 2.1 での確率評価に問題があり、(2) 式が誤っている。HIME(R) のプリミティブでは 1 つの暗号文に 4 つの平文が対応することが考慮されていない。また、(2) 式の導出に $s \neq s'$(s' は復号オラクルへの質問 $y' = (s'||t')^2 \bmod N$ での s'、一方 s は HIME(R) 攻撃者の攻撃対象である暗号文 $y = (s||t)^2 \bmod N$ での s) という事象が利用されているが、これが成立する理由が不明である。
- 復号オラクル・シミュレータの実行時間の評価式導出の根拠が説明されていないため、理解しにくい。
- ハッシュオラクル G と H のシミュレータは、同じ質問に対しては同じ答えを返す必要がある。このため、質問に対する動作として最初に過去の質問リストをチェックするはずである。これらが記述されていない。
- ターゲット暗号文 y に対し、$y = (s||t)^2 \bmod N$ なる s を H オラクルに質問した場合、法 N の素因数分解マシン M は直ちに停止し、H オラクルの質問リストに s が記載される事象は生じない。一方で、マシン M の成功確率は H オラクルの質問リストに上記の s が記載される事象で評価されているため、マシン M の動作の説明を修正しないと議論がかみ合わない。
- 復号オラクルがシミュレートに失敗した場合に返す記号*の説明がない。その他にも記述上のミスは数箇所にある。
- 自己評価書の証明が読みにくいという指摘が多数の外部評価者から寄せられた。

一方で、ある評価者は HIME(R) がランダムオラクルモデルにおいて IND-CCA2 であることの証明を独自に構成している。また、他の評価者も証明の修正は可能であり、HIME(R) は証明可能安全性を備えていると予想している。

安全性証明を独自に構成した評価者は次の 2 つの結果を与えている。

1. 自己評価書の証明手法に沿って問題箇所である補題 2.1 を修正した。さらに、他の部分が正しいものとして最終的に求められる IND-CCA2 攻撃者と法 N の素因数分解の間の実行時間と成功確率の評価式を修正した。
2. Shoup が RSA-OAEP+ の安全性証明において利用した証明技法 [6] を用いて安全性証明を再構築した。さらに、復号オラクルシミュレータを、Coppersmith アルゴリズム [3] をより積極的に利用した構成に変更し、法 N の素因数分解マシンの実行時間を改善した。この結果、$q_G q_H$ 回の HIME(R) 暗号化ステップを必要としていたものを $q_D q_H$ 回の Coppersmith アルゴリズムの実行ステップに変更した。

ただし、この評価者による安全性証明は外部評価レポートにおいてはじめて示されたものであり、その妥当性が多くの研究者により精査された状況とはいえないことに注意が必要である。従って、2002 年 9 月時点で HIME(R) が証明可能安全性を有するとは断定できなかった。

推奨パラメータサイズ

自己評価書では、RSA 型合成数 (pq 型) と $p^d q$ 型合成数のそれぞれに対する素因数分解の計算量が同等となるモジュラスのサイズを求めている。素因数分解アルゴリズムとして、楕円曲線

法と数体ふるい法のうち、指定したパラメータサイズの下で計算量のより少ないアルゴリズムでの計算量を評価している。その結果、RSA でのモジュラスが 1024 ビット、2048 ビット、4096 ビットのそれぞれに対して、HIME(R) のモジュラスは $d = 2$ の場合に 1344 ビット、2304 ビット、4032 ビットを、また $d = 3$ の場合に 1536 ビット、3072 ビット、4928 ビットをそれぞれ推奨している。こうしたモジュラスサイズの推奨値は妥当と考えられる。

なお、HIME(R) のモジュラスが p^2q 型 ($d = 2$) の場合に RSA 4096 ビットに相当するモジュラスサイズが 4032 ビットと短くなっているが、これは 1 ワードを 32 ビットとした場合の実装面の都合から算出された推奨サイズであって、一般にモジュラスサイズが 4096 ビット以上の場合に、p^2q 型の方が pq 型よりも短いサイズで同等の安全性が確保されるという意味ではない。

2.3.6.6 類似方式との比較

素因数分解をベースとする方式で守秘のカテゴリーに属する類似の方式との間で HIME(R) の特徴を比較した。比較対象としたのは、RSA-OAEP[5], RSA-OAEP+[6], Rabin-SAEP[8], Rabin-SAEP+[8] の各方式である。

下表において、k はモジュラスのサイズ、q_G, q_H, q_D はそれぞれハッシュオラクル G, ハッシュオラクル H, 復号オラクル D への質問回数である。また、暗号化、復号、平文長、暗号文長は RSA1024 ビット相当の安全性を持つパラメータにおいて比較した。暗号化処理と復号処理は 1024 ビットの乗算を mul とし、その回数に換算して見積った。ベース問題とはそれぞれの方式が安全性の根拠とする問題であり、Fact(p^2q) とは p^2q 型整数の素因数分解、RSA(pq, e) は公開鍵を $N(= pq)$, e とする RSA 問題をそれぞれ示している。

方式	ベース問題	帰着効率	暗号化	復号	平文長	暗号文長
HIME(R)	Fact(p^2q)	$q_Dq_H \times O(k^3)$	1.7mul	275mul	1087 ビット	1344 ビット
RSA-OAEP	RSA(pq, e)	$2q_Gq_H \times O(k^3)$	17mul	385mul	767 ビット	1024 ビット
RSA-OAEP+	RSA(pq, e)	$q_Gq_H \times O(k^3)$	17mul	385mul	767 ビット	1024 ビット
Rabin-SAEP	Fact(pq)	$q_Gq_H \times O(k^3)$	1mul	385mul	256 ビット	1024 ビット
Rabin-SAEP+	Fact(pq)	$q_H \times O(k^3)$	1mul	385mul	224 ビット	1024 ビット

- ベースとする問題への帰着効率の面からは、Rabin-SAEP+ が最も良く、他はほぼ同等である。ただし、HIME(R) の証明可能安全性が十分に認められた場合を仮定している。
- 暗号化処理時間はほぼ同等である。復号処理時間は HIME(R) が最も短い。
- 平文サイズは HIME(R) が最も大きくとれるのに対し、Rabin-SAEP と Rabin-SAEP+ は最も小さい。ただし、HIME(R) と RSA-OAEP および RSA-OAEP+ の差はモジュラスサイズの差によるものであって本質的とは考えられない。
- 暗号文サイズは HIME(R) が最も大きく、効率が悪い。

以上のように HIME(R) が優れる項目もあるが、劣る項目もある。他の類似方式に比べて HIME(R) が顕著な優位性をもつとは考えられない。

参考文献

[1] (株) 日立製作所, “暗号技術仕様書　HIME(R) 暗号”, 2002 年度 CRYPTREC 応募書類.
Available at http://www.sdl.hitachi.co.jp/crypto/hime/index-j.html

[2] (株) 日立製作所, “自己評価書　HIME(R) 暗号”, 2002 年度 CRYPTREC 応募書類.
Available at http://www.sdl.hitachi.co.jp/crypto/hime/index-j.html

[3] D.Coppersmith, “Finding a Small Root of a Univariate Modular Equation”, Advances in Cryptology – EUROCRYPT’96, Lecture Notes in Computer Science **1070**, Springer-Verlag, pp.155–165, 1996.

[4] D.Coppersmith, “Modifications to the Number Field Sieve”, Journal of Cryptology, Vol.6, No.3, pp.169-180, 1993.

[5] M.Bellare and P.Rogaway, “Optimal Asymmetric Encryption – How to Encrypt with RSA”, Advances in Cryptology – EUROCRYPT’94, Lecture Notes in Computer Science **950**, Springer-Verlag, pp.92–111, 1995.

[6] V.Shoup, “OAEP Reconsidered”, Advances in Cryptology – CRYPTO2001, Lecture Notes in Computer Science **2139**, Springer-Verlag, pp.239–259, 2001.

[7] E.Fujisaki, T.Okamoto, D.Pointcheval, and J.Stern, “RSA-OAEP is Secure under the RSA Assumption”, Advances in Cryptology – CRYPTO2001, Lecture Notes in Computer Science **2139**, Springer-Verlag, pp.260–274, 2001.

[8] D.Boneh, “Simplified OAEP for the RSA and Rabin Functions”, Advances in Cryptology – CRYPTO2001, Lecture Notes in Computer Science **2139**, Springer-Verlag, pp.275–291, 2001.

[9] T.Takagi, “Fast RSA-type Cryptosystem Modulo $p^k q$”, Advances in Cryptology – CRYPTO’98, Lecture Notes in Computer Science **1462**, Springer-Verlag, pp.318–326, 19981.

2.3.7 ECDH

2.3.7.1 評価対象暗号技術

- SEC 1: Elliptic Curve Cryptography,
  Certicom Research, standards for efficient cryptography group (SECG), September 20, 2000.
  Version 1.0.
  `http://www.secg.org/secg_docs.htm`

2.3.7.2 技術概要

ECDH は SECG (Standards for Efficient Crytography Group) によって策定された公開鍵暗号技術であり、楕円曲線を用いた鍵共有方式である。 本暗号技術は、2000 年度は ECDHS として CRYPTREC に応募されていたが、2001 年度には暗号技術名を ECDH に変更して応募されている。ただし、両者は同一の方式である。

2.3.7.3 技術仕様

ECDH の仕様の概要は以下の通りである。詳細については、仕様書を参照されたい。

楕円曲線パラメータ (i) (p, a, b, G, n, h) または (ii)$(m, f(x), a, b, G, n, h)$:
(i) 素体 $\mathbb{F}_p$ 上の楕円曲線 $E : y^2 = x^3 + ax + b (a, b \in \mathbb{F}_p)$ とその素数位数 n の有理点 $G \in E(\mathbb{F}_p)$ 及び $h = \#E(\mathbb{F}_p)/n$ からなるパラメータ。
(ii) $\mathbb{F}_2$ 上の m 次既約多項式 $f(x)$ によって定義される体 $\mathbb{F}_{2^m}$ 上の楕円曲線 $E : y^2 + xy = x^3 + ax^2 + b (a, b \in \mathbb{F}_{2^m})$ とその素数位数 n の有理点 $G \in E(\mathbb{F}_{2^m})$ 及び $h = \#E(\mathbb{F}_{2^m})/n$ からなるパラメータ
KDF: 鍵導出関数 (Key Derivation Function)

初期設定:

1. 利用者 U と V は鍵導出関数 (KDF) 及び楕円曲線パラメータ $((p, a, b, G, n, h)$ あるいは $(m, f(x), a, b, G, n, h))$ を決める。
2. 利用者 U と V は上記の楕円曲線パラメータに属する、秘密鍵 d_U, d_V と公開鍵 Q_U, Q_V をそれぞれ生成する:
   - 利用者 U: ランダムに整数 $d_U \in [1, n-1]$ を選び、$Q_U = d_U G$ を計算する
   - 利用者 V: ランダムに整数 $d_V \in [1, n-1]$ を選び、$Q_V = d_V G$ を計算する

鍵共有: 利用者 U と V は共有鍵情報 K を以下のようにして生成する:

- 利用者 U: $d_U Q_V$ の x 座標 $Z = x(d_U Q_V)$ を計算し、$K = \mathrm{KDF}(Z)$ を求める
- 利用者 V: $d_V Q_U$ の x 座標 $Z = x(d_V Q_U)$ を計算し、$K = \mathrm{KDF}(Z)$ を求める

2.3.7.4　安全性評価

ECDH の安全性は楕円曲線上の離散対数問題に依存している。現在、楕円曲線上の離散対数問題に対して種々の攻撃法が知られているが、ECDH においては、実用上有効な各種ビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC 2 ドキュメントに具体的に示されている (ただし、これらは推奨パラメータであって他の楕円曲線の使用を禁じるものではない)。それらの楕円曲線は検証可能な形でランダムに選定された楕円曲線と Koblitz 曲線と呼ばれる楕円曲線 (2.3.2.4 節の「Koblitz 曲線の安全性」を参照) からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC 2 に含まれているが、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある (2.3.2.4 節の「Koblitz 曲線の安全性」を参照)。

ECDH は楕円曲線を利用した Diffie-Hellman 鍵共有方式であり、鍵共有方式として最も基本的なものである。受動的攻撃に対して、大きな問題点は指摘されていない。しかし、能動的な攻撃に対しては安全でなく、最低限以下の 2 点に注意を払う必要である。

- 中間者攻撃を防ぐために、電子署名を用いるなどして公開鍵と利用者との結びつきを保証する。
- 利用者が同一の公開鍵を用いる限り、共有される鍵情報は同一である。ECDH を (更新を前提とする) セッション鍵共有方式として使用する場合は、交換する公開鍵は一時的なものとする。

ちなみに、ECDH プリミティブを用い、電子署名と組み合わせ、能動的な攻撃者に対して証明可能な安全性をもち、forward-secrecy を実現する鍵共有方式が複数の詳細評価者によって提案されている [2]。

参考文献

[1] Certicom Research, standards for efficient cryptography group (SECG), September 20, 2000. Version 1.0.
Available at `http://www.secg.org/secg_docs.htm`

[2] 2000 年度 ECDHS 詳細評価報告書、評価者#2,#3

2.3.8 DH

2.3.8.1 評価対象暗号技術

- W. Diffie and M. E. Hellman, “New directions in cryptography”[1]
  で発表された基本的なスキームを評価した。

- ANSI X9.42-2001,
  “Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography”,
  American National Standards Institute, 2001.

2.3.8.2 技術概要

DH は 1976 年に W.Diffie と M.E.Hellman により提案された、鍵共有機能を実現する公開鍵暗号技術である [1]。

2.3.8.3 技術仕様

ここではそれらに共通の基本的な処理 (秘密情報を共有する部分) について述べるにとどめる。ドキュメントではこの処理に基づいた 6 つのスキーム (dhStatic, dhEphem, dhOneFlow, dhHybrid1, dhHybrid2, dhHybridOneFlow) が定義されている。それらのスキームにおいて、最終的に共有される鍵は鍵導出関数 (KDF) によって生成されるようになっている。ドキュメント内では 2 種類の鍵導出関数が定義されており、実際にはどちらかを選んで使用することになる。したがってスキームと KDF の組み合わせにより、合計 12 種類の鍵共有法がこのドキュメントにおいて定義されている。詳細については参照ドキュメントを御覧いただきたい。実際に 12 種類のどれを利用するかを決定する際には、安全性評価の記述を参考に、使用環境において安全性を損なわないものを選択すべきである。参照ドキュメントの付録 E.3 には選択のガイドラインが述べられている。そちらも参考にしていただきたい。なお各パラメータについて、ドキュメント記載の妥当性確認処理を行なうことが求められている事実にも注意すべきである。

システム共通パラメータの設定

入力:	L, m, n	セキュリティパラメータ ある整数 $n \geq 4$ に対して $L = 256n$、$m \geq 160$
出力:	(p, q, g)	システム共通パラメータ

処理手順:

1. $2^{(L-1)} < p < 2^L$ なる素数 p、$p-1$ の約数で $2^{(m-1)} < q < 2^m$ を満たす素数 q を生成する。
2. $g \in \mathbb{Z}_p^*$ を位数 q の原始元とする。

ユーザの初期設定

入力:	(p, q, g)	システム共通パラメータ
出力:	x	ユーザの秘密鍵 (整数)
	y	ユーザの公開鍵 ($y \in \mathbb{Z}_p^*$)

処理手順:

1. ランダムに $1 \leq x \leq q-1$ なる整数 x を生成する。
2. $y = g^x \bmod p$ を計算する。

秘密情報共有の処理

秘密情報共有を開始するユーザ U は以下の処理を行なう。その相手であるユーザ V も同様の (以下の U, V を入れ換えた) 処理を行なう。

入力:	(p, q, g)	システム共通パラメータ
	x_U	ユーザ U の秘密鍵 (整数)
	y_V	ユーザ V の公開鍵 ($y_V \in \mathbb{Z}_p^*$)
出力:	"Failure" あるいは Z	

処理手順:

1. $Z = y_V^{x_U} \bmod p$ を計算する。
2. もし $Z = 1$ なら "Failure" そうでなければ Z を出力する。

2.3.8.4　安全性評価

a) 前節で述べたスキームは、非常に単純な基本形である。Diffie-Hellman 方式には、プロトコルに多くのバリエーションが存在するので、個々のプロコトル毎の評価が必要である (参考: 実使用されているプロトコルの例: RFC2631, ISO/IEC 11770-3, Oakley, PGP)。評価対象は、基本的なスキームのみである。

b) 秘密共有プロトコルとしての基本的な部分は、受動的攻撃のみを仮定した場合、Diffie-Hellman 問題に帰着される。ただし、ランダムなビット列と区別できないという意味においては、範囲を制限するなどの工夫によって Decisional Diffie-Hellman 問題に帰着される。

c) 共有秘密情報をセッション鍵として使用するスキームにおいては、安全性を左右する様々な要因がある。これらの要因の組み合わせは膨大な数になり、すべての組み合わせに関して網羅的な安全性評価を行うことは困難である。考慮すべき要因としては例えば以下のものが考えられる。

   1) 鍵対が固定なものか、一時的なものか (static/ephemeral)。
   2) 公開鍵とエンティティとの対応が保証されているか否か (nocert/cert)。更にエンティティが対応する秘密鍵を持っていることまで保証されているか否か (strongcert)。
   3) 公開鍵の交換時に公開鍵に署名をするか否か (unsigned/signed)。

d) 共有秘密をセッション鍵として使用するスキームにおいて、前節で述べた形のまま使用することは、次項で述べるような問題が考えられるため、使用に際しては、最低限「鍵とエンティティとの結びつきを保証する手段を備え、またセッション鍵として使用する場合、交換する公開鍵は一時的なものとする」ことが必要である。

e) 問題となる組み合わせ、具体的な攻撃法の例を以下にあげる。

1) 両者の鍵が固定の場合 (static)

Fixed-session-key attack: セッション鍵が固定となるため、counter mode で使用している場合、同じ Vernam pad を毎セッション用いることにより、秘密が露呈する。

2) 秘密鍵と公開鍵との結びつきに保証がない場合 (not strongcert)

Unknown key-share attacks:攻撃者が、各ユーザーの公開鍵を自分の公開鍵と偽ることにより、各ユーザーの間にはいり、あたかも自分が交信しているかのように見せかける。

3) その他

Captured session key attacks: 少なくともいずれか一方が固定鍵の場合、一旦セッション鍵がもれると、その後、同じセッション鍵を使い続けられる。

Key-translate attacks: nocert/unsigned の場合、鍵を α 倍することにより、異なる鍵を共有させる。

Reveal attacks: public な WS などでの操作で、secret coin(秘密にしておくべき情報) が漏れた場合、その他の秘密情報に影響を及ぼす (forward secrecy の欠如)。

Attacks intrinsic 2-flow AKE(Authenticated Key-Exchange protocols): 2 つしか flow がなく、2 つめの flow が 1 つ目の flow と独立な場合には、strong-corruption model で forward secrecy がない、A-to-B/B-to-A authentication がないなどの問題がある。

f) 公開鍵に対する署名を組み合わせて使うなどの改良を加えることにより、解決される問題もある。

参考文献

[1] W. Diffie and M. E. Hellman, “New directions in cryptography”, IEEE Trans. Information Theory, vol. IT-22, pp.644-654, 1976.

[2] ANSI X9.42-2001, “Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography”, American National Standards Institute, 2001.

2.3.9 PSEC-KEM

2.3.9.1 評価対象暗号技術

- “PSEC-KEM 仕様書”(2002 年 5 月 14 日), CRYPTREC 応募書類
  NTT 情報流通プラットフォーム研究所,
  http://info.isl.ntt.co.jp/psec/index-j.html

2.3.9.2 技術概要

PSEC-KEM は、楕円曲線上の離散対数問題をベースとする鍵カプセル化メカニズム (Key Encapsulation Mechanisms:KEM) である。KEM は、守秘目的の公開鍵暗号を実現する一方法であるハイブリッド暗号を構成する際に、その構成要素として利用できるものである。なお、上記のハイブリッド暗号は、Victor Shoup 氏が ISO/IEC JTC 1/SC27/WG2 で提案したアルゴリズム [6] が元になって研究が進められており、

- 公開鍵暗号ベースの証明可能安全性を有する KEM
- 共通鍵暗号ベースの証明可能安全性を有する Data Encapsulation Mechanism (DEM)

の組み合わせからなる。そのため、この構成を以下では KEM-DEM 構成と呼ぶことにする。KEM-DEM 構成は、KEM と DEM がそれぞれで定義される証明可能安全性を満たせば、組み合わせた時の証明可能安全性が満たされ、公開鍵暗号と共通鍵暗号の独立性 (フレキシビリティ) が高い方法として注目されている。

CRYPTREC は PSEC-KEM の利用にあたり SECG[*11] (Standards for Efficient Cryptography Group) における楕円曲線パラメータ選択法を利用して楕円曲線パラメータを選択することを推奨する。楕円曲線パラメータの選択については 2.5.3 節を参照して欲しい。

PSEC-KEM は、2001 年度に、守秘目的の PSEC-2(CRYPTREC2000 への応募暗号) の改訂版として鍵交換のカテゴリで応募されたが、公開鍵暗号評価小委員会での検討の結果、新規応募として扱うことになった。

2.3.9.3 技術仕様

PSEC-KEM の仕様 (概要) は以下の通りである。詳細については、仕様書を参照されたい。

PSEC-KEM は、以下に述べる鍵生成アルゴリズム G, 暗号化アルゴリズム E 及び復号アルゴリズム D からなる。

鍵生成アルゴリズム G

[*11] http://www.secg.org/

PSEC-KEM では楕円曲線上の演算を行うため、まず、適切な楕円曲線及びベースポイントを SECG により策定された SEC1 により決める。

楕円曲線 E 及びベースポイント P を SEC1 で示される方法で決める。その後、以下の処理を行って公開鍵 PK 及び秘密鍵 SK を出力する。

1. 乱数 $s \in \{0, \cdots, p-1\}$ を生成
   ここで、p はベースポイント P の位数とする
2. $W = sP$ を計算
3. 適切な鍵導出関数を選択
   ここで、鍵導出関数とは、任意長のハッシュ関数値を出力する関数であり、仕様書では、ハッシュ関数 $SHA-1$ を利用して構成したマスク生成関数 MGF(Mask Generation Function)[5] を推奨している。
4. 適切なビット長 $hLen$ を選択
5. 公開鍵 PK を $PK = (E, W, MGF, hLen)$ として出力
6. 秘密鍵 SK を $SK = (s, PK)$ として出力

なお、仕様書によれば、p のビット長は 160、$hLen$ は 160 を推奨している。

暗号化アルゴリズム E

暗号化アルゴリズム E は、PK を入力とし、暗号文 C 及び $keyLen$ ビットの共有鍵 K を出力する。

1. $hLen$ ビットの乱数 r を生成
2. $pLen + 128 + keyLen$ ビットの $G = MGF(0_{32} \| r)$ を生成
   ここで、0_{32} は、0 を 32 ビット表現したときのビット列である。
3. G を $G = t \| K$ として、G を $pLen + 128$ ビットの t と $keyLen$ ビットの K に分割
4. $\alpha = t \bmod p$ を計算
5. $Q = \alpha W, C_1 = \alpha P$ を計算
6. $hLen$ ビットの $H = MGF(1_{32} \| C_1 \| Q)$ を生成
   ここで、1_{32} は、1 を 32 ビット表現したときのビット列である。
7. $hLen$ ビットの $C_2 = r \oplus H$ を生成
8. 暗号文 $C = (C_1, C_2)$、及び共有鍵 K を出力

なお、仕様書によれば、$keyLen$ は 256 を推奨している。

復号アルゴリズム D

暗号アルゴリズム D は、SK 及び暗号文 C を入力とし、$keyLen$ ビットの共有鍵 K(もしくは”invalid”) を出力する。

1. 暗号文 C を $C = (C_1, C_2)$ に分割

2. $Q = sC_1$ を計算
3. $hLen$ ビットの $H = MGF(1_{32}||C_1||Q)$ を生成
   ここで、1_{32} は、1 を 32 ビット表現したときのビット列である。
4. $hLen$ ビットの $C_2 = r \oplus H$ を生成
5. $pLen + 128 + keyLen$ ビットの $G = MGF(0_{32}||r)$ を生成
   ここで、0_{32} は、0 を 32 ビット表現したときのビット列である。
6. G を $G = t||K$ として、G を $pLen + 128$ ビットの t と $keyLen$ ビットの K に分割
7. $\alpha = t \bmod p$ を計算
8. $C_1 = \alpha P$ が成立するかどうか検証
9. 成立すれば共有鍵 K を出力　(成立しなければ “invalid” を出力)

2.3.9.4　安全性評価

KEM の安全性定義

KEM の IND-CCA2(適応的選択暗号文攻撃に対して強秘匿) の定義は、公開鍵暗号の IND-CCA2 の定義と若干異なる。KEM は次の場合に、IND-CCA2 であると定義される。

「いかなる (多項式時間の) 攻撃者 A も、KEM の暗号文 C とビット列 K が与えられたとき、復号オラクルを用いて、ビット列 K が KEM により得られる正しい共有鍵 Key か、あるいはランダムなものかを有意な確率で判別できない」

上記の定義は、復号オラクルを用いる攻撃者が、多項式時間で、暗号文から、共有鍵の部分情報を解読することができないことを意味する。

PSEC-KEM の安全性評価

PSEC-KEM は、ランダムオラクルモデルにおいて、EC-CDH の困難性の仮定のもと、上記に述べた KEM としての IND-CCA2 の証明可能性を有する。

定理 10 PSEC-KEM に対する IND-CCA2 における攻撃者を $\mathcal{A}$ とする。$\mathcal{A}$ の攻撃成功確率を ε、攻撃時間を t、復号オラクル、ランダムオラクル G, H に対する質問回数を、それぞれ q_D 回、q_G 回、q_H 回とする。このとき、EC-CDH 問題は、次式を満たす成功確率 ε'、時間 t' で解ける。

$$\varepsilon' \geq \frac{\varepsilon}{2(1+2^{-128})} - \frac{(q_G + 3q_D)(1+2^{-128})}{p} - \frac{q_D + q_G}{2^{hLen}}$$
$$t' \leq t + q_H \times (T + \mathrm{O}(1))$$

ここで、T は楕円曲線上の乗算 2 回を行う計算時間である。

この定理では、ランダムオラクルモデルの仮定の下で、PSEC-KEM の IND-CCA2 を破る攻撃

者が存在すれば、その攻撃者は EC-CDH を計算できることを示している。また、この定理は、PSEC-KEM の IND-CCA2 を破る攻撃者を仮定して、実際に EC-CDH 問題を計算する多項式時間アルゴリズムを構成し、その成功確率を求めることにより、証明される。

2.3.9.5 KEM 概要

KEM を用いた守秘用ハイブリッド暗号

KEM は、別途定義される DEM と組み合わせることにより、IND-CCA2 の証明可能安全性を有する守秘目的のハイブリッド暗号を構成する (KEM-DEM 構成)。KEM-DEM 構成では、まず、KEM により、公開鍵暗号ベースでセッション鍵を送受信者双方で共有する。一方、DEM は共通鍵暗号部 (SYM) と通信文の安全性を保障するための認証子生成部 (MAC) からなる。送信側は、平文を SYM においてセッション鍵で暗号化し、また、平文に対して MAC で認証子を生成し、双方を結合して暗号文として伝送する。受信側は、暗号文を分割し、まず SYM においてセッション鍵で復号し復号文を求める。次に、復号文に対して MAC で認証子を生成して、送信側から送信されたものと比較して両者が一致していれば、復号文を出力する。

KEM-DEM 構成は、他の IND-CCA2 を満たす方法に比べ、長い平文を効率的に暗号化でき、また証明可能安全性を満たす KEM と DEM を独立に構成することにより、高いフレキシビリティを有する。

KEM、および DEM を構成する SYM と MAC の証明可能安全性のインフォーマルな定義は次の通りである。

- KEM : いかなる (多項式時間の) 攻撃者 $\mathcal{A}$ も、KEM の暗号文 C とビット列 K が与えられたとき、復号オラクルを用いて、ビット列 K が KEM により得られる正しい共有鍵 Key か、あるいはランダムなものかを有意な確率で判別できない
- SYM : 攻撃者 $\mathcal{A}$ が選んだ 2 つのメッセージのいずれかに対応した暗号文が得られたとき、その暗号文からもとのメッセージのいずれであるかを有意な確率で判別できない
- MAC : 使い捨ての鍵で生成される One-time の認証子生成関数である

なお、DEM、つまり SYM と MAC としてどのようなものを推奨するのかについては、標準化の次のステップとして挙げられており [6]、今後の課題である。

また、最近 NESSIE にて、KEM-DEM 構成の安全性に関する論文 [2] が発表された。論文では、KEM-DEM 構成は、DEM が KEM に依存するという特殊な状況を仮定すると、安全性の問題が発生する可能性を指摘しており、DEM に求められる安全性条件を再定義している。

KEM を用いたその他の応用

[3] において、KEM で共有したセッション鍵を、守秘以外の例えば鍵配送や認証、鍵共有にも使用できる可能性が示唆されている。ただし、現時点では、守秘以外の暗号スキームにおける安全性証明のモデルが定まっていない。

提案されている KEM スキーム

- ACE-KEM:
  ACE-KEM の安全性は、Universal one-way hash 関数の仮定の下で、EC-DDH への帰着が証明されている。
- ECIES-KEM:
  ECIES-KEM の安全性は、ランダムオラクルモデルの下で、EC-gap-DH への帰着が証明されている。
- PSEC-KEM:
  PSEC-KEM の安全性は、ランダムオラクルモデルの下で、EC-CDH への帰着が証明されている。
- RSA-KEM:
  RSA-KEM の安全性は、ランダムオラクルモデルの下で、RSA-oneway への帰着が証明されている。

これらを比較すると、安全性の面ではモデルや帰着する問題の安全性の関係がまだ未解決であるため、どの方法が最も優れているかは現時点では言及できない。また、演算時間の点では、ECIES-KEM と PSEC-KEM が処理が速い。

2.3.9.6　まとめ

PSEC-KEM は、KEM としての証明可能安全性を有し、別途定義される証明可能安全性を有する DEM と組み合わせて KEM-DEM 構成における利用を前提として、実用上安全である。

参考文献

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, “Relations Among Notions of Security for Public-Key Encryption Schemes,”Proc. of Crypto’98, Springer-Verlag, LNCS 1462, pp.26-45 (1998).

[2] L. Granboulan, “RSA hybrid encryption schemes,”
Available at https://www.cosic.esat.kuleuven.ac.be/nessie/reports/phase2/rsaenc-pub.pdf.

[3] B. Kaliski, “Key Encapsulation: An Emerging Paradigm for Public-key Cryptoraphy,” RSA-Conference-2002-Japan (2002.5.29-30) Conference Note, 2002.

[4] NTT 情報流通プラットフォーム研究所, “PSEC-KEM 仕様書”(2002 年 5 月 14 日), CRYPTREC 応募書類
Available at http://info.isl.ntt.co.jp/psec/index-j.html

[5] RSA Laboratories, “PKCS#1 v2.1: RSA Encryption Standard, ”
June 14, 2002.

[6] V. Shoup : “A Proposal for an ISO Standard for Public Key Encryption,”
Available at http://shoup.net/papers/iso-2.pdf.

2.4 数論的問題の困難さに関する評価

2.4.1 素因数分解問題

有理整数の素因数分解問題の困難さに依拠して暗号プリミティブの安全性を主張する暗号スキームを横断的に評価するため、素因数分解問題の現状に関する詳細評価を 2001 年度に実施した。ここでは、それらの評価結果を踏まえて、現在最も有力な素因数分解アルゴリズム、現実的に安全と考えられる合成数のサイズとその将来予想などを総括する。また、問題そのものの難しさに影響を与えうる事項に関しても最後に補足する。

2.4.1.1 有力なアルゴリズム

暗号プリミティブの評価を主眼として素因数分解問題を考えるため、ここでは分解の対象とする合成数を $n = p^r q$ (p, q: 素数, $p < q, r \geq 1$) とする。$r = 1$ のときは RSA 等で使われる合成数となる。

素因数分解問題を解くアルゴリズムはいくつか知られているが、それらは実行時間の関数を決める主要因に応じて 2 種類に大別される。ひとつは、 n に含まれる最小素因数のサイズ $|p|$ に依存して実行時間が決まるものであり、 他方は n のサイズ $|n|$ だけに依存して実行時間が決まるものである。前者に属するアルゴリズムのうち最高速のものは楕円曲線法 (ECM)[9] であり、後者に属するもので最高速は一般数体ふるい法 (GNFS) [10] である。どちらも準指数関数時間を要する。ただし GNFS に関しては、アルゴリズムの一部 (線形代数と呼ばれる最終段階の処理) を専用ハードウェアに実行させて高速化を図る手法 (circuit-based NFS) が提案された [3]。またこのアイデアに基づき、専用ハードウェアを改良するとともに、その利用を前提として GNFS を最適化する手法も考案されている [12]。具体的には以下の表となる。

	素因数分解アルゴリズム	
	楕円曲線法 (ECM)	一般数体ふるい法 (GNFS)
入力	n ($n = p^r q, p < q, r \geq 1$)	
実行時間	$O(\exp(c\sqrt{\log_e p \log_e \log_e p}) \cdot (\log_2 n)^2)$ (主に $\lvert p \rvert$ の関数. $c = 1.414$)	$O(\exp(c(\log_e n)^{1/3}(\log_e \log_e n)^{2/3}))$ ($\lvert n \rvert$ の関数. $c = 1.901$ (文献 [7]). ただし, circuit-based NFS では $c = 1.868$ と評価される (文献 [12]))
現実的な 実行条件	$\lvert p \rvert \leq \theta_e$ (2002 年現在 $\theta_e = 183$)	$\lvert n \rvert \leq \theta_g$ (2002 年現在 $\theta_g = 524$)

上記の表で「現実的な実行条件」とは、現在のそのアルゴリズムと計算資源をもってすれば現実的な時間内に分解が実現できる範囲のことであり、現実的という曖昧な言葉で定義されているうえに、利用可能な計算資源にも制限がないことから、当然のことながら θ_e や θ_g は規格化されていない不確定な値である。しかしながら、実際に分解に成功した例を目安にすることはそれほど不自然ではなく、その意味で 2002 年現在では $\theta_e = 183$ (文献 [14])、$\theta_g = 524$ (文献 [2]) と考えることができる[*12]。θ_e と θ_g の将来予想については後述する。また、専用ハードウェアによる GNFS (circuit-based NFS) に関しては、後に節を改めて [12] の概要を述べる。

なお ECM も GNFS も、アルゴリズムの基本方針を踏襲しながら実行時間を改良するための研究が常になされおり (例えば ECM については [15]、GNFS については [7] 及び専用ハードウェアを用いることを前提とした [3, 12])、その観点からは、両アルゴリズムとも 2002 年現在のものであることに注意を要する。

このほか、$n = p^r q$ 型の合成数に特化したアルゴリズムとして格子利用法 (LFM)[4] がある。このアルゴリズムの実行時間は、$\lvert p \rvert = \lvert q \rvert$ で r が $\log_2 p$ 程度のときは $\lvert n \rvert$ の多項式時間という速度にまで達する (従って容易に分解できる)。しかし r が小さい定数 (例えば $r = 1, 2, 3$) のときは指数関数時間を要し、 ECM や GNFS の速度には及ばない。

2.4.1.2 安全な合成数のサイズ

ここでいう「ある時点において安全な合成数のサイズ」とは、その時点において、そのサイズ及び型の n を分解する最速のアルゴリズムと計算資源をもってしても、現実的には分解を達成し得ない範囲にあると考えられることを意味するものとする。

[*12] 2003 年 4 月 1 日付のネットワーク上のアナウンスによれば、$\lvert n \rvert = 530$ (10 進 160 桁) の $n = pq$ 型の合成数が GNFS により分解された。F. Bahr, J. Franke, T. Kleinjung, M. Lochter and M. Böhm, “RSA-160”, `http://www.loria.fr/~zimmerma/records/rsa160`

有力な素因数分解アルゴリズムに関する上記の議論から明らかなように、$n = p^r q\ (p < q, r \geq 1)$ の合成数が安全であるためには、次の全てを満たすこと ((1) かつ (2) かつ (3)) が必要条件である。

(1) r が小さい定数であること
(2) $|p| \gg \theta_e$ であること
(3) $|n| \gg \theta_g$ であること

条件 (1) は LFM による攻撃の回避、(2) は ECM による攻撃の回避、(3) は GNFS による攻撃の回避が目的である。

しかしながら、θ_e と θ_g はもともと規格化された量ではなく、また仮に規格化されたとしても、$|p|$ や $|n|$ を不必要に大きくとればスキームのパフォーマンス (暗号化・復号・署名生成・署名検証等の計算時間) に望ましくない副作用をもたらす。そこで、その時点で目安となる θ_e や θ_g の値を勘案して、実際的な意味で $|p|$ や $|n|$ の安全な範囲を評価する必要がある。

2001 年度に実施した詳細評価では、評価者 5 人のうち 4 人は、$|p| = |q|$ ならば次の (a)(b) がどちらも 2001 年時点で成り立つ主張と認めている (他の 1 人は、安全と考えられる具体的範囲については明言していない)。

(a) 2001 年時点で、$n = pq$ は $|p| = |q|$ かつ $|n| \geq 1024$ で安全と考えられること
(b) 2001 年時点で、$n = p^2 q$ は $|p| = |q|$ かつ $|n| \geq 1024$ で安全と考えられること

ただし、細かな点でいくつか重要な補足意見があった。具体的にはマージンの問題である。直観的には、アルゴリズムの現実的な実行条件を決めるサイズ (θ_e や θ_g) と実際のサイズ ($|p|$ や $|n|$) の差がマージンに関係していると考えてよい。マージンを考える趣旨は、それが小さいと、アルゴリズムや利用可能な計算資源が仮に劇的に改良されたときには θ_e や θ_g が上昇し、そのままの p や n では分解される可能性を警戒するためである。例えば、ECM による攻撃に対するマージンを考えたとき、$n = pq$ と $n = p^2 q$ の場合、どちらも $|n| = 1024$ ならば最小素因数サイズ $|p|$ に違いが現れ、明らかに $n = p^2 q$ の場合のほうがマージンが小さい。

マージンの議論に関係することであるが、θ_e や θ_g が将来どのような値になるかを予測することは、素因数分解問題に依拠した各スキームの安全性の寿命を検討するうえで重要である。そのような予測は実際には簡単ではないが、文献 [5] では、Moore の法則を勘案し、過去に分解された合成数サイズの実績を未来に外挿することで、西暦何年にどういうサイズの合成数が分解可能になるかの予想式が導出されている。それによれば、$n = pq$ と $n = p^2 q$ がともに $|n| = 1024$ で $|p| = |q|$ の場合、前者の最小素因数 (512 ビット) が ECM で現実的に計算できるようになるのは 2048 年頃、後者の最小素因数 (342 ビット) が ECM で現実的に計算できるようになるのは 2027 年頃となる。また型にかかわらず $|n| = 1024$ の場合、この n が GNFS で現実的に計算できるようになるのは 2018 年頃ということである。したがって、「文献 [5] の予測が正しいとすれば」という仮定のもとで、(a)(b) において $|n|$ をともに下限の 1024 にとった場合、マージンの年次低下は不可避ではあるものの、 2002 年から向こう 10 年間は、分解を達成し得ない範囲の合成数として機能することになる。

一方、文献 [13] では、その時点 (西暦年) で推奨される $|n|$ の下限を与えている。上述の文献 [5] では、その時点で分解可能と予測される $|n|$ を示しているのに対して、文献 [13] では、その時点で確保すべきマージンまで推定して推奨値の下限を提示しているため、同一時点で比較すると、文献 [5] より文献 [13] のほうが当然ながら大きな値となる。なお、文献 [13] では Moore の法則を若干拡張した独自版の仮定をもとに予測を行っている。

2.4.1.3 補足

一般に、既存のアルゴリズムは研究され改良される可能性があるので、最高速の楕円曲線法 (ECM) や一般数体ふるい法 (GNFS) がさらに高速になる可能性は否定できない。特に、専用ハードウェアの利用による GNFS の実装に関しては、GNFS の線形代数部分のハードウェア化 [3, 12] だけでなく、ふるいの部分に対するハードウェア化 [18, 20, 8] も提案されている。これらの現実的な実装の可能性については、半導体技術の進歩によるハードウェアのコストと性能の変化に密接に関係しており、理論分野だけでなく情報通信技術全般を見渡しながらの注意が今後も必要である。ただし 2002 年の時点では、$|n| = 1024$ に関する限り、専用ハードウェアによる分解の脅威は切迫していないとする見解が大勢である [12, 17]。一方、格子利用法 (LFM) は、合成数の型によっては多項式時間で分解できる能力を有しており、考案されてから時間が経っていないことを考えれば、その潜在的な可能性は相対的に大きい。さらに、マージンの議論で述べたように、マージンの年次低下が不可避であることにも注意しなければならない。以上を踏まえたうえで、(a)(b) の主張の成立を今後も継続して監視すべきである。

そのほか、素因数分解問題の難しさを左右する可能性のあるいくつかの事項を最後に列挙する。

- Shor [21] のアルゴリズムを実装した量子計算機は、素因数分解問題を効率的に解いてしまう。もしそのような量子計算機が大規模な入力に対しても処理できるまでのレベルに達したならば、素因数分解問題は、少なくとも暗号プリミティブとしての役割を事実上、終えることになる。2002 年現在、Shor のアルゴズムを実装した量子計算機上で分解された合成数としては 4 ビットの 15 (=3×5) が報告されている [22]。
- 素因数分解問題と帰着関係にある数論的な問題に対する効率的なアルゴリズムにも注意を要する。例えば、$n = p^r q$ $(r > 1)$ の分解問題は平方無縁部分 (squarefree part) 抽出問題に多項式時間で帰着する。平方無縁部分抽出問題とは、入力 n に対して、$n = u^2 v$ かつ v が平方無縁となる $\{u, v\}$ を出力する問題である (v が平方無縁とは、v が a^2 型の因数 $(a > 1)$ をもたないことである)。もし、これに対する効率的アルゴリズムが発見されれば、結果として $n = p^r q$ $(r > 1)$ 型の分解問題は効率的に解けることになる。ただし、$n = pq$ $(r = 1)$ の分解問題が平方無縁部分抽出問題に帰着するとは知られていない。
  なお、与えられた有理整数が素数かどうかを決定性多項式時間で判定するアルゴリズムが 2002 年に発見された [1]。素因数分解問題が決定性多項式時間で解けるならば、素数判定問題も決定性多項式時間で解ける。しかし、逆は知られていない。したがって、文献 [1] の結果は素因数分解問題の困難さに直接的に影響を与えるものではない。
- 構造的計算量理論において、計算量のクラスの包含関係に関する劇的な結果が証明されれ

ば、素因数分解問題を含む数論的問題の多くが、その難しさに劇的な影響を受ける可能性がある。なお、素因数分解問題と多項式時間 Turing 帰着の意味で等価な言語は、NP ∩ co-NP に存在する。

- Moore の法則がいつ破綻するかにより、現実的に分解可能なサイズは影響を受ける。Moore の法則の寿命は 2005 年以降に延びたと考えられているが、もしそれが破綻すれば、少なくとも単体の CPU の高速化による脅威は緩和され、現実的に分解可能なサイズは頭打ちになる可能性がある。ただしその場合でも、莫大な数の CPU が参加する分散処理 (massively distributed computing) による分解が脅威として依然残る。

2.4.1.4 専用ハードウェアによる数体ふるい法の最適化

本節では文献 [12] の要旨を紹介する。

素因数分解問題の困難性に安全性の根拠をおく公開鍵暗号プリミティブの多くは、一般的な素因数分解アルゴリズムである一般数体ふるい法 (GNFS) の計算量を基にして、その安全性が評価される。最近、その GNFS を、より効率的に行う方法として専用 H/W を用いる方法が提案された ([3, 12] 参照)。素因数分解アルゴリズムへの専用 H/W 適用の先行研究としては、[11, 16, 18] があるが、[3] ではそれらに対する優位性も主張されている。

ここでは GNFS で計算負荷の高い部分を、“関係式収集ステップ”と“行列計算ステップ”の 2 ステップに大きく分けて記述する。[3] では、両ステップに対して専用 H/W アプローチの提案が述べられているが、特に、行列計算ステップに対してある種のソーティングアルゴリズムを用いる具体的改良法が記述されている。 また [3] においては、同ステップに対してルーティングアルゴリズムの適用可能性も触れられている。その後、[12] で、同ステップに対しルーティングを基にした専用 H/W が具体的に提案された。

[3, 12] では、GNFS にかかる “スループットコスト”(時間 × 金額) を測度にして、行列計算ステップへの専用 H/W 使用の有効性を見積もっている。[12] において主張されているように、$1.17 \cdot n$ ビット合成数に対する従来法と n ビット合成数に対する改良法をほぼ同等のスループットコストとする漸近的な見積もりが妥当である。

スループットコストを測度にして評価することには、まだ議論の余地が残されているが、上述の結果は [3, 12] のアプローチの有効性を示している

また [12] では、[3] での専用 H/W と比べ、よりコンパクトな H/W 実現法を提案することにより、1024 ビット合成数に対する行列計算ステップは、(マスク生成費を除いて) 約 5 千ドルの専用 H/W を用いれば数時間でできるようになると主張している。但し、著者らは上述の評価を “optimistic”な前提でのものであると断り書きしている。つまり、その見積もりの際に想定されている疎行列 (“小行列”) のサイズは、漸近的なスループットコスト関数を最適化した時に得られる値であり、実際の 1024 ビット合成数 GNFS に関して述べる時には、注意を払う必要があるということである。

[12] では、512 ビット合成数 GNFS に成功した [6] での実際の行列サイズを基に、従来の漸近関数を使っての 1024 ビット合成数に対する行列サイズ (“大行列”) も見積もっており、また、そ

れに対する実際的なスループットコストも見積もっている。現在 GNFS 全体で、関係式収集ステップが律速ステップであることを考慮すると、これら評価値も重要な判断基準になると思われる (詳細は [12] 参照)。

[12] に記述された専用 H/W の実現には技術的困難は依然存在すると思われるが、量子計算機などと比べて、現時点でも、この専用 H/W の実現可能性は高いと思われる。

[12] においては、この専用 H/W の提案により、1024 ビット RSA の安全性の根拠は、GNFS 中の関係式収集ステップの計算困難性に集約されたとし、現時点で 1024 ビット RSA は依然安全であるとしている。

また、関連研究として、数体ふるい法の専用 H/W による実現方法について論じた論文 [8] が発表された。

参考文献

[1] M. Agrawal, N. Kayal, N. Saxena, “PRIMES is in P,” August 2002.
Availabe at http://www.cse.iitk.ac.in/news/primality.pdf

[2] F. Bahr, J. Franke, T. Kleinjung, “Factorization of 158-digit cofactor of $2^{953} + 1$,” January 21, 2002.
Availabe at http://www.crypto-world.com/announcements/c158.txt

[3] D.J. Bernstein, “Circuits for integer factorization: a proposal,” preprint,
Available at http://cr.yp.to/papers.html#nfscircuit

[4] D. Boneh, G. Durfee, N. Howgrave-Graham, “Factoring $N = p^r q$ for large r,” Proc. Crypto’99, LNCS 1666, Springer-Verlag, pp.326–337, 1999.

[5] R. P. Brent, “Recent progress and prospects for integer factorisation algorithms,” Proc. COCOON 2000, LNCS 1858, Springer-Verlag, pp.3–22, 2000.

[6] S. Cavallar, B. Dodson, A.K. Lenstra, W. Lioen, P.L. Montgometry, B. Murphy, H.J.J. te Riele, et.al., “Factorization of a 512-bit RSA modulus”, Proc. Eurocrypt 2000, LNCS 1807, Springer-Verlag, pp.1–17, 2000.

[7] D. Coppersmith, “Modifications to the number field sieve,” J. Cryptology, vol.6, pp.169–180, 1993.

[8] W. Geiselmann, R. Steinwandt, “A Dedicated Sieving Hardware,”, Proc. PKC 2003, LNCS 2567, Springer-Verlag, pp.254–266, 2003.

[9] H. W. Lenstra, Jr., “Factoring integers with elliptic curves,” Annals of Mathematics, vol.126, pp.649–673, 1987.

[10] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, “The number field sieve,” Proc. 22nd STOC, pp.564–572, 1990.

[11] A.K. Lenstra, A. Shamir, “Analysis and optimization of the TWINKLE factoring device,” Proc. Eurocrypt 2000, LNCS 1807, Springer-Verlag, pp.35–52, 2000.

[12] A.K. Lenstra, A. Shamir, J. Tomlinson, E. Tromer, “Analysis of Bernstein’s Factorization Circuit”, Proc. Asiacrypt 2002, LNCS 2501, Springer-Verlag, pp.1–26, 2002.

[13] A. K. Lenstra, E. Verheul, “Selecting cryptographic key sizes,” Proc. PKC 2000, LNCS 1751, Springer-Verlag, pp.446–465, 2000.

[14] The ECMNET Project
Availabe at http://www.loria.fr/~zimmerma/records/ecmnet.html

[15] E. Okamoto, R. Peralta, “Faster factoring of integers of a special form,” IEICE Trans. Fundamentals, vol.E79-A, pp.489–493, 1996.

[16] C. Pomerance, J. W. Smith, R. Tuler, “A pipeline architecture for factoring large integers with the quadratic sieve algorithm,” SIAM Journal on Computing, vol.17, pp.387–403, 1988.

[17] RSA Security Inc, “Has the RSA algorithm been compromised as a result of Bernstein’s Paper?” April 8, 2002.
Availabe at http://www.rsasecurity.com/rsalabs/technotes/bernstein.html

[18] A. Shamir, “Factoring large numbers with the TWINKLE device,” Cryptographic Hardware and Embedded Systems (CHES’99), LNCS 1717, Springer-Verlag, pp.2–12, 1999.

[19] A. Shamir, “Fatoring large numbers with the TWIRL device,” presented at Asiacrypt 2002 Rump Session, December 3, 2002.

[20] A. Shamir, E. Tromer, “Factoring large numbers with the TWIRL device (preliminary draft),” Preliminary draft
Available at http://www.wisdom.weizmann.ac.il/~tromer/

[21] P. W. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,” SIAM J. Computing, vol.26, no.5, pp.1484–1509, 1997.

[22] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, I. L. Chuang, “Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance,” Nature, Vol.414, pp.883–887, 20/27 December 2001.

2.4.2 離散対数問題

有限群の離散対数問題 (以下、DLP(Discrete Logarithm Problem) と記述) の困難さに依拠して暗号プリミティブの安全性を主張する暗号スキームを横断的に評価するため、DLP の現状に関する詳細評価を実施した。ここでは、それらの評価結果に基づき、代表的な攻撃アルゴリズム、現実的に安全と考えられる鍵サイズとその将来予想などを総括する。また、問題そのものの難しさに影響を与えうる事項に関しても最後に補足する。

2.4.2.1 離散対数問題の定義

G を有限群とする。G の元 g と $u = g^x (x \in \mathbb{Z})$ が与えられたときに x を求める問題を離散対数問題 (DLP) という。暗号アプリケーションで用いられる DLP は、G として有限体の乗法群、または有限体上定義された楕円曲線有理点群をとることが多い。有限体上定義された楕円曲線有理点群の場合 (ECDLP: Elliptic Curve Discrete Logarithm Problem) は次節にゆずり、本節では主に有限体乗法群の DLP について述べる。

2.4.2.2 攻撃法

DLP に対する攻撃法は、一般の有限群に適用できる一般法と、有限体乗法群の性質を用いる index calculus 法 [7] に大別される。各々について知られている攻撃法およびその計算量を表にまとめる。一般法の計算量は、群位数ビット長の指数時間オーダとなるが、index calculus 法の計算量は、準指数時間オーダとなり、index calculus 法の方が高速であるといえる。

一般法

N を群の位数とすると、一般法の計算量は N に依存する。下記のうち、Pollard 法は並列化可能であり、m 台の計算機を用いれば、DLP を解くのに要する計算量は 1 台あたり、$\sqrt{\pi N/2}/m$ と見積もられる [8]。ECDLP の場合の解読記録に関しては、次節を参照されたい。

攻撃法の名称	計算量	文献
全数探索法	$O(N)$	
Pohlig-Hellman 法	素位数部分群上の DLP に帰着	[9]
Baby-Step/Giant-Step 法	$O(\sqrt{N})$	[11]
Pollard 法	$\sqrt{\pi N/2}$	[10]

index calculus 法

index calculus 法は、素体の乗法群に対して適用できる数体ふるい法と、小さな標数の拡大体の乗法群に対して適用できる関数体ふるい法とに大別される。数体ふるい法は更に、特殊数体ふるい法と一般数体ふるい法とに類別される。また 2002 年になって、Coppersmith 法 [3] を用いて $\mathbb{F}_{2^{607}}$ における DLP を解読したという報告がされた。各々についてその計算量および 2002 年

における解読記録を表にまとめる。

表中、q は有限体の位数を表し、記号 $L_q[a,b]$ は、$L_q[a,b] = e^{b(\log q)^a(\log\log q)^{1-a}}$ を表すものとする。

攻撃法の名称	計算量	解読記録	文献
一般数体ふるい法	$L_q[1/3, c+o(1)], c = (64/9)^{1/3} = 1.9229...$	120 桁	[4, 12]
特殊数体ふるい法	$L_q[1/3, c+o(1)], c = 1.5262...$	129 桁	[15, 12]
関数体ふるい法	$L_q[1/3, c+o(1)], c = (32/9)^{1/3}$	$\mathbb{F}_{2^{521}}$ = 157 桁	[5, 1]
Coppersmith 法	$L_q[1/3, c+o(1)], c \approx 1.4$	$\mathbb{F}_{2^{607}}$ = 183 桁	[13, 14]

2.4.2.3 安全な鍵のサイズ

DLP の困難性に安全性の根拠を置く暗号プリミティブを採用する場合、その鍵サイズは対応する DLP を現実的に解くことが困難になるように十分大きく取る必要がある。今後長期間にわたって安全な鍵サイズを考える際に、計算機の進歩、解読アルゴリズムの進歩等様々な要因を考慮する必要が生ずる。ここでは、よく引用されている以下の有名な 2 つの「予測」を取り上げ、その見方を解説する。

Brent の予測式 [2]

ある桁の素因数分解問題が解かれるであろう年を予測した式。これまでの素因数分解記録に基づいている。10 進数 D 桁の素因数分解問題は、西暦

$$Y = 13.24D^{1/3} + 1928.6$$

年に解かれ得るであろうと予測している。素体 DLP の場合、素因数分解よりも、20 ビット分解読の手間が遅くなるという報告があり、それを加味すると D 桁の素体 DLP が解かれ得るであろう西暦年 Y の予測式は、

$$Y = 13.24(D+6)^{1/3} + 1928.6$$

となる。$1024, 2048, 4096$ ビットの場合の Y 年は、以下のようになる。

標数 2 の有限体乗法群の場合は、これまでの解読記録は素体よりも大きな桁が解かれているため、より長いビット長を使用することが推奨される。

年	ビット長
2019	1024
2042	2048
2070	4096

Lenstra-Verheul の表 [6]

Lenstra-Verheul の表 (以下、LV と記述) は、1982 年の DES と同程度の強度をその年に持つための鍵長を表している。Brent の予測式が、その年に解かれ得る鍵長を表しているのに対し、

LV はその年に安全と思われるかなり大きなマージンをとった鍵長になっていることに注意されたい。

DLP 攻撃の最速アルゴリズム index calculus 法に対しては、基礎となる体のビット長を考慮する必要があるが、DSA など位数が比較的小さい部分群を用いるものは、部分群に対する一般攻撃法が index calculus 法よりも効率的にならないことも考慮にいれる必要がある。部分群 DLP を用いるものは、基礎となる体のビット長とともに部分群位数の項も表の値以上に大きく取る必要がある。以下に LV の表から 10 年おきの値を抜粋する。

年	ビット長	群位数
2002	1028	127
2010	1369	138
2020	1881	151
2030	2493	165
2040	3214	179
2050	4047	193

2.4.2.4　その他

そのほか、DLP の難しさを左右する可能性のあるいくつかの事項を最後に列挙する。

- 量子計算機が実用化レベルに達したならば、DLP は効率的に解けるようになり、DLP は、少なくとも暗号プリミティブとしての役割は終えることになる。
- 何らかの数学的なブレイクスルーが起こり、index calculus 法より効率的な DLP 解読アルゴリズムが発見された場合には、上記予測は、大きく変更され得ることに注意を払う必要があるであろう。

参考文献

[1] L. Adleman, “The function field sieve,”In ANTS I(1994), LNCS 877, pp.108–121, Springer-Verlag, 1994.

[2] R. P. Brent, “Recent progress and prospects for integer factorisation algorithms,” Proc. COCOON 2000, LNCS 1858, pp.3–22, Springer-Verlag, 2000.

[3] D. Coppersmith, “Fast evaluation of logarithms in fields of characteristic two,” IEEE Trans. on Inform. Theory, IT-30, pp.587–594, 1984.

[4] A. Joux and R. Lercier, Discrete logarithms in GF(p), Announcement on the NMBRTHRY Mailing List, 17. April 2001.

[5] A. Joux and R. Lercier, Discrete logarithms in GF(2^n), Announcement on the NMBRTHRY Mailing List, 25. September 2001.

[6] A. K. Lenstra, E. Verheul, “Selecting cryptographic key sizes,” Proc. PKC 2000, LNCS 1751, pp.446–465, Springer-Verlag, 2000.
Available at `http://www.cryptosavvy.com/table.htm`

[7] K. McCurley, “The discrete logarithm problem, Cryptography and Computational Number Theory,” Proc. Symp. Appl. Math, AMS Lecture Notes, 42, pp.49–74, 1990.

[8] P. van Oorschot and M. Wiener, “Parallel collision search with applications to hash functions and discrete logarithms,” 2nd ACM Conference on Computer and Communications Security, pp.210–218, ACM Press, 1994.

[9] S. Pohlig and M. Hellman, “An improved algorithm for computing logarithms over GF(p) and its cryptographic significance,” IEEE Trans. on Inform. Theory, IT-24, pp.106–110, 1978.

[10] J. Pollard, “Monte Carlo methods for index computations mod p,” Math. Comp., 32, pp.918–924, 1978.

[11] D. Shanks, “Class number, a theory of factorization and genera,” In Proc. Symp. Pure Math. 20, AMS, Providence, R.I., pp.415–440, 1971.

[12] O. Schirokauer, “Discrete logarithms and local units,” Phil. Trans. R. Soc. London A 345, pp.409–423, 1993.

[13] E. Thomé, “Computation of discrete logarithms in GF(2^{607}),” In Proc. ASIACRYPT 2001, LNCS 2248, pp. 107–124, 2001.

[14] E. Thomé, Discrete Logarithms in GF(2^{607}).
Available at `http://www.lix.polytechnique.fr/Labo/Emmanuel.Thome/announcement/announcement.html`

[15] D. Weber and T. Denny, “The solution of mccurleys discrete logarithm challenge,” In Advances in Cryptology - Crypto’98, LNCS 1462, pp. 458–471, 1998.

2.4.3 楕円曲線上の離散対数問題

2.4.3.1 定義

$K = \mathbb{F}_q$ を標数 p の素体の k 次拡大である、位数 q の有限体とする ($q = p^k$)。K 上の楕円曲線 E とは

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

という形の式によって定義される非特異な平面曲線である。ここで a_i は K の元である。

楕円曲線 E の有理点の集合 $E(K)$ は群をなす。$P \in E(K)$ を位数 N の有理点とし、Q を P が生成する巡回群の要素とする。

楕円曲線上の離散対数問題 (ECDLP) 与えられた $E, \mathbb{F}_q, P, N$ および Q に対して、$Q = nP$ となる整数 $n \in [0, N-1]$ を求めよ。

上記 $E, \mathbb{F}_q, P, N$ は楕円曲線暗号では楕円曲線パラメータと呼ばれる。特に、点 P はベースポイントと呼ばれる。

2.4.3.2 攻撃法

ECDLP に対する攻撃法は、楕円曲線自体の性質を用いない一般法と楕円曲線の性質を用いる特殊法に大別される。一般法と特殊法ごとに知られている攻撃法とそれを回避するための条件を表にまとめる。表中、計算量/帰着関係項目の「→ xxx 上の DLP」は ECDLP が群 xxx 上の DLP に帰着されることを示す。

一般法

攻撃法の名称	計算量/帰着関係	回避条件	文献
全数探索法	O(N)	N を十分大に	
Pohlig-Hellman 法	→ 素位数部分群上の DLP	N をほぼ素数に (注 1)	[1]
Baby-Step/Giant-Step 法	O($\sqrt{N}$)	N のビット長 ≥ 160 (注 2)	
Pollard 法	$\sqrt{\pi N/2}$ (注 3)	N のビット長 ≥ 160 (注 2)	[2]

注 1 N がほぼ素数であるとは、N が N と同程度の大きさの素数と小さな整数 (1,2,4 等) の積であることを意味する。

注 2 Baby-Step/Giant-Step 法や Pollard 法は、2002 年現在で N のビット長が 160 以上であれば実行不可能であると思われる。

注 3 Pollard 法は “Las Vegas” タイプのアルゴリズムで、その計算量評価は必要となる楕円曲線上の加算回数の統計的な見積もりである。また、Pollard 法は並列化可能であり、m 台の計算機を用いれば、ECDLP を解くのに要する計算量は 1 台あたり、$\sqrt{\pi N/2}/m$ と見積もられる [3]。

特殊法

N はほぼ素数であり、その最大素因子を l とする。

攻撃法の名称	計算量/帰着関係	回避条件	文献
自己同型法	位数 m の自己同型を用いるとき、Pollard 法を $\sqrt{m}$ 倍高速化	(注 1)	[4, 5]
Weil/Tate Pairing 法	→ 拡大体 $\mathbb{F}_{q^s}$ の乗法群上の DLP	$N \nmid q^s - 1$ $(1 \leq s \leq 30)$	[6, 7]
Anomalous curve 法	→ 素体 $\mathbb{F}_p$ の加法群上の DLP	$p \neq l$	[8, 9, 10]
Weil descent 法	→ 超楕円曲線上の DLP	$p = 2$, k:素数 または $p \neq 2$ (注 2)	[11]

注 1 自己同型法では楕円曲線の任意の自己同型が使用できるわけではない。Koblitz 曲線がもつ自己同型が自己同型法の対象となる典型である。ただし、その場合でも、自己同型法の効果は 163 ビット楕円曲線の場合に、ECDLP を解く計算量が最大で 5 ビット少なくなる程度である。Koblitz 曲線については第 2.3.2 節 ECDSA 参照。

注 2 最近、Diem[12] によって奇標数の場合にも $k = [\mathbb{F}_q : \mathbb{F}_p] = 5, 7$ のときには、Weil descent 法が成立する場合があることを示唆する結果が報告されている。OEF(Optimal Extension Field) を用いる場合注意が必要である。

2.4.3.3 実験結果

Certicom 社は 1997 年から ECDLP 解読の研究を促すため、ECC challenge を主催している。

Escott ら [13] は並列化された Pollard 法を用いて、ECC challenge の一つである ECCp-97 を解いている。ECCp-97 は素体上の位数 97 ビットの楕円曲線である。ECCp-97 を解くために、1200 台以上の計算機を用い、53 日をかけて、2×10^{14} 回の楕円曲線上の加算を実行したと報告されている。

また、最近になって、Monico らは ECCp-109 を解読した [14]。約 10000 人、247 チームが参加し、549 日かけて約 3.6×10^{16} 個の楕円曲線の点 (その内、”distinguished point”は、約 6.8×10^7 個) を計算したと報告されている。

また、Harley ら [15] は自己同型法を援用した並列化 Pollard 法を用いて、ECC challenge の一つである ECC2K-108 を解いている。ECC2K-108 は標数 2 の有限体上の位数 108 ビットの Koblitz 曲線である。ECC2K-108 を解くために、約 9500 台の計算機を用い、4 ヶ月をかけて、2.3×10^{15} 回の楕円曲線上の加算を実行したと報告されている。

2.4.3.4 安全な群位数サイズ

2002 年現在、楕円曲線上の離散対数問題は、特殊法にあげた特殊な楕円曲線を除けば、群位数 (より正確には、ベースポイントの位数) が 160 ビット以上の素因子を含めば十分安全であるとされている。将来のある時点における、安全な群位数サイズを見積もるには、解読アルゴリズ

ムの実行に必要な実際的な計算量、インターネット資源等を活用して実行可能な最大計算量の増加、解読アルゴリズムの進歩等を見極める必要があり、その正確な数字を算出するのは困難である。ここでは、参考情報として、Lenstra と Verheul による結果 ([16]) を紹介する。

Lenstra と Verheul による安全な群位数サイズの見積もり [16]

Year	群位数のビット長 (no progress)	群位数のビット長 (with progress)
2002	135	139
2010	146	160
2020	161	188
2030	176	215
2040	191	244
2050	206	272

上記の表は、1982 年当時の DES と同程度の強度を ECDLP が該当年に持つための群位数のビット長を示す。また、群位数のビット長 (no progress) フィールドの値は解読アルゴリズム自体の進歩は仮定しない場合の値を、群位数のビット長 (with progress) フィールドの値は解読アルゴリズムが、ECDLP を解くのに必要な計算量を 18 ヶ月で半減させる割合で、進歩すると仮定した場合の値を示す。

参考文献

[1] S. Pohlig and M. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE Trans. on Infor. Th., 24, 106-110, 1978.

[2] J. Pollard, Monte Carlo methods for index computations mod p, Math. Comp., 32, 918-924, 1978.

[3] P. van Oorschot and M. Wiener, Parallel collision search with applications to hash functions and discrete logarithms, 2nd ACM Conference on Computer and Communications Security, 210-218, ACM Press 1994.

[4] R. Gallant, R. Lambert and S. Vanstone, Improving the parallelised Pollard lambda search on binary anomalous curves, Math. Comp., 69, 1699-1705, 2000.

[5] M. J. Wiener and R. J. Zuccherato, Faster attacks on elliptic curve cryptosystems, Selected Areas in Cryptography - SAC 1999, Springer-Verlag LNCS 1556, 190-200, 1999.

[6] A. Menezes, T. Okamoto and S. Vansone, Reducing elliptic curve logarithms to logarithms in finite fields, IEEE Trans. on Infor. Th., 39, 1639-1646, 1993.

[7] G. Frey and H. -G. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, Mathematics of Computation, 62, 865-874, 1994.

[8] P. N. Smart, The discrete logarithm problem on elliptic curves of trace one, J. Cryptology 12, 193-196, 1999.

[9] T. Satoh and K. Araki, Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves, COMMENTARII MATHEMATICI UNIVERSITATIS SANCTI PAULI, vol. 47, No. 1, 81-92, 1998.

[10] I. A. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curves in characteristic p, Math. Comp. 67, 353-356, 1998.

[11] P. Gaudry, F. Hess and N. Smart, Constructive and Destructive Facets of Weil Descent on Elliptic Curves, Journal of Cryptology, Vol.15, No.1, pp19-46, 2002.

[12] C. Diem, The GHS-attack in odd characteristic, preprint, 2001.
Available at http://www.exp-math.uni-essen.de/~diem/english.html

[13] A. Escott, J. Sager, A. Selkirk and D. Tsapakidis, Attacking elliptic curve cryptosystems using the parallel Pollard rho method, CryptoBytes - The Technical Newsletter of RSA Laboratories, volume 4, number 2, Winter 1999, 15-19.
Also available at http://www.rsasecurity.com

[14] Chris Monico et. al, ECCp-109 : The ECCp-109 Challenge is Solved!
Available at http://www.nd.edu/~cmonico/eccp109/solved.html

[15] R. Harley, Elliptic Curve Discrete Logarithms: ECC2K-108.
Available at http://cristal.inria.fr/~harley/ecdl7/

[16] A. K. Lenstra, E. Verheul, Selecting cryptographic key sizes, Proc. PKC 2000, LNCS 1751, Springer-Verlag, pp.446–465, 2000.

[17] Wiener, M.J., Zuccherato,R.J., Faster attacks on elliptic curve cryptosystems, Proc. SAC1998, LNCS 1556, Springer-Verlag, pp.190–200, 1999.

2.5　公開鍵暗号技術に関するパラメータの選択方法について

2.5.1　素因数分解問題に係わる暗号技術

2.5.1.1　RSA パラメータの選択

RSA アルゴリズムを利用した暗号技術の安全性はその基礎となる RSA 暗号基本演算 (RSA 検証基本演算) と復号基本演算 (署名基本演算) すなわち、RSAEP(RSAVP) と RSADP(RSASP)(2.3.4.3 節の RSA プリミティブを参照のこと) の安全性に密接に関係している。多くの研究者によって RSAEP/RSADP(RSASP/RSAVP) の安全性が検討されており、RSA の不適切な使用は大いに危険であることが示されている。

ここでは、RSAEP/RSADP(RSASP/RSAVP) に対する主な攻撃方法について解説することにする。

鍵対生成

RSA 公開鍵は、ランダムに生成された、ほぼ同じサイズの異なる 2 つの奇素数 p および q の積 N と、$\mathrm{GCD}(e, p-1) = \mathrm{GCD}(e, q-1) = 1$ を満たす 3 以上で $N-1$ 以下の整数 e(公開指数) の組 (N, e) である。

RSA 秘密鍵は、RSA 公開鍵と同じ、ほぼ同じサイズの異なる 2 つの奇素数 p および q の積 N と、$ed = 1 (\mathrm{mod}\ \mathrm{LCM}(p-1, q-1))$(ただし、$e$ は対応する公開指数である。) を満たす N より小さい正の整数 d(秘密指数) の組 (N, d) である。

ランダムに素数を生成する方法については、現時点で CRYPTREC で特に指定する方法はないが、例えば、[1] や [8] の Annex A などの方法がある。

なお、規格書 PKCS #1 v2.1[9] においては、multi-prime RSA と呼ばれる鍵対生成方式が含まれているが、RSA セキュリティ社からの CRYPTREC 応募資料 (RSA-OAEP、RSA-PSS) において応募の対象外であったため、CRYPTREC としては評価対象外であり、CRYPTREC で指定する RSA 鍵対生成方法は、2 つの異なる素数の積による鍵対生成方法のみとする。

事実 11 Let (N, e) be an RSA public key. Given the private key d, one can efficiently factor the modulus $N = pq$. Conversely, given the factorization of N, one can efficiently recover d.

証明は、[2] を参照のこと。この事実により、複数のユーザ間で同じ N を共有することは安全でないので、N をユーザ間で共有してはいけない。

RSAEP の逆演算について

一般的に、現在知られているこの問題を解く唯一の方法は素因数分解による方法である。素因数分解の困難さについては、2.4.1 節を参照のこと。なお、CRYPTREC では、「2001 年時点で、

N のサイズについては、1024 ビット以上で安全と考えられる」と判断している。

D. Boneh, R. Venkatesan[5] により、小さな公開指数の場合、RSAEP の逆演算を求めることと、N の素因数分解を求めることとは等価ではないことが示されているが、素因数分解によらずに、RSAEP の逆演算を求めることは依然として未解決である。

小さな秘密指数 d について

定理 12 (M. Wiener) Let $N = pq$ with $q < p < 2q$. Let $d < \frac{1}{3}N^{1/4}$. Given (N, e) with $ed = 1 \bmod (p-1)(q-1)$, an attacker can efficiently recover d.

証明は、[2] を参照のこと。

また、D. Boneh, G. Durfee[3] により、RSAEP/RSADP の単独実装では、$d < N^{0.292}$ のとき、(N, e) から d を効率的に解くことができる可能性があることが示されている。従って、計算効率の点から d を小さな値に制限することはすべきでない。

Coppersmith の定理について

ここで、Coppersmith による、後述の攻撃において最も基本的な役割を果たす定理を引用する[6]。

定理 13 (D. Coppersmith) Let $p(x)$ be a polynomial of degree δ in one variable modulo an integer N of unknown factorization. Let X be the bound on the desired solution x_0. If

$$X < \frac{1}{2}N^{1/\delta-\epsilon},$$

then in time polynomial in $(\log N, \delta, 1/\epsilon)$, we can find all integers x_0 with $p(x_0) = 0 \pmod N$ and $|x_0| < X$

同報通信について

Coppersmith の定理を適用することにより、ごく単純なパディング手法を使用した場合、複数のユーザに対して同一の暗号文を送ることは平文解読につながる危険性があることが導かれる。

定理 14 (J. Håstad) Let $N_1, \cdots, N_k$ be a pairwise relatively prime integers, and set $N_{min} = min_i(N_i)$. Let $g_i \in \mathbb{Z}_{N_i}[x]$ be k polynomials of maximum degree d. Suppose there exists a unique $M < N_{min}$ satisfying

$$g_i(M) = 0 \bmod N_i \, for \forall i = 1, \cdots, k \tag{2.5}$$

Under the assumption that $k > d$, one can efficiently find M given $(N_i, g_i)_{i=1}^k$.

証明は、[2] を参照のこと。

従って、特に RSAES-PKCS1-v1_5 では、この攻撃法に対して、EME-PKCS1-v1_5 エンコーディング手法 (図 2.4 参照) において、各暗号化処理ごとに独立にランダム文字列 PS を生成する必要性があることに注意を要する。

小さな公開指数 e について

公開指数 e が小さいとき、同じ RSA 秘密鍵で暗号化された暗号文の間に既知の多項式で表わされる関連性が存在する場合、平文が解読される危険性があることが示されている [7]。

従って、特に RSAES-PKCS1-v1_5 では、この攻撃法に対して、EME-PKCS1-v1_5 エンコーディング手法 (図 2.4 参照) において、各暗号化処理ごとに独立にランダム文字列 PS を生成する必要性があることに注意を要する。

また、Coppersmith の定理を適用することにより、公開指数 e が小さいとき、平文の大部分が既知であるような場合、平文解読につながる危険性があることが導かれる [6]。同様に、公開指数 e が小さいとき、平文にランダム文字列をパディングしたような場合、2 つの暗号文から平文が解読される危険性があることが示されている [6]。

定理 15 Let (N, e) be a public RSA key where N is n-bits long. Set $m = n/e^2$. Let $M \in \mathbb{Z}_N^*$ be a message of length at most $n - m$ bits. Define $M_1 = 2^m M + r_1$ and $M_2 = 2^M + r_2$, where r_1 and r_2 are distinct integers with $0 \leq r_1, r_2 < 2^m$. If an attacker is given (N, e) and the encryptions C_1,C_2 of M_1,M_2 (but is not given r_1 or r_2), he can efficiently recover M.

証明は、[2] を参照のこと。

従って、特に RSAES-PKCS1-v1_5 では、この攻撃法に対して、十分大きな公開指数 e を取る必要性があることに注意を要する。

部分情報の漏洩について

■因数 p, q に関する部分情報について　p(q も同様) の最下位 $\lceil \log_2 n/4 \rceil$ ビット、あるいは p(q も同様) の最上位 $\lceil \log_2 n/4 \rceil$ ビットが与えられれば、効率的に N を因数分解することが可能である [6]。従って、p および q は全体を保護しなければならない。

■秘密指数 d に関する部分情報について　RSA アルゴリズムでは、e が非常に小さい場合は、d の上位半数のビットが漏洩する [2]。残りのビットを決定することは依然難しいが、

秘密指数 d の最下位 $\lceil \log_2 n/4 \rceil$ ビットが与えられた時、もし $e < \sqrt{n}$ ならば、d の全ビットを再構築することが可能である [4]。従って、秘密指数 d の残りのビットを正しく保護することは大変重要である。

参考文献

[1] ANSI X9.80, “Prime Number Generation, Primality Testing, and Primality Certificates”, American National Standard for Financial Services, 2001.

[2] D. Boneh, “Twenty years of attacks on the RSA cryptosystem”, Notice of the AMS, volume 46 no 2, 1999.

[3] D. Boneh and G. Durfee, “Cryptanalysis of RSA with Private Key d Less than N^0.2992”, In J. Stern editor, Advances in Cryptology - Eurocrypt’99, Lecture Notes in Computer Science, volume 1533, pp1-11, Springer Verlag 1999.

[4] D. Boneh, G. Durfee and Y. Frankel, “An attack on RSA given a fraction of the private key bits”, In K. Ohta and D. Pei editors, Advances in Cryptology - Asiacrypt’98, Lecture Notes in Computer Science, volume 1514, pp25-34, Springer Verlag 1998.

[5] D. Boneh and R. Venkatesan, “Breaking RSA may not be equivalent to factoring”, In N. Nyberg editor, Advances in Cryptology - Eurocrypt’98, Lecture Notes in Computer Science, volume 1403, pp59-71, Springer Verlag 1998.

[6] D. Coppersmith, “Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities”, Journal of Cryptology, 10:233-260, 1997.

[7] D. Coppersmith, M. Franklin, J. Patarin and M. Reiter, “Low-Exponent RSA with Related Messages”, In U. Maurer editor, Advances in Cryptology - Eurocrypt’96, Lecture Notes in Computer Science, volume 1070, pp1-9, Springer Verlag 1996.

[8] “IEEE Std 1363-2000 IEEE Standard Specifications for Public Key Cryptography”,

[9] “PKCS #1 v2.1: RSA Cryptography Standard”, RSA Laboratories, June 14, 2002, available at `http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html`

2.5.2 離散対数問題に係わる暗号技術

離散対数問題に係わる暗号技術は、署名技術の DSA と鍵共有法の DH である。パラメータ選択にあたっては、それぞれの仕様書に定められた範囲を守って設定すべきである。素数や乱数の生成も、仕様書に書かれた生成方法や妥当性確認法を用いるべきである。

2.5.3 楕円曲線上の離散対数問題に係わる暗号技術

2.5.3.1 楕円曲線パラメータの選択

ある特殊な楕円曲線のクラスを用いた楕円曲線暗号に対しては、その特殊性を利用した効率的な解読攻撃が存在することが知られている。暗号で利用する楕円曲線の選択にあたっては、そのような攻撃が適用できないことをチェックする必要がある。

楕円曲線パラメータ選択のアプローチは大別すると、ランダムに選択する方法と、特殊なパラ

メータを使う方法の 2 種類となる。

ランダムに選択する方法は、曲線の係数を任意に選択して、その曲線が既存の攻撃法の対象とならないかどうかをチェックする方法である。既存攻撃の対象とならない適切な曲線が見つかるまで、少なくとも位数計算を繰り返す必要がある。位数計算には、通常 SEA(Schoof-Elkies-Atkin) 法 [26, 3, 17] が用いられていたが、標数 2 の場合は、近年、p 進的方法 [21]、算術幾何平均法 [9] 等が提案され、その改良・高速化研究も急速に進展している。[23, 27, 4, 5, 24, 10]

特殊なクラスの楕円曲線パラメータを使用する方法には、CM(Complex Multiplication) 法 [1, 16, 12]、Koblitz 曲線 [11]、超特異曲線 [15]、anomalous 曲線 [13] 等があるが、このうち超特異曲線および anomalous 曲線は、効率的な攻撃法 [14, 25, 20, 22] が発見されており、現在では使用されていない。

CM 法は、既存の攻撃法の対象とならない位数を先に決定し、その位数を持つような曲線を構成するものであるが、類数が比較的小さな判別式に対してしか使用できないという意味において、曲線のクラスが制限されている。しかしながら、この性質を用いた攻撃法は発見されていないため、小さな類数が、直ちに弱いことを示すものではない。CM 法による曲線の生成時間は、ランダム選択法より速いと期待されるが、圧倒的な差があるわけではないと考えられている。

超特異曲線、anomalous 曲線は導入当初は、効率的な攻撃法は見つかっていなかったが、後になって、乗法群への埋め込み [14]、加法群への埋め込み [25, 20, 22] という効率的な攻撃法がそれぞれ発見され、現在では、楕円曲線暗号システムで使用すべきではない。従って、曲線をランダムに生成した場合、これらの曲線ではないことをチェックする必要がある。

Koblitz 曲線 [11] は、スカラー倍算高速化のために導入されたが、攻撃法も高速化可能である [28]。現時点では、この攻撃法は致命的ではないが、特殊なクラスの楕円曲線に対して、効率的な攻撃法が発見される可能性は、すべての楕円曲線に対する効率的な攻撃法が発見される可能性より遥かに大きいため、今後注意が必要である。

ドメインパラメータ正当性チェック

楕円曲線暗号標準仕様 SEC1[19] で規定されている、ドメインパラメータの正当性チェック方法について以下に述べる。正当性チェックの目的は、パラメータの値が正しいものであることの確認と効率的な攻撃が適用できないことの確認の 2 つである。

パラメータを自ら生成する場合や、他者から与えられたパラメータを使用する場合などに、このチェックを行う必要がある。

パラメータは、演算を行う有限体の種類によって、大きな奇素数標数 p の場合と標数 2 の場合とに分かれる。それぞれの場合の確認は大体同じだが、細かい違いがあるため、以下それぞれの場合について示す。

1. 標数 p の場合

以下、ドメインパラメータを (p, a, b, G, n, h) とする。ただし、p は標数、a, b は曲線 $y^2 =$

$x^3 + ax + b$ の係数、$G = (x_G, y_G)$ は曲線上のベースポイント、n はベースポイントの位数、h は cofactor をそれぞれ表すものとする。

(1) 鍵長の確認

p は、そのビット長 $\lceil \log_2 p \rceil$ が $160, 192, 224, 256, 384, 521$ のいずれかであるような奇素数であることを確認する。セキュリティ強度が十分にあることを保証する。[*13]

(2) 曲線係数及びベースポイント座標の範囲確認

a, b, x_G, y_G は、

$$0 \leq a, b, x_G, y_G \leq p - 1$$

を満たす整数であることを確認する。曲線係数およびベースポイント座標が正しい範囲にあることを保証する。

(3) 係数の関係の確認

$$4a^3 + 27b^2 \neq 0 (\mathrm{mod}\, p)$$

であることを確認する。曲線が非特異であることを保証する。

(4) ベースポイント座標の確認

$$y_G^2 = x_G^3 + ax_G + b (\mathrm{mod}\, p)$$

であることを確認する。ベースポイントが曲線上にあることを保証する。

(5) 位数が素数であることの確認

n が素数であることを確認する。n が合成数の場合、Pohlig-Hellman 攻撃 [18] により、セキュリティ強度が低下する。

(6) cofactor の確認

$$h \leq 4, h = \lfloor (\sqrt{p} + 1)^2 / n \rfloor$$

であることを確認する。h が大きくなると n が小さくなり、セキュリティ強度が低下する。h の値が十分小さく、かつ正しいことを保証する。

(7) 位数の確認

$$nG = O$$

であることを確認する。位数が正しいことを保証する。

(8) 各種攻撃対策の確認

$$1 \leq \forall B < 20 \text{ に対し } p^B \neq 1 \bmod n, nh \neq p$$

であることを確認する。これらの条件が満たされないとすると、MOV 攻撃 [14] もしくは FR 攻撃 [6] もしくは SSSA 攻撃 [25, 20, 22] と呼ばれる攻撃のいずれかが効率的に適用できる。

2. 標数 2 の場合

以下、ドメインパラメータを $(m, f(x), a, b, G, n, h)$ とする。ただし、m は拡大次数、$f(x)$ は $\mathbb{F}_2[x]$ における m 次の既約多項式、$a, b \in \mathbb{F}_{2^m}$ は曲線 $y^2 + xy = x^3 + ax^2 + b$ in $\mathbb{F}_{2^m}$ の係数、

[*13] SEC1[19] の仕様では、$112, 128$ も p のビット長として認めているが、セキュリティ強度の観点からは推奨できない。

$G = (x_G, y_G)$ はベースポイント、n はベースポイントの位数、h は cofactor をそれぞれ表すものとする。

(1) 鍵長の確認

m が $163, 193, 233, 239, 283, 409, 571$ のいずれかであることを確認する。セキュリティ強度が十分にあることを保証する。[*14]

また、m が合成数の場合、Weil descent 攻撃 [7] が効率的に適用できる可能性があるため、この場合を排除している。

(2) 既約多項式の確認

$f(x)$ があらかじめ与えられている、$\mathbb{F}_2[x]$ における m 次の既約多項式であることを確認する。

(3) 曲線係数及びベースポイント座標の範囲確認

a, b, x_G, y_G が $\mathbb{F}_2[x]$ における $m-1$ 次以下の多項式であることを確認する。曲線係数およびベースポイント座標が正しい範囲にあることを保証する。

(4) 係数の関係の確認

$$\mathbb{F}_{2^m} \text{において } b \neq 0$$

であることを確認する。曲線が非特異であることを保証する。

(5) ベースポイント座標の確認

$$\mathbb{F}_{2^m} \text{において } y_G^2 + x_G y_G = x_G^3 + a x_G^2 + b$$

であることを確認する。ベースポイントが曲線上にあることを保証する。

(6) 位数が素数であることの確認

n が素数であることを確認する。n が合成数の場合、Pohlig-Hellman 攻撃 [18] により、セキュリティ強度が低下する。

(7) cofactor の確認

$$h \leq 4, h = \lfloor (\sqrt{2^m} + 1)^2 / n \rfloor$$

であることを確認する。h が大きくなると n が小さくなり、セキュリティ強度が低下する。h の値が十分小さく、かつ正しいことを保証する。

(8) 位数の確認

$$nG = O$$

であることを確認する。位数が正しいことを保証する。

(9) 各種攻撃対策の確認

$$1 \leq \forall B < 20 \text{ に対し } 2^{mB} \neq 1 \bmod n, nh \neq 2^m$$

であることを確認する。これらの条件が満たされないとすると、MOV 攻撃 [14] もしくは FR 攻撃 [6] もしくは SSSA 攻撃 [25, 20, 22] と呼ばれる攻撃のいずれかが効率的に適用できる。

[*14] SEC1[19] の仕様では、$113, 131$ も m の値として認めているが、セキュリティ強度の観点からは推奨できない。

曲線のランダム性の保証について

SEC1[19] では、seed を S とし、S から SHA-1 を使用することにより得られた出力値と曲線の係数 a, b とを関係づけることにより、曲線の係数がランダムに選択されたものであることを保証する手法 [2] を用いている。SHA-1 の出力値をコントロールすることは困難であるため、パラメータの中に S を併記することにより、SHA-1 の出力値と関連づけられた係数 a, b を共に作為的に選択したものではないことが保証される。

Kobilitz 曲線の安全性

Koblitz 曲線とは 2 の拡大体上の曲線で次式によって定義されるもので、anomalous binary curve(ABC 曲線) とも呼ばれている:

$$y^2 + xy = x^3 + ax^2 + 1(a \in \{0, 1\})$$

この曲線は定義体が $\mathbb{F}_{2^m}$ である $\mathbb{F}_2$ 上の曲線である。Koblitz 曲線の特徴として Frobenius 写像を用いることで点のスカラー倍演算を高速に計算できることがあげられる。なお、素体 $\mathbb{F}_p$ 上でも自己準同形写像が高速に計算できる曲線が存在し、SEC1 では、こうした曲線も含めて「Koblitz 曲線」として総称している。

Koblitz 曲線に固有の攻撃として、Wiener-Zuccherato[28] および Gallant-Lambert-Vanstone[8] が Koblitz 曲線上の離散対数問題に対して、ρ 法での並列衝突探索法の若干の高速化手法を示している。この手法では、Koblitz 曲線の特長である点演算の高速性を利用しており、$\mathbb{F}_{2^m}$ 上の Koblitz 曲線に対し、$\sqrt{2m}$ 倍高速に離散対数を求めることができる。具体的には m が 160 のときに離散対数の計算ステップ数が約 2^{76} に相当するため、通常の楕円曲線の場合と比べて約 16 倍高速に解けるが、計算量として大きな改善ではない。

上記の ρ 法の高速化手法以外には現状では固有の攻撃法は発見されていないものの、Koblitz 曲線はかなり限定された曲線のクラスであり、そのクラス特有の攻撃が発見される可能性には注意すべきである。

楕円曲線上の離散対数問題は「第 2.4.3 節」も参照されたい。

参考文献

[1] A.O.L. Atkin, F. Morain, Elliptic curves and primality proving, Math. Comp., 61, 29–67, 1993.

[2] ANSI X9.62-1998, Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm(ECDSA), American Bankers Association, 1999.

[3] N. D. Elkies, Elliptic and modular curves over finite fields and related computational issues, Computational perspectives on number theory (Chicago, IL, 1995), AMS/IP Stud. Adv. Math., **7**, 21-76, Providence, RI: AMS, 1998.

[4] Fouquet, M., Gaudry, P., Harley, R., An extension of Satoh's algorithm and its implementation, J. Ramanujan Math. Soc. 15 (2000) 281-318.

[5] Fouquet, M., Gaudry, P., Harley, R., Finding secure curves with the Satoh-FGH algorithm and an early-abort strategy, Advances in Cryptology - Eurocrypt 2001 (Innsbruck, Austria, May 2001), Lect. Notes in Comput. Sci., 2045, 14-29, ed. Pfitzmann, B., Berlin, Heidelberg Springer Verlag, 2001.

[6] G. Frey and H. -G. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, Mathematics of Computation, 62, 865-874, 1994.

[7] P. Gaudry, F. Hess and N. Smart, Constructive and Destructive Facets of Weil Descent on Elliptic Curves, Journal of Cryptology, Vol.15, No.1, pp19-46, 2002.

[8] R.Gallant, R.Lambert and S.Vanstone, “Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms”, Advances in Cryptology – CRYPTO2001, Lecture Notes in Computer Science **2139**, Springer-Verlag, pp.190–200, 2001.

[9] R. Harley, Counting points with the arithmetic-geometric mean (joint work with J.-F. Mestre and P. Gaudry), Eurocrypt 2001, Rump session, 2001.

[10] Kim, H., Park, J., Cheon, J., Park, J., Kim, J., Hahn, S., Fast elliptic curve point counting using Gaussian normal basis, Algorithmic number theory (Sydney, Australia, July 2002), Lect. Notes in Cimput. Sci., 2369, 292-307, ed. Fieker, C., Kohel, D., Berlin: Springer, 2002.

[11] N. Koblitz, CM-curves with good cryptographic properties, Advances in cryptology—CRYPTO ’91 (Santa Barbara, CA, 1991), Lecture Notes in Comput. Sci., **576**, 279-287, Berlin: Springer-Verlag, 1992.

[12] G.-J. Lay, H.G. Zimmer, Constructing elliptic curves with given group order over large finite fields, In ANTS-1, Algorithmic Number Theory, 250–263, Springer-Verlag, LNCS 877, 1994.

[13] B. Mazur, Rational points of Abelian varieties with values in towers of number fields. *Invent. Math.* **18** (1972) 183-266.

[14] A. Menezes, T. Okamoto and S. Vansone, Reducing elliptic curve logarithms to logarithms in finite fields, IEEE Trans. on Infor. Th., 39, 1639-1646, 1993.

[15] A. Menezes and S. Vansone, The Implementation of Elliptic Curve Cryptosystems, Proc. of AUSCRYPT’90, LNCS 453, 2–13, 1990.

[16] F. Morain, Building cyclic elliptic curves modulo large primes, In Advances in Cryptology, EUROCRYPT91, 328–336, Springer-Verlag, LNCS 547, 1991.

[17] V. Müller, Ein Algorithmus zur Bestimmung der Punktanzahl elliptischer Kurven über endlichen Körpern der Charakteristik grösser drei, (1995) Dissertation der Universität des Saarlandes.

[18] S. Pohlig and M. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE Trans. on Infor. Th., 24, 106-110, 1978.

[19] Standards for Efficient Cryptography, “SEC1:Elliptic Curve Cryptography”, Certicom Research, Ver.1.0, September 2000.

[20] P. N. Smart, The discrete logarithm problem on elliptic curves of trace one, J. Cryptology 12, 193-196, 1999.

[21] T. Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting, in J. Ramanujan Math. Soc. 15, 247-270, 2000.

[22] T. Satoh and K. Araki, Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves, COMMENTARII MATHEMATICI UNIVERSITATIS SANCTI PAULI, vol. 47, No. 1, 81-92, 1998.

[23] T. Satoh, B. Skjernaa, Y. Taguchi, Fast Computation of Canonical Lifts of Elliptic curves and its Application to Point Counting, (2001) preprint.

[24] B. Skjernaa, Satoh's algorithm in characteristic 2, Math. Comp. **72** (2003), 477-487.

[25] I. A. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curves in characteristic p, Math. Comp. 67, 353-356, 1998.

[26] R. Schoof, Elliptic curves over finite fields and the computation of square roots mod *p*. *Math. Comp.* **44** (1985) 483-494.

[27] Vercauteren, F., Preneel, B., Vandewalle, J., A memory efficient version of Satoh's algorithm, Advances in Cryptology - Eurocrypt 2001, Lect. Notes in Comput. Sci., 2045, 1-13, ed. Pfitzmann, B., Berlin, Heidelberg: Springer Verlag, 2001.

[28] M. J. Wiener and R. J. Zuccherato,"Faster attacks on elliptic curve cryptosystems", Selected Areas in Cryptography, Lecture Notes in Computer Science **1556**, pp.190–200, 1999.

第 3 章

共通鍵暗号技術の評価

3.1　評価方法

3.1.1　安全性評価方法

暗号の一番重要な評価尺度は、安全性である。暗号の安全性には、情報量的安全性と計算量的安全性の 2 種類が存在する。情報量的安全性は Shannon によって示された理論であり、いかなる計算機能力を費やしても、情報量の面で、解読不可能であることを示す安全性である。しかし、これを満足するために、暗号化する平文の量以上の秘密鍵を毎回費やさねばならず、現実的ではない。そのため、暗号の安全性は情報量的安全性ではなく、後者の計算量的安全性で示される。最良の解読アルゴリズムと最速の計算機能力を費やしても、現実的に解読が不可能である事を示すのが、計算量的安全性である。計算量的安全性は、秘密の鍵を推定する困難さの度合いとも言い換えることができる。しかしながら、計算量的安全性を示す絶対的な評価手法は現在のところ存在しない。

共通鍵暗号の計算量的安全性においては、現在知られている各種の攻撃法を適用しそのときに必要とするコスト (計算量、データ量、メモリ量) を求め、総合的に安全性を評価する。攻撃法には、大別して、

- 全数探索
- ショートカット法

がある。

また、攻撃者が利用可能な情報の条件によって、攻撃方法は一般に以下のように分類される。

- 暗号文単独攻撃
  暗号文及び平文の統計情報が利用できる場合。
- 既知平文攻撃
  平文と対応する暗号文が利用できる場合。攻撃者は、何らかの方法により入手した平文/暗号文対を用いて攻撃を行う。

- 選択平文攻撃
  攻撃者が任意に選択した平文とそれに対応した暗号文が利用できる場合。既知平文攻撃との違いは、攻撃者が対象となる暗号器をコントロールでき、攻撃に都合の良い平文/暗号文対を選んで利用できる点にある。

利用できる情報により攻撃の難易が存在し、後者ほど攻撃者に有利となる。

暗号文単独攻撃で解読可能な暗号は、一般に安全ではないといえる。また、攻撃者が無限の計算量を持つと仮定すれば、ある程度の既知平文を利用すると鍵の全数探索により必ず秘密鍵を発見できる。従って十分な量の暗号文に対応する平文が既知である、という条件を持つ攻撃 (既知平文攻撃や選択平文攻撃) に関して、ある攻撃方法が必要とするコストが、鍵の全数探索の場合よりも少なくて済む時、その攻撃が (学術的に) 成功すると判断する。具体的には k ビットの秘密鍵を用いる暗号が、2^k よりも少ないコストで攻撃可能な解読法が発見された時、その暗号は (学術的に) 安全ではないと判断する。

■全数探索法　全数探索法は、1 組 (又は数組) の平文と暗号文の対を使い、やみくもに秘密鍵を総当たりする手法である。秘密鍵のビット数を k とすれば、2^k の計算量で秘密の鍵が求まる。DES は実効鍵長 56 ビットであり、DES 制定時は 56 ビットの鍵長の全数探索は現実的には不可能と考える見方もあった。しかしながら、それから 20 年以上経過した 1999 年 1 月、DES Challenge III という解読コンテストでは 22 時間で鍵の総当たりによる秘密鍵導出が成功し、56 ビット程度の総当たりは現実的なものとなった。さらに、Secret-Key Challenge(1997 年から開始された RC5 解読コンテスト。ユーザ鍵 (64 ビット) の全数探索による解読に約 4 年要した。2002 年 9 月 26 日に解読成功と発表。) などに見られる現状や、コンピュータ処理性能の向上の速さを考慮すると、64 ビット鍵の 2^{64} 程度の計算時間は、現時点で十分処理可能と見積もられる。したがって、2^k ($> 2^{64}$) よりも少ない計算量で解読可能であっても実用上問題ない場合も無いとはいえないが、暗号としての使用は薦められない。

さらに、将来の計算機能力の進歩を年率 2 倍の速度向上と多めに見込み、解読アルゴリズムの進歩やシステム交換に要する期間を考えるならば、10 年程度の期間安全に使用するためには、2^{100} 以上の解読計算量が望ましい。AES を含め、最近の共通鍵暗号の秘密鍵ビット数は、128 ビット以上となっている。CRYPTREC においても、128 ビット以上の秘密鍵ビット数に相当する 2^{128} 以上の解読計算量で安全と判断している。

■ショートカット法　ショートカット法は、公開されている暗号アルゴリズムの内部の解析に基づき、解読計算を効率化し、2^k 未満の計算量で、秘密鍵 (又は実質的な秘密鍵) を導出する手法である。すなわち、暗号アルゴリズムに内在する、瑕疵を見いだし総当たり攻撃よりも少ない計算量で、解読を目指すものである。暗号の強度評価を目的としており、攻撃者に有利な攻撃条件設定である選択平文攻撃まで許される。そこでは、任意に選択した平文とそれに対応した暗号文が解読に利用できる。言い換えるならば、攻撃者は対象となる暗号器をコントロールでき、解読に都合の良い平文/暗号文対を選ぶことが許される。ショートカット法で 2^k ($> 2^{64}$) よりも少ない計算量で解読可能な暗号であっても実際のシステムにおいては、攻撃条件が必ずしも満足され

ず、直ちに使用を禁止する必要がない場合もありうるが、CRYPTREC としては、その使用は薦めない。

攻撃方法には、あるカテゴリに含まれる暗号すべてに適用可能な汎用的な攻撃方法と、ある暗号に特化した攻撃方法がある。共通鍵暗号は、64 ビットブロック暗号、128 ビットブロック暗号、ストリーム暗号に分類されるので、それぞれの攻撃方法に対して、3.2.1 節に 64 ビットブロック暗号、3.2.2 節に 128 ビットブロック暗号、3.2.3 節にストリーム暗号に関する安全性評価について記す。

これらの攻撃を考え、現時点の最良の解読技術を適用しても、秘密鍵の総当たりである 2^{128} 以上の計算量が必要と判断されるものを、CRYPTREC では、安全と判断している。

3.1.1.1 ブロック暗号

ブロック暗号に対して、今回の評価では、汎用的な攻撃方法として以下に対する強度を評価した。

- 差分攻撃
- 線形攻撃
- 高階差分攻撃

さらに、出力の統計的性質を評価するアバランシュ性評価も実施した。

■差分攻撃・線形攻撃に対する安全性 差分攻撃は、Biham と Shamir によって 1990 年に提案された DES に対する攻撃方法であるが、ブロック暗号全体に適用可能な汎用的攻撃方法でもある。2 組の平文/暗号文組に対し、平文同士の差と暗号文同士の差に相関がある時に適用可能な選択平文攻撃の一つである。線形攻撃は、差分攻撃と同様に、1993 年に三菱電機の松井によって提案された DES に対する攻撃方法であると同時に、ブロック暗号全体に適用可能な汎用的攻撃方法でもある。特定の入力ビットの排他的論理和と出力ビットの排他的論理和に相関がある時に適用可能な既知平文攻撃の一つである。

これらの攻撃法に対する耐性は、最大差分確率・最大線形確率で与えられ、この確率が十分小さければ安全と判断される。しかしながら、最大差分確率・最大線形確率の真値を求めることは困難であるので、それに準ずる最大差分特性確率・最大線形特性確率を用いる場合もある。これらは、

- 構成部品ごとに評価を行い、特性確率の上界を求める方法
- 計算機探索より求める方法

などによって見積もられる。

■差分攻撃・線形攻撃に対する証明可能安全性 暗号によっては、差分攻撃・線形攻撃に対する安全性を証明可能安全性の議論で示している場合もある。Nyberg は、1992 年に、Feistel 型構造

のブロック暗号に対して、ラウンド関数の最大差分確率が p であるとき、段数が 4 段以上で構成されていれば、その暗号全体の最大差分確率は $2p^2$ 以下であることを数学的に証明した。その後、線形攻撃に対しても同様の指標を与え、差分攻撃・線形攻撃に対する証明可能安全性としてまとめた。さらに、松井や青木らによって、より高度な議論へと発展した。数学的に安全性を証明できる手法であるが、差分攻撃と線形攻撃に対してのみ有効な議論であることを留意されたい。

■高階差分攻撃に対する安全性　高階差分攻撃法は 1994 年に Lai によって示され、Knudsen と Jakobsen が 1997 年に実験的ブロック暗号である $\mathcal{KN}$ 暗号への攻撃で利用した。出力の高階差分値が、平文固定値と拡大鍵によらない定数となる時に適用可能な選択平文攻撃の一つである。$\mathcal{KN}$ 暗号は、上記の差分攻撃・線形攻撃に対する証明可能安全性を有するが、ラウンド関数の代数次数が小さいことから、高階差分攻撃で解読可能であることが示された。

攻撃の効果は用いる階数に依存し、小さいほどコストが少ない。一方、出力の代数次数は平文のどのビットを変数とするかに依存するので、攻撃に必要な最小階数は変数ビットの選び方で決定される。しかしながら、最適な選択方法はまだ存在しない。N ビット入出力の暗号の場合、入力ビットをすべて変数としても出力の代数次数は N を超えないが、一般には、出力ブロックの形式的な代数次数が N より大きくなった時、高階差分攻撃に対して安全であると判断する。

■アバランシュ性評価　アバランシュ性評価とは、入力に特定の差分値を与えた場合の出力差分値について、出力ビット位置ごとに差分の出現頻度を調査する評価で、出力ビット位置ごとの挙動を知ることができる。この評価方法は、アルゴリズムをブラックボックス的に扱い、さらに評価量を数値化することで、構造の違いによらない統一的な比較を可能にしている。

鍵スケジュール部を含む暗号化処理、鍵スケジュール部単体およびラウンド関数単体における入出力を対象とし、差分の出現頻度、差分の拡散量、差分出現状態の相関係数、有効鍵量の項目について調査した。

すべてのブロック暗号は

- データランダム化部 (ラウンド関数の集合)
- 鍵スケジュール部

から構成されていると言ってよい。そこで、アバランシュ性の評価として、

- ラウンド関数単体
- データランダム化部
- 鍵スケジュール部単体
- 鍵スケジュール部を含む暗号化処理全体

の評価を行なった。

⟨⟨ **評価項目** ⟩⟩

アバランシュ性の評価項目として取り上げたものは、以下のとおりである。

- AVA (差分の出現頻度)
  出力差分値が 1 となる頻度と 0 となる頻度の差。今回の調査では、入力差分 ΔX、鍵差分 ΔK のハミング重みが $m = 1, 2$ について実施。
- AVD (差分の拡散値)
  出力差分値のハミング重みの平均値。
- CC (差分出現状態の相関係数)
  出力差分値の i ビット位置と j ビット位置での相関係数。
- UKV (有効鍵量)
  ハミング重み 1 の鍵差分値を与えた時の AVA のうち相対基準値を満たしている評価値の割合。

また、ブロック暗号の構成要素および全体システムの統計的性質の調査対象には大別して

- 入力と出力との相関
- 鍵と出力との相関

の 2 種類がある。

⟨⟨ **統計検定の手法** ⟩⟩

こうした統計的データ検定を行なうためには、擬似乱数生成関数が必要であり、さらに擬似乱数生成関数を用いて乱数列生成を行なう必要がある。こうして得られたデータを収集分析して以下の規範で検定を行なう。

- AVA の最悪偏差率が相対基準値以下になれば統計的な偏りが無いものとみなす
- AVD が $n/2$ に近いほど統計的偏りが少ないとする (n: 出力データ長)
- CC の絶対値は 1 以下であるが、0 に近づくほど独立性が高いと判断する
- UKV は鍵データ長に近いほど望ましい

3.1.1.2 ストリーム暗号

ストリーム暗号は、一般的には、擬似乱数生成器からの出力を鍵系列とし、その初期値を秘密鍵とする。ここでは、出力系列の統計的性質の評価に、FIPS PUB 140-2 に記載されている、以下の方法等を採用した。

- 長周期性
- 線形複雑度
- 0/1 等頻度性
- モノビットテスト

- ポーカーテスト
- ランテスト
- ロングランテスト

これらの評価は、あくまでも統計的な性質を調査するのみなので、これらの統計的な性質についての評価が良いだけでは暗号的に安全であるとは言いがたい。そこで、Divide-and-Conquer attack, Correlation attack 等、汎用的な攻撃方法に対する耐性を、ブロック暗号と同様に評価した。

3.1.2　ソフトウェア実装評価方法

暗号は安全面だけでなく、使用状況を想定し実装面も考慮する必要がある。電子政府における暗号実装に対する要求事項は現在のところ不明であるが、ソフトウェア実装の評価においては、評価時点で一般的と思われる PC 環境、現時点で最も普及していると思われるサーバ環境、高性能を実現しているハイエンド環境の 3 つの環境を想定した。一般的な PC 環境は、すべての応募暗号が実装を想定している環境でもあるので、これに関する評価はすべての (応募) 暗号に対して行った。

ブロック暗号については、残りのサーバ環境とハイエンド環境について、その暗号の設計思想を尊重して応募者の選択としたうえで、評価を実施した。また、8 ビット CPU 等のロースペック環境 (IC カード環境) については、128 ビットブロック暗号の一部について、Z80 シミュレータ上での評価を実施した。

なお、ハッシュ関数と擬似乱数生成系に対してはソフトウェア評価を実施しなかった。

実際の評価に際しては、応募者が実装した暗号プログラムを利用して、委員の立会いで測定を行った。測定は同一の評価用ハードウェア、評価用プログラムを用い、できる限り公平な条件で行った。なお、文献によっては本報告書に記載されている数値と異なる場合があるが、これは測定プログラムや測定環境の違いに起因する。また同一のハードウェア、測定プログラムであっても、オペレーティングシステムや常駐プログラムの組み合わせなどにかなり影響される。したがって、この数値が必ず実現されるわけではないことに注意されたい。本ソフトウェア実装評価は、(応募) 暗号間の処理速度の傾向の比較が目的である。

使用したハードウェア環境を表 3.1 に示す。

■ブロック暗号での評価方法　ブロック暗号でのソフトウェア実装評価では、IC カード環境を除く各測定環境において、

- データランダム化部処理速度
- 鍵スケジュール部 + データランダム化部処理時間

の 2 種類の測定を実施した。また、IC カード環境においては、鍵スケジュール部 + データランダム化部処理時間の測定を実施した。

表 3.1: ソフトウェア実装評価に使用した環境

PC 環境	
CPU	Pentium III (650MHz)
OS	Windows98 SE
搭載メモリ	64MB
コンパイラ	Visual C++ Ver6.0 SP3
サーバ環境	
CPU	UltraSPARC IIi (400MHz)
OS	Solaris 7
搭載メモリ	256MB
コンパイラ	Forte C 6
ハイエンド環境	
CPU	Alpha 21264 (463MHz)
OS	Tru64 UNIX V5.1
搭載メモリ	512MB
コンパイラ	DEC C
IC カード環境	
CPU	Z80 (5MHz)
記述言語	アセンブリ
備考	指定のパッチを当てた Z80 シミュレータを PC 環境上で動作

データランダム化部処理速度の測定は、例えば 64 ビットブロック暗号の場合、64 ビットの平文を暗号文に変換するのに必要な CPU のクロック数 (CPU の動作周波数に依存しない計算量) をカウントして行った。128 ビットブロック暗号の場合は 128 ビットの平文の暗号文への変換に対して同様のカウントを行った。この測定では、1MB の平文 (暗号文) に対して鍵を設定し、暗号化 (復号) を行い測定をした。したがって、鍵セットアップ (鍵スケジュール部) にかかる計算相当量は無視できる。1 回の測定で、1MB の暗号化 (復号) を 128 回行い最速値と平均値を採取した。測定は 3 回行った。

鍵スケジュール部 + データランダム化部処理時間の測定では、鍵セットアップから 1 ブロックデータ (64 ビットまたは 128 ビット) の暗号化 (復号) が終了するまでに必要な CPU のクロック数をカウントして行った。この測定では、1 ブロックの暗号化 (復号) ごとに鍵のセットアップを行った。その他の測定条件はデータランダム化部処理速度の測定と同じである。この条件で測定した処理時間は、認証にブロック暗号を用いる場合などに参考となるデータである。なお、上述したデータランダム化部処理速度の値をこの値から引いても、実装方法の違いにより、鍵スケジュール部そのものの処理速度とはならない場合もある。

IC カード実装においては、ROM や RAM などの使用メモリ量も実装上の重要なファクタとなるので、処理時間の測定のほかに、各使用メモリ量もあわせて測定した。

■ストリーム暗号での評価方法　ストリーム暗号でのソフトウェア実装評価では、PC 環境において、

- 暗号化・復号処理速度 (擬似乱数生成を含む)
- 鍵セットアップ処理時間

の 2 種類の測定を実施した。

暗号化・復号処理速度の測定では、1MB の平文 (暗号文) に対して鍵を設定し、暗号化 (復号) するのに必要な CPU のクロック数 (CPU の動作周波数に依存しない計算量) をカウントして測定した。1 回の測定で、1MB の暗号化 (復号) を 128 回行い最速値と平均値を採取した。測定は 3 回行った。この場合、ブロック暗号のときと同様、鍵セットアップにかかる計算相当量は無視できる。なお、暗号化・復号処理には、実際の暗号化 (復号) 処理のほか、処理に用いる鍵値を生成する擬似乱数生成処理を含む。

鍵セットアップ処理時間の測定では、暗号化 (復号) するデータを 128 ビットとした時の鍵セットアップ処理時間を計測した。1 回の測定で、鍵セットアップを 128 回行い、最速値と平均値を採取した。測定は 3 回行った。

3.1.3　ハードウェア実装評価方法

2000 年度のハードウェア実装評価では、一部の共通鍵暗号アルゴリズムに対して、応募書類に記載されているシミュレーション結果に基づくハードウェア実装情報の妥当性の検証を実施した。

2002 年度のハードウェア実装評価では、「応募書類 (アルゴリズム仕様書およびテストベクタ)」だけの情報から第三者によるハードウェア実装が可能であることを確認する目的で、各共通鍵暗号アルゴリズムの実装検証を行った。

回路設計のターゲットデバイスとしては、ASIC (Application Specific Integrated Circuit) と FPGA (Field Prgramble Gate Array) の二つに大別することができる。 ASIC 実装の場合、同じ回路設計データを用いても、その性能は半導体プロセスやライブラリに大きく依存する。また、実機での動作確認には大きな製造コストと長い製造期間を要するため、CAD 上のシミュレーション結果を元に評価せざるを得ない。一方、FPGA では各アルゴリズムの性能を同一の IC 上で評価することが可能であり、実機上での動作確認を比較的短期間で行なうことができるというメリットがある。

そこで 2002 年度は、12 種類の共通鍵暗号アルゴリズムに対して同一 FPGA によるハードウェア実装評価を行なった。FPGA は 2002 年 6 月現在入手可能で、大規模な回路が実装可能な Xilinx 社の XC2V6000 をターゲットデバイスとし、この FPGA を搭載した市販の評価環境を使用した。評価環境の概要を表 3.2 に、FPGA の概要を表 3.3 に示す。

なお、回路記述言語 XST Verilog を用いて設計し、シミュレーションは ModelSim XE 5.5e を、

論理合成は ISE Foundation4.2i を用いた。

表 3.2: ハードウェア実装評価に使用した FPGA 開発環境

カードサイズ		106 mm × 312 mm (PCI Full サイズ)
論理規模		12.3 M システムゲート (メーカー公称値)
搭載 FPGA	PCI 用	XCV300 × 1 石
	内部用	XC2V6000 × 2 石
搭載 RAM		2.14 Gbit SDRAM メモリ (SODIMM スロット ×2)
		256 Mbit SDRAM メモリ (256 Mbits×1 石構成)
		5.3 Mbit FPGA 内蔵超高速メモリ

表 3.3: 評価に用いた FPGA の概要 [公称値]

ターゲットデバイス	Xilinx XC2V6000
ロジックセル	76,032
システムゲート	6M
CLB	8,448 (33,792 slices) (67,854 LUTs) (1,081,344 bits 分散 RAM) (67,584 FFs)
18×18-bit 乗算器	144
18 Kbit Block RAM	144 (2,592 Kbits)

CLB: Configurable Logic Block の略、構成可能な論理ブロック、1 CLB = 4 slices =128-bit 分散 RAM
LUT: Look Up Table の略、組合せ論理を実現する機能ブロック
FF: Flip-Flop の略

■ブロック暗号での評価方法 ブロック暗号の実装にあたって、データランダム化部は、基本的にラウンド関数 1 段分の回路を用意して、それを段数分繰返し使用するループ・アーキテクチャを採用した。また、鍵スケジュール部は、on-the-fly アーキテクチャ (拡大鍵生成と暗号化処理を並列実行するアーキテクチャ) が使用可能なアルゴリズムについては、on-the-fly アーキテクチャを採用した。その他の注意点は以下のとおりである。

- 暗号化機能に対するデータランダム化部と鍵スケジュール部だけを実装し、復号機能は有していない。
- アルゴリズムによってはデータランダム化部 (の一部) と鍵スケジュール部 (の一部) とで共用できる構成となっているものもあるが、本実装では両者を独立に設計している。
- ブロック暗号の主要コンポーネントである S-box は、基本的に入出力の関係を記述したルックアップ・テーブル実装である。
- 128 ビットブロック暗号においては、鍵長 128 ビットだけをサポートした。

今回の実装は動作検証が主目的であるため、各アルゴリズムの性質を考慮した特別な回路規模縮小や動作速度向上などの工夫は行なわず、ストレート・フォワードなアーキテクチャをとって

いる。そのため、必ずしも最適な実装が行なわれているとは言えず、今回の結果に基づいて各アルゴリズムの回路実装効率を公平に比較評価することはできない。従って、Triple DES との相対比較はもとより、回路規模と動作速度の数値も公表しないこととした。ただし、いずれの暗号アルゴリズムも上記 FPGA 開発環境において 33MHz 動作を確認している。なお、評価回路は Verilog-HDL 言語により記述し、Xilinx 社製の論理合成ツール ISE Foundation 4.2i を用いて、Xilinx 社製を用の FPGA XC2V6000 上への実装を行なった。

個別暗号のハードウェア実装評価の項で表に示した 1 ブロックの暗号化に要するサイクル数は実測によるものであり、FPGA に搭載されているハードウェアマクロの乗算器を使用しているものは 1 ラウンド/サイクルで実行できないため、サイクル数が多くなる傾向にある。従って、サイクル数は暗号のラウンド数と必ずしも一致しない。

■ストリーム暗号での評価方法　ストリーム暗号のハードウェア実装評価においては、アルテラ社の FPGA 上で、C 言語で作成されたプログラムから、Verilog-HDL により回路記述し、シミュレーションを行った。ストリーム暗号は、回路で実現することが多いので、論理合成時のオプションは速度優先とした。

3.2　評価結果総評

3.2.1　64 ビットブロック暗号

対象は、CIPHERUNICORN-E、Hierocrypt-L1、MISTY1 および Triple DES の 4 種類である。CIPHERUNICORN-E から MISTY1 までの応募があり、Triple DES はその他評価が必要な暗号として 2000 年度から評価対象として追加した。評価概要を表を含め以下に示す。記述内容は以下のとおり。

■特徴　提案組織、暗号発表年、構造上の特徴、データランダム化部で使用する演算等の特徴を載せた。なお、段数等可変パラメータを持つ暗号については、本公募への提案者の推奨値を記した。

■安全性　3 つの観点 (差分攻撃・線形攻撃に対する耐性、代数的およびその他攻撃に対する耐性、アバランシュ性) から記述する。

- 差分攻撃・線形攻撃に対する耐性では、汎用的な確率的攻撃方法である差分攻撃・線形攻撃に対する強度評価指標として、最大差分確率・最大線形確率、または最大差分特性確率・最大線形特性確率を示す。
- 代数的およびその他攻撃に対する耐性では、高階差分攻撃や補間攻撃、Square 攻撃等の代数的手法に対する耐性や鍵関連攻撃、mod n 攻撃等、その他の攻撃に対する耐性を述べる。高階差分攻撃や補間攻撃の評価は、暗号系の基本的弱点を代数的観点から探る手法であり、段数が大きい場合、通常これらの手法の攻撃で問題になることは少ない。ただし、

そこで得られた弱点は、他の攻撃手法と組み合わせが可能な場合、最終的な暗号強度に影響を与える可能性がある。

- アバランシュ性は、暗号系におけるデータ撹拌の様子を統計的に捉えるものであり、通常は、直接、解読に結びつくことは無いが、暗号の部分関数の弱点を探る際の糸口を与える。

■ソフトウェア実装評価 本評価は 2000 年度に行われた。暗号は安全面だけでなく、使用状況を想定し実装面も考慮する必要がある。電子政府における暗号実装に対する要求事項は現在のところ不明であるが、ソフトウェア実装の評価においては、評価時点で一般的と思われる PC 環境 (必須)、現時点で最も普及していると思われるサーバ環境 (選択)、高性能を実現しているハイエンド環境 (選択) の 3 つの環境を想定した。データランダム化部処理速度と鍵スケジュール部 + データランダム化部処理時間の 2 種類の測定を実施した。

■ハードウェア実装評価 2002 年度のハードウェア実装評価では、「応募書類 (アルゴリズム仕様書およびテストベクタ)」だけの情報から第三者によるハードウェア実装が可能であることを確認する目的で、FPGA 環境において各アルゴリズムの実装検証を行った。

今回の実装は動作検証が主目的であるため、各アルゴリズムの性質を考慮した特別な回路規模縮小や動作速度向上などの工夫は行なわず、ストレート・フォワードなアーキテクチャをとっている。そのため、必ずしも最適な実装が行なわれているとは言えず、今回の結果に基づいて各アルゴリズムの回路実装効率を公平に比較評価することはできない。従って、Triple DES との相対比較はもとより、回路規模と動作速度の数値も公表しないこととした。ただし、いずれの暗号アルゴリズムも FPGA 実装環境において 33MHz 動作を確認している。FPGA 実装環境の概要については、「3.1.3 ハードウェア実装評価方法」を参照されたい。

■総合評価 安全性および実装評価を総合した観点から、評価結果を表 3.4 に示す。なお、UNI-E は CIPHERUNICORN-E、HC-L1 は Hierocrypt-L1 のことである。

3.2.1.1 安全性評価結果の総評

■差分攻撃・線形攻撃に対する耐性 差分攻撃・線形攻撃に対する耐性は最大差分確率・最大線形確率で与えられる。この確率が安全性保証の観点から評価されているのは MISTY1 と Hierocrypt-L1 である。MISTY1 は 3 段で 2^{-56} 以下であり、差分攻撃や線形攻撃には十分安全と考えられる。Hierocrypt-L1 も 2 段でこれらの確率として 2^{-48} 以下が保証されている。この保証を差分攻撃・線形攻撃に対する証明可能安全性という。

最大差分確率・最大線形確率の真値を求めることは困難であり、それに準じた指標として最大差分特性確率・最大線形特性確率がある。最大特性確率の評価は、

- 構成部品の最大差分確率・最大線形確率をもとに特性確率の上界を出す方法
- 計算機探索により最大特性確率を求める方法

表 3.4: 64 ビットブロック暗号の評価結果総評

UNI-E	特徴
	NEC (1998) Feistel 型構造、16 段。ラウンド関数は複雑。安全性を高める意図で本流部と一時鍵生成部で構成。ラウンド関数は S-box を基本部品とし、T、K、Y 関数で構成。S-box は 8×8 の 4 種類。GF(2^8) 上の逆数演算をベースに設計し、差分攻撃・線形攻撃に耐性。テーブル参照、加算、EXOR、AND、シフト演算。暗号評価支援システムで、有意な相関が見られないようにラウンド関数構造の設計。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は遅いグループである。
HC-L1	特徴
	東芝 (2000) 入れ子型 SPN 構造、6 段。各段は XS 関数の 2 並列および P 層で構成。XS 関数は、P 層を 4 並列の S-box 2 層で挟んだ構造。S-box は 8×8 の 1 種類。GF(2^8) 上のべき乗演算をベースに設計し、差分攻撃・線形攻撃に耐性。テーブル参照、EXOR、AND。入れ子型 SPN 構造の採用により安全性と計算効率との両立。P 層設計では、活性 S-box 数の考えで評価。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
MISTY1	特徴
	三菱電機 (1996) Feistel 型構造、8 段。2 段ごとに FL 関数を挿入。ラウンド関数の内部構造で変形 Feistel 型構造を再帰的に使用。S-box は 7×7 および 9×9 の 2 種類。拡大体上のべき乗演算をベースに設計し、差分攻撃・線形攻撃に耐性。ハードウェア実装を考え、低い代数次数。テーブル参照、EXOR、AND、OR。差分攻撃・線形攻撃に対する証明可能安全性。次世代携帯電話用 KASUMI 暗号の源。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
Triple DES (3-Key)	特徴
	IBM (1979) DES を 3 回繰り返した組み合わせ暗号。 DES は 1977 年に FIPS PUB 46 で規格化。DES の構造は Feistel 型構造、16 段。S-box は 6×4 の 8 種類。ランダムに構成した S-box から、ある評価基準で選択。テーブル参照、EXOR、巡回シフト演算。ハードウェア志向の設計。DES は 25 年以上経つ歴史的暗号であり、現代暗号のルーツだが、2004 年に FIPS PUB 46-3 から廃止される (Triple DES は 2004 年以降も FIPS PUB 46-4 として継続する予定)。今後は、AES (FIPS PUB 197) に引き継がれる予定。
	総合評価
	安全性について、FIPS 等で保証されている間は、問題ないと考える。

がある。

特性確率の上界として評価されているのが Hierocrypt-L1 と CIPHERUNICORN-E である。前者は 2 段で差分特性確率・線形特性確率が 2^{-90} を超えないことが示されている。

CIPHERUNICORN-E は、そのラウンド関数が複雑な構造であるため、解析が難しい。自己評価書では、差分特性確率・線形特性確率が、ラウンド関数を簡略したものを用いて見積もられている。2001 年度は、この簡略化の正当性を検証し、適切な見積もり方を検討した。その結果、ラウンド関数の評価については自己評価書に記載されている結果と異なる結果が導かれたが、仕様段数である 16 段の CIPHERUNICORN-E は、差分攻撃・線形攻撃に対して十分な安全性を持つと考えられる、という結論が得られた。

64 ビットブロック暗号のこれら特性確率が 2^{-64} 以下になることを安全性の証とする手法を、差分攻撃・線形攻撃に対する実際的安全性保証という。以上のように、評価対象となった 64 ビットブロック暗号はすべて、学術的な差分攻撃・線形攻撃に対する耐性が保証されている。

■代数的およびその他の攻撃に対する耐性 高階差分攻撃や補間攻撃においては、暗号化関数のガロア体 GF(2) または適切な拡大体における展開式を使って攻撃を行う。一般的に、これらの攻撃は段数の多い暗号に適用することは困難であるが、差分攻撃・線形攻撃に対する耐性を強く意識して設計された暗号に対しては、相対的に、これらの攻撃が効果的である場合が多い。高階差分攻撃に対する耐性は暗号化関数の代数次数で与えられるが、入力変数の取り方や着目する出力変数の選び方によりこの次数や個数は変わり、すべての可能性を尽くしてそれらの最小値を求めることは計算量的に不可能である。CRYPTREC では、暗号化関数の構成部品に着目し、その代数次数を基にした形式的代数次数の評価を行うとともに、一つの S-box 入力を変数とした場合について次の二つの評価を行った。

- 1 つの S-box 入力を変数とする 8 階以下の高階差分攻撃に対する耐性。
- S-box の全単射性に基づく高階差分攻撃に対する耐性 (SQUARE 攻撃耐性)。

結果として、いずれのアルゴリズムも提案段数においては、これらの攻撃に対して耐性を持つことを確認した。高階差分攻撃を適用することで、差分攻撃・線形攻撃に比べ、より高段数まで攻撃が可能となる暗号は、Hierocrypt-L1 と MISTY1 である。Hierocrypt-L1 は、32 階の高階差分攻撃 (32 階の SQUARE 攻撃) で平文組数 2^{37}、計算量 2^{117} を使い 3.5 段まで攻撃可能である。FL 関数なしの変形 MISTY1 の場合、7 階の高階差分攻撃により、平文組数 2^{11}、計算量 2^{93} で 6 段まで攻撃可能である。MISTY1 ではそれが、平文組数 2^{22}、計算量 2^{33} で 5 段までとなる。

補間攻撃 (またはそれを一般化した線形和攻撃) に対する耐性は、暗号化関数を補間多項式で表したときの未知の補間係数個数で与えられる。しかし、入力変数の取り方および着目する出力変数によりその個数は変わり、すべての可能性を尽くすことは計算量的に不可能である。CRYPTREC では、平文を 8 ビット単位の小ブロック 8 個 (64 ビットブロック暗号) に区切り、その小ブロックをガロア体 GF(2^8) の多項式基底で表現した場合について線形和攻撃に対する耐性を評価した。いずれのアルゴリズムも鍵の全数探索より効率のよい解読方法は発見されていない。

Triple DES (3-key) は、組み合わせ暗号であることに着目した中間一致攻撃により、学術的には 2^{56} の選択平文と $2^{108.2}$ の計算量で解読可能であるが、現実的な意味では安全と考えられる。

その他、カイ 2 乗攻撃、不能差分攻撃、ブーメラン攻撃、mod n 攻撃、非全単射攻撃等について、現在のところ、どのアルゴリズムも実用的観点から安全性に関する問題点は報告されていない。

■アバランシュ性評価 本評価は 2000 年度に行われた。「鍵スケジュール部を含む暗号化処理全体」では、すべてのアルゴリズムが期待値を満たした。しかし「鍵スケジュール部単体」では、Hierocrypt-L1、MISTY1 で期待値を満たさない部分を検出した。一方、「ラウンド関数部単体」でも、Hierocrypt-L1、MISTY1 で期待値を満たさない部分を検出した。

表 3.5: 64 ビットブロック暗号のアバランシュ性評価結果

UNI-E	データランダム化部では 4 段以降の攪拌に特徴は見られない。鍵スケジュール部では特徴は見られない。ラウンド関数では特徴は見られない。
Hierocrypt-L1	データランダム化部では 2 段以降の攪拌に特徴は見られない。鍵スケジュール部では秘密鍵と拡大鍵間に大きな関係が存在する。ラウンド関数では期待値から離れている部分がある。
MISTY1	データランダム化部では 4 段以降の攪拌に特徴は見られない。鍵スケジュール部では秘密鍵と拡大鍵間に大きな関係が存在する。ラウンド関数では期待値から離れている部分がある。

3.2.1.2 ソフトウェア実装評価の総評

本評価は 2000 年度に行われた。記載されている数値は、2000 年度に測定された数値である。

■データランダム化部処理速度 1MB の平文 (暗号文) に対して鍵を設定し、暗号化 (復号) を行ったときの 1 ブロック (64 ビット) あたりの処理時間を測定した。測定値は [clocks/block] だが、分かりやすいように、処理速度 [Mbps] に変換した。この値が大きいほど高速である。測定値は実行環境にかなりの影響を受けるので、この値が必ず実現されるとは限らない。また、測定プログラムの誤差や前述の変換等による誤差が生じている。さらに、測定プログラムの主旨を違えない程度の改変 (後述) を加えるのみで、測定値が変わる場合もある。したがって、この表の値のみで断定するのは危険である。各測定値欄に下段にも値が記載されているものは、応募者による測定プログラムの改変した場合の測定値である。測定プログラムはすべての応募暗号に同一の条件を与えるため、メモリ領域を多めに確保している。ここでいう改変とは、多めのメモリ領域を各暗号ごとに最適化した場合のことである。この改変は、

- 実際の実装状況により近いこと
- メモリ領域の大きさが速度に与える影響の原因が不明なこと

を考慮し、今回の報告書では両方の値を記載することにした。

表 3.6: PC 環境 [Pentium III (650MHz)]

64 ビットブロック暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
CIPHERUNICORN-E	29.0	(28.9)	29.3	(29.2)
Hierocrypt-L1	209.0	(207.0)	203.9	(202.2)
MISTY1	195.3	(193.8)	200.0	(197.8)
Triple DES	48.7	(48.6)	48.7	(48.6)

表 3.7: サーバ環境 [UltraSPARC IIi (400MHz)]

64 ビットブロック暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
CIPHERUNICORN-E	17.5	(17.4)	17.5	(17.4)
Hierocrypt-L1	67.7	(67.4)	51.2	(50.8)
	77.1	(76.2)	84.2	(83.2)

表 3.8: ハイエンド環境 [Alpha 21264 (463MHz)]

64 ビットブロック暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
CIPHERUNICORN-E	18.8	(18.7)	18.9	(18.8)
Hierocrypt-L1	141.1	(138.7)	141.1	(139.8)
	165.5	(162.8)	165.5	(162.8)
MISTY1	139.1	(138.0)	143.8	(142.5)

1. PC 環境

この結果から、PC 環境においては、Triple DES を比較対象とすると、CIPHERUNICORN-E が遅いグループに分類され、残りは十分速いグループに属すると言える。暗号化と復号で若干の速度差が見られる暗号もあるが、実装において問題となるほどのものでは無いと判断できる。また、平均値と最速値が著しく乖離している暗号も見られないので、応募暗号は PC 環境において安定して動作することが期待できる。

2. サーバ環境

この結果から分かることは、CPU スペックの向上がそのまま直に暗号の処理速度向上に結びつかない場合があることである。Hierocrypt-L1 は、応募者が測定プログラムを改変した場合の値が欄の下段に記載されている。メモリ確保を効率化することにより 1 割程度の速度向上が見ら

れる。

暗号化と復号で若干の速度差が見られる暗号もあるが、実装において問題となるほどのものでは無いと判断できる。また、平均値と最速値が著しく乖離している暗号も見られないので、サーバ環境において安定した動作が期待できる。

なお、サーバ環境は応募者の選択環境である。表に記載されていない暗号も実際にはこの環境で実装することは可能であるが、設計思想に合わない場合もあるので、応募者の意向を尊重し選択環境とした。

3. ハイエンド環境

Alpha 21264 は 64 ビット CPU で巨大な 1 次キャッシュを持つ。今後このような構造へ汎用 CPU が進化するならば、応募暗号間において、この結果から分かるような傾向があると見積もられる。

なお、ハイエンド環境も応募者の選択環境である。表に記載されていない暗号も実際にはこの環境で実装することは可能であるが、設計思想に合わない場合もあるので応募者の意向を尊重し選択環境とした。

■鍵スケジュール部 + データランダム化部処理時間 鍵セットアップから 1 ブロックデータ (64 ビット) の暗号化 (復号) が終了するまでの処理時間を測定した。測定値は [clocks] だが、分かりやすいように、処理時間 [μsec] に変換した。この値が小さいほど高速である。測定値は実行環境にかなりの影響を受けるので、この値が必ず実現されるとは限らない。また、測定プログラムの誤差や前述の変換等による誤差が生じている。さらに、測定プログラムの主旨を違えない程度の改変 (後述) を加えるのみで、測定値が大幅に変わる場合もある。したがって、この表の値のみで断定するのは危険である。各測定値欄に下段にも値が記載されているものは、応募者による測定プログラムの改変した場合の測定値である。測定プログラムは、すべての応募暗号に同一の条件を与えるため、メモリ領域を多めに確保している。ここでいう改変とは、多めのメモリ領域を各暗号ごとに最適化した場合のことである。この改変は、

- 実際の実装状況により近いこと
- メモリ領域の大きさが速度に与える影響の原因が不明なこと

を考慮し、今回の報告書では両方の値を記載することにした。

この値は認証にブロック暗号を用いる場合などの参考になる。したがって、数 μsec で処理が終了することが望ましい。

以上の結果から、評価対象となった暗号技術は、このような実装環境において充分実用に耐える動作が期待できることが分かる。

ソフトウェア実装の性能は、応募者の開発により日々向上している。本報告書に記載されている値よりも、速い実装が実現されていることが予想される。最新の状況については、応募者に問い合わせるのが望ましい。

表 3.9: PC 環境 [Pentium III (650MHz)]

64 ビットブロック暗号	処理時間 [μsec]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
CIPHERUNICORN-E	3.72	(3.73)	3.70	(3.72)
Hierocrypt-L1	0.58	(0.58)	0.95	(0.95)
MISTY1	0.55	(0.55)	0.54	(0.54)
Triple DES	3.02	(3.03)	3.03	(3.04)

表 3.10: サーバ環境 [UltraSPARC IIi (400MHz)]

64 ビットブロック暗号	処理時間 [μsec]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
CIPHERUNICORN-E	7.21	(7.23)	7.34	(7.36)
Hierocrypt-L1	1.80	(1.80)	3.01	(3.04)
	1.54	(1.55)	2.53	(2.58)

表 3.11: ハイエンド環境 [Alpha 21264 (463MHz)]

64 ビットブロック暗号	処理時間 [μsec]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
CIPHERUNICORN-E	5.14	(5.16)	5.66	(5.69)
Hierocrypt-L1	0.84	(0.85)	1.35	(1.41)
	0.83	(0.84)	1.33	(1.40)
MISTY1	0.72	(0.73)	0.68	(0.73)

表 3.12: 64 ビットブロック暗号の安全性余裕と処理速度比 [Pentium III]

	安全性余裕= 段数/解読可能段数	処理速度比 (データランダム化部)	処理速度比 (鍵スケジュール部込み)
UNI-E	16 / —*	0.60	0.82
HC-L1	6 / 3.5	4.25	3.97
MISTY1	8 / 5	4.07	5.57
Triple DES	48 / 48	1.00	1.00

*CIPHERUNICORN-E は、学術的な解読段数がまだ知られていない。

3.2.1.3　安全性余裕と速度

同じ暗号であれば、繰り返し段数を増加させることにより、定性的には安全性が増加し、暗号化の速度は低下する。ここでは、解読計算量が鍵の全数探索未満かつ解読に必要な平文が全平文数未満で解読できることを学術的な解読と呼ぶ。各暗号に対し、学術的な解読が知られている解読可能段数と実際の仕様段数の比を安全性余裕とし、今回の速度測定値を Triple DES に対する相対速度として示したものが表 3.12 である。なお、速度は暗号化と復号の最速値を平均したものである。

3.2.2　128 ビットブロック暗号

対象は、AES (Rijndael)、Camellia、CIPHERUNICORN-A、Hierocrypt-3、RC6 Block Cipher[*1]および SC2000 の 6 種類である。Camellia から SC2000 までの応募があり、AES はその他評価が必要な暗号として 2001 年度から評価対象として追加した。評価概要を表を含め以下に示す。記述内容は以下のとおり。

■特徴　提案組織、暗号発表年、構造上の特徴、データランダム化部で使用する演算等の特徴を載せた。なお、段数等可変パラメータを持つものについては、本公募への提案者の推奨値を記した。

■安全性　3 つの観点 (差分攻撃・線形攻撃に対する耐性、代数的およびその他攻撃に対する耐性、アバランシュ性) から記述する。

- 差分攻撃・線形攻撃に対する耐性では、汎用的な確率的攻撃方法である差分攻撃・線形攻撃に対する強度評価指標として、最大差分確率・最大線形確率、または最大差分特性確率・最大線形特性確率を示す。
- 代数的およびその他攻撃に対する耐性では、高階差分攻撃や補間攻撃、Square 攻撃等の代数的手法に対する耐性や鍵関連攻撃、mod n 攻撃等、その他の攻撃に対する耐性を述べる。高階差分攻撃や補間攻撃の評価は、暗号系の基本的弱点を代数的観点から探る手法であり、段数が大きい場合、通常これらの手法の攻撃で問題になることは少ない。ただし、そこで得られた弱点は、他の攻撃手法と組み合わせが可能な場合、最終的な暗号強度に影響を与える可能性がある。
- アバランシュ性は、暗号系におけるデータ撹拌の様子を統計的に捉えるものであり、通常は、直接、解読に結びつくことは無いが、暗号の部分関数の弱点を探る際の糸口を与える。

[*1] CRYPTREC 事務局では、2002 年 10 月 16 日付文書で RSA セキュリティ株式会社より、「知的財産権上の問題により、今後 RC6 の普及活動は行わない」との連絡を受領している。

■ソフトウェア実装評価 IC カード環境での評価以外は、2000 年度に評価が行われた。暗号は安全面だけでなく、使用状況を想定し実装面も考慮する必要がある。電子政府における暗号実装に対する要求事項は現在のところ不明であるが、ソフトウェア実装の評価においては、評価時点で一般的と思われる PC 環境 (必須)、現時点で最も普及していると思われるサーバ環境 (選択)、高性能を実現しているハイエンド環境 (選択) の 3 つの環境を想定した。データランダム化部処理速度と鍵スケジュール部 + データランダム化部処理時間の 2 種類の測定を実施した。IC カード環境での評価では、一部のアルゴリズムについて、鍵スケジュール部 + データランダム化部処理時間の測定を実施した。

■ハードウェア実装評価 2002 年度のハードウェア実装評価では、「応募書類 (アルゴリズム仕様書およびテストベクタ)」だけの情報から第三者によるハードウェア実装が可能であることを確認する目的で、FPGA 環境において各アルゴリズムの実装検証を行った。

今回の実装は動作検証が主目的であるため、各アルゴリズムの性質を考慮した特別な回路規模縮小や動作速度向上などの工夫は行なわず、ストレート・フォワードなアーキテクチャをとっている。そのため、必ずしも最適な実装が行なわれているとは言えず、今回の結果に基づいて各アルゴリズムの回路実装効率を公平に比較評価することはできない。従って、Triple DES との相対比較はもとより、回路規模と動作速度の数値も公表しないこととした。ただし、いずれの暗号アルゴリズムも FPGA 実装環境において 33MHz 動作を確認している。FPGA 実装環境の概要については、「3.1.3 ハードウェア実装評価方法」を参照されたい。

■総合評価 安全性および実装評価を総合した観点から、評価結果を表 3.13、表 3.14 に示す。なお、UNI-A は CIPHERUNICORN-A のことである。

3.2.2.1 安全性評価結果の総評

■差分攻撃・線形攻撃に対する耐性 差分攻撃・線形攻撃に対する耐性は最大差分確率・最大線形確率で与えられる。この確率が安全性保証の観点で評価されているのは AES と Hierocrypt-3 である。それぞれ 4 段、2 段で 2^{-96} 以下が保証されている。しかし、128 ビットブロック暗号としての安全性を十分保証するほどの小さな確率で保証されている暗号は評価対象には無い。

最大差分確率・最大線形確率の真値を求めることは困難であり、それに準じた指標として最大差分特性確率・最大線形特性確率がある。最大特性確率の評価は、

- 構成部品の最大差分確率・最大線形確率をもとに特性確率の上界を出す方法
- 計算機探索により最大特性確率を求める方法

がある。

特性確率の上界が活性 S-box 数評価を使って示されているのが Camellia、Hierocrypt-3、AES である。Camellia は、FL/FL^{-1} 関数を除いて、12 段で差分特性確率・線形特性確率が 2^{-132} を超

表 3.13: 128 ビットブロック暗号の評価結果総評 (1/2)

AES (Rijndael)	特徴
	NIST (2000) SPN 構造、10 段 (128 ビット鍵)、12 段 (192 ビット鍵)、14 段 (256 ビット鍵)。S-box は 8×8 の 1 種類。GF(2^8) 上の逆数演算をベースに設計し、差分攻撃・線形攻撃に耐性。拡散層 P は、バイト単位の転置 (ShiftRow)、バイト処理による 4 バイト内拡散 (MixColumn) で構成。テーブル参照、EXOR、AND。Square 暗号の後継。P 層設計では、活性 S-box 数の考えで評価。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
Camellia	特徴
	NTT、三菱電機 (2000) Feistel 型構造、18 段 (128 ビット鍵)、24 段 (192 ビット鍵、256 ビット鍵)、6 段ごとに FL/FL^{-1} 関数。初期、最終処理として拡大鍵 EXOR。ラウンド関数は 8 個の S-box とバイト単位演算の P 層。S-box は 8×8 の 1 種類。GF(2^8) 上の逆数演算をベースに設計し、差分攻撃・線形攻撃に耐性。テーブル参照、EXOR、AND、OR、巡回シフト。P 層設計では、活性 S-box 数の考えで評価。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
UNI-A	特徴
	NEC (2000) Feistel 型構造、16 段。ラウンド関数は複雑。安全性を高める意図で本流部と一時鍵生成部で構成。ラウンド関数は S-box を基本部品とし、T、A 関数で構成。S-box は 8×8 の 4 種類。GF(2^8) 上のべき乗演算をベースに設計し、差分攻撃・線形攻撃に耐性。テーブル参照、加算、乗算、EXOR、AND、巡回シフト演算。暗号評価支援システムで、有意な相関が見られないようにラウンド関数構造を設計。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は遅いグループである。
Hierocrypt-3	特徴
	東芝 (2000) 入れ子型 SPN 構造、6 段 (128 ビット鍵)、7 段 (192 ビット鍵)、8 段 (256 ビット鍵)。各段は XS 関数の 4 並列および P 層で構成。XS 関数は、P 層を 4 並列の S-box 2 層で挟んだ構造。S-box は 8×8 の 1 種類。GF(2^8) 上のべき乗演算をベースに設計し、差分攻撃・線形攻撃に耐性。テーブル参照、EXOR、AND。Hierocrypt-L1 と相似な構造。P 層設計では、活性 S-box 数の考えで評価。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。

表 3.14: 128 ビットブロック暗号の評価結果総評 (2/2)

RC6	特徴
	RSA セキュリティ (1998) 32 ビット 4 ブロックからなる変形 Feistel 型構造、20 段。ラウンド関数は、簡潔な構造。32 ビット入力、32+5 ビット出力。2 ブロックに EXOR および、データ依存巡回シフトで影響。ラウンド関数は、乗算、加算、巡回シフトで構成。演算は、いずれも 32 ビットワード長に対する処理で、32 ビット CPU を意識した構成。ワード長、段数、鍵長の選択可能な可変パラメータ構造。RC5 の設計思想を継承。
	総合評価
	安全性について、今のところ問題は見つかっていない。Pentium III 上の暗号化で最速であるが、ソフトウェア処理速度はプラットフォームに大きく依存。なお、CRYPTREC 事務局では、2002 年 10 月 16 日付け文書で、RSA セキュリティ株式会社から、「知的財産権上の問題により、今後 RC6 の普及活動は行わない」との連絡を受領している。
SC2000	特徴
	富士通 (2000) Feistel 型構造と SPN 構造の重ね合わせ。データランダム化部の段数は、19 段 (128 ビット鍵)、22 段 (192 ビット鍵、256 ビット鍵)。SPN 構造部で 4×4 の S-box を、Feistel 型構造部で 5×5 および 6×6 の 2 種類の S-box を使用。拡大体上のべき乗関数をベースに設計し差分攻撃・線形攻撃に耐性。代数的攻撃に耐性。テーブル参照、EXOR、AND。SPN 構造部は、高速実装法の Bitslice 法適用可。P 層設計では、活性 S-box 数の考えで評価。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。

えず、Hierocrypt-3 は 2 段、AES は 4 段で差分特性確率・線形特性確率が 2^{-150} を超えないことが示されている。

CIPHERUNICORN-A は、そのラウンド関数が複雑な構造であるため、解析が難しい。自己評価書では、簡略化したラウンド関数 mF に対する truncated vector 探索を交えて、15 段差分特性確率 2^{-140}、同線形特性確率 $2^{-140.14}$ の上界が示されている。2001 年度は、この簡略化の正当性を検証し、適切な見積もり方を検討した。その結果、簡略化したラウンド関数 mF による評価では、CIPHERUNICORN-A の安全性を示す根拠として乏しいことが明らかとなった。また、安全性を脅かすものではないが、弱鍵の存在が明らかとなった。さらに、2002 年度も引き続き、簡略化したラウンド関数での詳細な評価が進められた結果、仕様段数である 16 段の CIPHERUNICORN-A は差分攻撃・線形攻撃に対して十分な安全性を持つと考えられる、という結論が得られた。

RC6 は構造が簡明であるが、32 ビットワード長の処理が基本であり厳密な評価が難しい。しかし、その前身の RC5 に対する評価研究および AES 応募に係わる研究により、14 段最大差分特性確率 2^{-140}、18 段最大線形特性確率 2^{-155} とされている。

SC2000 は truncated vector 探索により、15 段最大差分特性確率が 2^{-134} を、同最大線形特性確率が 2^{-142} を超えないことが示されている。さらに、同じ構造を持つ差分特性で、11 段差分特性

確率が 2^{-117} となるものが提案者らにより発見されており、前述の解析結果の信頼性を補強している。

128 ビットブロック暗号のこれら特性確率が 2^{-128} 以下になることを安全性の証とする手法を、差分攻撃・線形攻撃に対する実際的安全性保証という。以上のように、いずれの 128 ビットブロック暗号も、現在その値を下回っており、学術的な差分攻撃・線形攻撃に対する耐性が保証されている。

■代数的およびその他の攻撃に対する耐性　高階差分攻撃や補間攻撃に対する耐性に関して、64 ビットブロック暗号と同様に評価した。

結果として、いずれのアルゴリズムも提案段数においては、鍵の全数探索より効率のよい解読方法は発見されていない。高階差分攻撃を適用することで、差分攻撃・線形攻撃に比べ、より高段数まで攻撃が可能となる暗号は、Hierocrypt-3 と AES である。Hierocrypt-3 に対しては、32 階の高階差分攻撃 (32 階の Square 攻撃) を基本にする攻撃法で、128 ビット鍵に対しては 6 段中 3 段まで、192 ビット (または 256 ビット) 鍵に対しては 8 段 (または 10 段) 中 3.5 段まで攻撃可能である。AES についても、Square 攻撃 (32 階の高階差分攻撃) を適用し、部分総和法を用いることで、それぞれ 128 ビット鍵に対しては 10 段中 7 段まで、192 ビット鍵に対しては 12 段中 8 段まで、256 ビット鍵に対しては 14 段中 8 段までが鍵の全数探索より効率よく解読可能である。AES に対するこれらの攻撃方法は、128 ビットブロック暗号で生成可能な平文組数 2^{128} とほぼ同等である $2^{128} - 2^{119}$ 個の平文組を必要とする。256 ビット鍵については、関連鍵攻撃を用いることで、さらに 14 段中 9 段までが鍵の全数探索より効率よく解読できる。

Camellia に関しては、高階差分攻撃により 128 ビット鍵に対して 18 段中 8 段まで、256 ビット鍵に対して 24 段中 11 段まで攻撃が可能である。さらに、選択暗号文を用いた高階差分攻撃により 256 ビット鍵に対して 24 段中 11 段まで攻撃できるという報告がある。

その他の攻撃の中では、RC6 に関してカイ 2 乗攻撃が効果を挙げている。この攻撃により、それぞれ 20 段中、128 ビット鍵に対しては 12 段まで、192 ビット鍵に対しては 14 段まで、256 ビット鍵に対しては 15 段まで鍵の全数探索より効率よく解読可能である。

その他、不能差分攻撃、ブーメラン攻撃、mod n 攻撃、非全単射攻撃等について、現在のところ、どのアルゴリズムも実用的観点から安全性に関する問題点は報告されていない。

■アバランシュ性評価　本評価は 2000 年度に実施された。「鍵スケジュール部を含む暗号化処理全体」では、すべてのアルゴリズムが期待値を満たした。しかし「鍵スケジュール部単体」では、Camellia、Hierocrypt-3、SC2000 で期待値を満たさない部分を検出した。一方、「ラウンド関数単体」では、Camellia、Hierocrypt-3、RC6、SC2000 で期待値を満たさない部分を検出した。

3.2.2.2　ソフトウェア実装評価の総評

本評価は、IC カード環境での評価を除き、2000 年度に行われた。記載されている数値は、2000 年度に測定された数値である。

表 3.15: 128 ビットブロック暗号のアバランシュ性評価結果

Camellia	データランダム化部では 4 段以降の攪拌に特徴は見られない。鍵スケジュール部では秘密鍵長によって異なる特徴が見られる。ラウンド関数では期待値から離れている部分がある。
UNI-A	データランダム化部では 3 段以降の攪拌に特徴は見られない。鍵スケジュール部では特徴は見られない。ラウンド関数では特徴は見られない。
Hierocrypt-3	データランダム化部では 2 段以降の攪拌に特徴は見られない。鍵スケジュール部では秘密鍵と拡大鍵間に大きな関係が存在する。ラウンド関数では期待値から離れている部分がある。
RC6	データランダム化部では 4 段以降の攪拌に特徴は見られない。鍵スケジュール部では特徴は見られない。ラウンド関数では期待値から離れている部分がある。
SC2000	データランダム化部では 4 段以降の攪拌に特徴は見られない。鍵スケジュール部では秘密鍵が 192 ビットおよび 256 ビットの時に特徴が見られる。ラウンド関数では期待値から離れている部分がある。

■データランダム化部処理速度　1MB の平文 (暗号文) に対して鍵を設定し、暗号化 (復号) を行ったときの 1 ブロック (128 ビット) あたりの処理時間を測定した。測定値は [clocks/block] だが、分かりやすいように、処理速度 [Mbps] に変換した。この値が大きいほど高速である。測定値は実行環境にかなりの影響を受けるので、この値が必ず実現されるとは限らない。また、測定プログラムの誤差や前述の変換等による誤差が生じている。さらに、測定プログラムの主旨を違えない程度の改変 (後述) を加えるのみで、測定値が変わる場合もある。したがって、この表の値のみで断定するのは危険である。各測定値欄に下段にも値が記載されているものは、応募者による測定プログラムの改変した場合の測定値である。測定プログラムはすべての応募暗号に同一の条件を与えるため、メモリ領域を多めに確保している。ここでいう改変とは、多めのメモリ領域を各暗号ごとに最適化した場合のことである。この改変は、

- 実際の実装状況により近いこと
- メモリ領域の大きさが速度に与える影響の原因が不明なこと

を考慮し、今回の報告書では両方の値を記載することにした。

1. PC 環境

比較のため、最終段に Triple DES の測定値を記載した。Triple DES は 64 ビットブロック暗号である。

この結果から、PC 環境においては、Triple DES を比較対象とすると、CIPHERUNICORN-A 以外は十分速いグループに属すると言える。暗号化と復号で若干の速度差が見られる暗号もある

表 3.16: PC 環境 [Pentium III (650MHz)]

128 ビットブロック暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
Camellia	255.2	(254.4)	255.2	(254.2)
CIPHERUNICORN-A	53.0	(52.9)	52.9	(52.7)
Hierocrypt-3	205.9	(204.9)	195.3	(194.4)
RC6	322.5	(320.4)	317.6	(313.6)
SC2000	214.4	(212.6)	203.9	(202.6)
Triple DES (64 ビット暗号)	48.7	(48.6)	48.7	(48.6)

表 3.17: サーバ環境 [UltraSPARC IIi (400MHz)]

128 ビットブロック暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
Camellia	144.2	(142.9)	144.2	(143.3)
CIPHERUNICORN-A	22.5	(22.4)	22.2	(22.0)
Hierocrypt-3	100.4	(92.3)	67.6	(62.1)
	108.7	(108.2)	83.7	(83.1)
RC6	25.0	(24.5)	25.3	(24.7)
SC2000	165.2	(163.4)	165.7	(164.1)
	186.2	(184.2)	181.6	(179.0)

表 3.18: ハイエンド環境 [Alpha 21264 (463MHz)]

128 ビットブロック暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
Camellia	210.2	(205.3)	210.2	(205.6)
CIPHERUNICORN-A	32.4	(32.2)	33.5	(33.3)
Hierocrypt-3	141.1	(139.9)	138.8	(137.9)
	148.5	(145.9)	153.5	(150.7)
SC2000	205.1	(200.0)	210.2	(203.9)
	226.2	(214.5)	215.5	(205.1)

が、実装において問題となるほどのものでは無いと判断できる。また、平均値と最速値が著しく乖離している暗号も見られないので、応募暗号は PC 環境において安定して動作することが期待できる。

2. サーバ環境

この結果から分かることは、CPU スペックの向上がそのまま直に暗号の処理速度向上に結びつかない場合があることである。例えば、PC 環境において最速の RC6 はサーバ環境ではむしろ遅いグループに属している。Hierocrypt-3、SC2000 は、応募者が測定プログラムを改変した場合の値が欄の下段に記載されている。メモリ確保を効率化することにより 1 割程度の速度向上が見られる。

Hierocrypt-3 は暗号化と復号で速度に乖離があるが、これは暗復非対称の構造のため、復号側処理の最適化が十分なされていないことが原因に挙げられる。他は、暗号化/復号、最速値/平均値に著しい乖離が見られず、安定した動作が期待できる。

なお、サーバ環境は応募者の選択環境である。表に記載されていない暗号も実際にはこの環境で実装することは可能であるが、設計思想に合わない場合もあるので、応募者の意向を尊重し選択環境とした。

3. ハイエンド環境

提案から開発期間までの期間が短い暗号もあり、この結果のみで結論を出すのは問題があるが、この結果と以上の結果から分かることは、暗号は実装環境に応じて得意不得意があることである。例えば、サーバ環境では SC2000 が最速であるが、ハイエンド環境では Camellia が最速である。

Alpha 21264 は 64 ビット CPU で巨大な 1 次キャッシュを持つ。今後このような構造へ汎用 CPU が進化するならば、応募暗号間において、この結果から分かるような傾向があると見積もられる。

なお、ハイエンド環境も応募者の選択環境である。表に記載されていない暗号も実際にはこの環境で実装することは可能であるが、設計思想に合わない場合もあるので、応募者の意向を尊重した。

■鍵スケジュール部 + データランダム化部処理時間 鍵セットアップから 1 ブロックデータ (128 ビット) の暗号化 (復号) が終了するまでの処理時間を測定した。測定値は [clocks] だが、分かりやすいように、処理時間 [μsec] に変換した。この値が小さいほど高速である。測定値は実行環境にかなりの影響を受けるので、この値が必ず実現されるとは限らない。また、測定プログラムの誤差や前述の変換等による誤差が生じている。さらに、測定プログラムの主旨を違えない程度の改変 (後述) を加えるのみで、測定値が大幅に変わる場合もある。したがって、この表の値のみで断定するのは危険である。各測定値欄に下段にも値が記載されているものは、応募者による測定プログラムの改変した場合の測定値である。測定プログラムは、すべての応募暗号に同一の条件を与えるため、メモリ領域を多めに確保している。ここでいう改変とは、多めのメモリ領域を各暗号ごとに最適化した場合のことである。この改変は、

- 実際の実装状況により近いこと
- メモリ領域の大きさが速度に与える影響の原因が不明なこと

を考慮し、今回の報告書では両方の値を記載することにした。

この値は認証にブロック暗号を用いる場合などの参考になる。したがって、数 μsec で処理が終了することが望ましい。

表 3.19: PC 環境 [Pentium III (650MHz)]

128 ビットブロック暗号	処理時間 [μsec]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
Camellia	0.72	(0.75)	0.73	(0.76)
CIPHERUNICORN-A	7.36	(7.42)	7.38	(7.42)
Hierocrypt-3	1.12	(1.12)	2.07	(2.09)
RC6	2.51	(2.53)	2.51	(2.52)
SC2000	1.23	(1.24)	1.26	(1.26)
Triple DES (64 ビット暗号)	3.02	(3.03)	3.03	(3.04)

表 3.20: サーバ環境 [UltraSPARC IIi (400MHz)]

128 ビットブロック暗号	処理時間 [μsec]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
Camellia	1.01	(1.02)	1.01	(1.02)
CIPHERUNICORN-A	19.92	(20.40)	22.01	(22.57)
Hierocrypt-3	2.06	(2.07)	6.68	(6.71)
	1.90	(2.06)	6.53	(6.57)
RC6	10.19	(10.28)	10.05	(10.14)
SC2000	1.56	(1.57)	1.55	(1.56)

表 3.21: ハイエンド環境 [Alpha 21264 (463MHz)]

128 ビットブロック暗号	処理時間 [μsec]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
Camellia	0.97	(0.98)	0.94	(0.95)
CIPHERUNICORN-A	9.96	(9.99)	10.95	(11.01)
Hierocrypt-3	1.46	(1.47)	2.44	(2.47)
	1.44	(1.45)	2.44	(2.47)
SC2000	1.24	(1.25)	1.27	(1.28)

ソフトウェア実装の性能は、応募者の開発により日々向上している。本報告書に記載されている値よりも、速い実装が実現されていることが予想される。最新の状況については、応募者に問い合わせるのが望ましい。

■IC カード環境での実装評価 ローエンド型 IC カードでは、処理時間もさることながら、実装コード量や実行時における使用 RAM サイズがボトルネックとなる場合も多い。そこで、128 ビット秘密鍵を用いて 1 ブロック (128 ビットデータ) を暗号化・復号するときの処理時間のほか、ROM や RAM の使用メモリを測定した。その際、実際の IC カード利用環境を想定し、処理速度の高速化よりも使用 RAM サイズが適度に小さくなる (64 bytes 程度) ことを優先した実装を心がけた。

処理時間の測定値は [clocks] だが、分かりやすいように、処理時間 [msec] に変換した。この値が小さいほど高速である。また、コードサイズについては、暗号化処理のみが実行可能な実装形態、復号処理のみが実行可能な実装形態、暗号化処理と復号処理の両方が実行可能な実装形態、の 3 種類に分けてそれぞれのサイズを測定している。

表 3.22: IC カード環境 [Z80 シミュレータ (5MHz)]

128 ビットブロック暗号	実装形態	コード [bytes]	使用 RAM [bytes]	スタック [bytes]	処理時間 [msec]
Camellia	暗号化	1,023	48	12	7.12
	復号	1,042	48	12	7.51
	暗復共用	1,268	—	—	—
Hierocrypt-3	暗号化	2,577	73	8	9.98
	復号	3,662	73	8	14.36
	暗復共用	4,746	—	—	—
SC2000	暗号化	2,192	64	6	18.77
	復号	2,192	64	6	18.86
	暗復共用	2,350	—	—	—
AES (参考*1)	暗号化	—	—	—	7.14
	復号	—	—	—	10.42
	暗復共用	1,221	63	—	—

*1: (出典) Sano, Ohkuma, Shimizu, Motoyama and Kawamura, ”Efficient Implementation of Hierocrypt”, 2nd NESSIE Workshop 予稿集.

3.2.2.3 安全性余裕と速度

同じ暗号であれば、繰り返し段数を増加させることにより、定性的には安全性が増加し、暗号化の速度は低下する。ここでは、解読計算量が鍵の全数探索未満かつ解読に必要な平文が全平文数未満で解読できることを学術的な解読と呼ぶ。128 ビットブロック暗号では、鍵長 128、192、

256 ビットの 3 とおりの仕様で提案されている。ここでは、256 ビット鍵仕様における学術的な解読可能段数と実際の仕様段数の比を安全性余裕とし、今回の速度測定値を Triple DES に対する相対速度として示したものが表 3.23 である。なお、速度は 128 ビット鍵仕様の暗号化と復号の最速値を平均したものである。基準となる Triple DES の 128 ビットデータ処理時間は、測定値より次式で換算した。128 ビットブロック暗号間の相対比較として数値を利用されたい。

データランダム化部処理時間 [128 ビット] = データランダム化部処理時間 [64 ビット] × 2
鍵スケジュール込み処理時間 [128 ビット] = 鍵スケジュール込み処理時間 [64 ビット]
+ データランダム化部処理時間 [64 ビット]

表 3.23: 128 ビットブロック暗号の安全性余裕と処理速度比 [Pentium III]

	安全性余裕= 段数/解読可能段数	攻撃法	処理速度比 (データランダム化部)	処理速度比 (鍵スケジュール部込み)
AES	14 / 8 14 / 9	SQUARE 攻撃 関連鍵攻撃	2.15	1.23 *1
Camellia (FL 有)	24 / 11	高階差分攻撃	5.24	6.00
UNI-A	16 / —	未 *2	1.02	0.59
Hierocrypt-3	8 / 3.5	SQUARE 攻撃	4.12	2.73
RC6	20 / 15	カイ 2 乗攻撃	6.57	1.73
SC2000	22 / 13	差分攻撃	4.29	3.49
(Triple DES)	48 / 48	中間一致攻撃	1	1

*1. 参考値 Pentium III 600MHz, C, 文献 Lawrence E. Bassham, “Efficiency Testing of ANSI C Implementations of Round 2 Candidate Algorithms for the Advanced Encryption Standard,” AES3 conference, 5.1 節, Table 6 (128Blocks)
*2. CIPHERUNICORN-A は、学術的な解読段数がまだ知られていない。

3.2.3　ストリーム暗号

対象は、MUGI、MULTI-S01 および RC4 の 3 種類である。MUGI と MULTI-S01 の応募があり、RC4 はその他評価が必要な暗号として 2001 年度から評価対象として追加した。安全性および実装評価を総合した観点から、評価結果を表 3.24 に示す。

3.2.3.1　安全性評価結果の総評

■MUGI について　MUGI について、主に以下の観点からの安全性評価を行った。

- 差分攻撃・線形攻撃に対する耐性
- Linear masking 解析法に対する耐性
- XL 攻撃に対する耐性

表 3.24: ストリーム暗号の評価結果総評

MUGI	特徴
	日立 (2001) 1998 年に Daemen と Clapp が提案した擬似乱数生成器 Panama と同様の構造を持ち、既存のブロック暗号の解析手法がより適用し易いような設計と安全性の評価を目指した、ストリーム暗号向け擬似乱数生成器である。秘密鍵 128 ビット、初期ベクトル (公開値) 128 ビットをパラメータに持つ。設計思想の 1 つに既存の評価された暗号技術を再利用することを掲げており、AES の構成要素 (例えば S-box) が採用されている。
	総合評価
	安全性について、いまのところ問題は見つかっていない。ソフトウェアによる処理速度は速いグループである。
MULTI-S01	特徴
	日立 (2000) 擬似乱数生成とその利用モードを実現する暗号化処理、復号処理からなる。擬似乱数生成器は秘密鍵 (256 ビット) から鍵ストリームを生成する。この鍵ストリームを用いてメッセージを暗号化する。メッセージ秘匿だけでなく、メッセージ認証を同時に達成する点が特徴である。MULTI-S01 は擬似乱数生成器 Panama と暗号利用モード部で構成される。
	総合評価
	安全性について、いまのところ問題は見つかっていない。ソフトウェアによる処理速度は速いグループである。
RC4	特徴
	RSA セキュリティ (1987) RC4 アルゴリズムは非公開アルゴリズムであるため、RSA セキュリティ社との交渉により、文献 [2](p.255) に記載のアルゴリズムを CRYPTREC としては RC4 とみなして安全性評価を実施した。 この RC4 の中核技術は、擬似乱数生成器であり、ワード長 n ビットと n から決定される状態数 2^n の状態表で規定され、状態表の内容を置換しながら擬似乱数を生成している。秘密鍵の役割は状態表における初期状態の決定である。
	総合評価
	標準仕様 (ワード長 n=8、状態数 256) の RC4 については、現在のところ、現実的な解読法は提案されていない。しかし、秘密鍵から生成される初期状態によっては、必ずしも安全ではないという報告がなされている。従って、RC4 の利用に関してその初期状態の決定には注意が必要である。また SSL3.0/TLS1.0 での利用に関しては、現在のところその安全性に関して欠陥は報告されていない。ただし、SSL/TLS の仕様上は 40 ビット秘密鍵でも利用可能であるが、CRYPTREC としては 40 ビット秘密鍵での利用は安全ではないと判断する。

- 統計的性質の検証

以上の評価結果からは、どの評価者も、2^{128} よりも少ない計算量で秘密鍵を導出する攻撃法の存在は現時点でないとしており、今のところ、MUGI の安全性に致命的欠陥は見つかっていない。

また、擬似乱数生成の設計方法が Panama とよく似ていることから、他のストリーム暗号 (MULTI-S01, Panama) との比較検討を行った。この結果、MUGI は、Panama 以上に既存のブロック暗号の解析手法がより適用し易い設計であり、この点は暗号設計評価上、重要な優位性と評価できる。

■MULTI-S01 について　MULTI-S01 について、主に以下の安全性評価を行った。

評価内容 1 [MULTI-S01 の安全性に関する検証]

MULTI-S01 の暗号化・復号処理部は、ブロック暗号の暗号利用モード (Modes of Operation) と似た構造である。NIST 主催の Modes of Operation Workshop で見られるようなモードに対する安全性評価手法を用い、MULTI-S01 の暗号化・復号処理部の安全性を検証した。

評価内容 2 [擬似乱数生成器 Panama の安全性に関する検証]

MULTI-S01 の安全性は Panama に負うところがあるが、Panama の安全性は十分検証されているとは言い難い。暗号用擬似乱数生成器に対して提案されている様々な攻撃法を適用し、Panama の安全性を検証した。

評価内容 3 [擬似乱数生成器 Panama の統計的性質の検証]

Panama の乱数性は十分検証されているとは言い難い。2001 年度評価では FIPS PUB 140-1 に記載されている暗号用乱数として最低限必要な評価を行った。一方、NIST SP 800-21 には、暗号用乱数に対する、更に詳細な評価方法が記載されている。この評価を Panama に対して適用し、統計的性質の検証を行った。

〈〈 評価結果 〉〉

以上の評価結果から、以下のことが明らかになった。

- MULTI-S01 の安全性は Panama に帰着できる。
- Panama の乱数検定からは特段の欠陥は見あたらない。
- Panama の安全性について致命的な問題点は見あたらない。

■RC4 について　RC4 について、主に以下の安全性評価を行った。

- 統計的な出力偏りを利用した攻撃法に対する耐性

- 乱数系列から初期状態を推定する攻撃に対する耐性
- SSL/TLS で利用した場合の安全性
- WEP での RC4 解析など、既知の安全性評価のまとめ

以上の評価結果から、標準仕様 (ワード長 $n = 8$、状態数 256) の RC4 については、現在のところ、現実的な解読法は提案されていないものの、秘密鍵から生成される初期状態によってはかならずしも安全ではない、との結論が得られた。従って、RC4 の利用に関してその初期状態の決定には注意が必要である。なお、128 ビット鍵での SSL3.0/TLS1.0 での利用に関しては、現在のところその安全性に関して欠陥は報告されていない。

3.2.3.2 ソフトウェア実装評価の総評

本評価は 2002 年度に行われた。なお、測定環境は PC 環境のみである。

■暗号化・復号処理速度 1MB の平文 (暗号文) に対して鍵を設定し暗号化 (復号) を行ったときの 128 ビットデータの暗号化 (復号) に換算した処理時間を測定値 [clocks/128bits] として採用した。ここでは、分かり易いように、処理速度 [Mbps] に変換した。この値が大きいほど高速である。測定値は実行環境にかなりの影響を受けるので、この値が必ず実現されるとは限らない。また、測定プログラムの誤差や前述の変換等による誤差が生じている。さらに、測定プログラムの主旨を違えない程度の改変 (後述) を加えるのみで、測定値が変わる場合もある。従って、ブロック暗号の場合と同様、この表の値のみで断定するのは危険である。各測定値欄に下段にも値が記載されているものは、応募者による測定プログラムの改変した場合の測定値である。測定プログラムはすべての応募暗号に同一の条件を与えるため、メモリ領域を多めに確保している。ここでいう改変とは、多めのメモリ領域を各暗号ごとに最適化した場合のことである。この改変は、

- 実際の実装状況により近いこと
- メモリ領域の大きさが速度に与える影響の原因が不明なこと

を考慮し、今回の報告書では両方の値を記載することにした。

比較のため、最終段に Triple DES の測定値を記載した。Triple DES は 64 ビットブロック暗号である。この結果から、Triple DES を比較対象とすると、MUGI、MULTI-S01 とも十分速いグループに属すると言える。暗号化と復号で若干の速度差が見られる暗号もあるが、実装において問題となるほどのものでは無いと判断できる。

■鍵セットアップ処理時間 暗号化 (復号) するデータを 128 ビットとしたときの鍵セットアップ処理時間を測定した。測定値は [clocks/key] だが、分かり易いように、処理時間 [μsec/key] に変換した。この値が小さいほど高速である。測定値は実行環境にかなりの影響を受けるので、この値が必ず実現されるとは限らない。また、測定プログラムの誤差や前述の変換等による誤差が生じている。さらに、測定プログラムの主旨を違えない程度の改変 (後述) を加えるのみで、測定値が大幅に変わる場合もある。従って、この表の値のみで断定するのは危険である。各測定値欄に下段にも値が記載されているものは、応募者による測定プログラムの改変した場合の測定値で

表 3.25: PC 環境 [Pentium III (650MHz)]

ストリーム暗号	処理速度 [Mbps]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
MUGI	523.7	(420.4)	518.7	(410.3)
	524.8	(516.5)	522.4	(515.1)
MULTI-S01	347.5	(283.5)	366.3	(294.9)
	349.8	(346.5)	368.8	(364.5)
Triple DES (64 ビット暗号)	48.7	(48.6)	48.7	(48.6)

ある。測定プログラムは、すべての応募暗号に同一の条件を与えるため、メモリ領域を多めに確保している。ここでいう改変とは、多めのメモリ領域を各暗号ごとに最適化した場合のことである。この改変は、

- 実際の実装状況により近いこと
- メモリ領域の大きさが速度に与える影響の原因が不明なこと

を考慮し、今回の報告書では両方の値を記載することにした。

表 3.26: PC 環境 [Pentium III (650MHz)]

ストリーム暗号	処理時間 [μsec/key]			
	暗号化		復号	
	最速値	(平均値)	最速値	(平均値)
MUGI	35.96	(78.08)	31.33	(43.20)
	29.31	(66.64)	37.82	(50.20)
MULTI-S01	8.69	(32.10)	8.76	(15.19)
	8.51	(32.51)	8.36	(13.52)
Triple DES (64 ビット暗号)	3.02	(3.03)	3.03	(3.04)

ソフトウェア実装の性能は、応募者の開発により日々向上している。本報告書に記載されている値よりも、速い実装が実現されていることが予想される。最新の状況については、応募者に問い合わせるのが望ましい。

■ハードウェア実装評価　2002 年度のハードウェア実装評価では、「応募書類 (アルゴリズム仕様書およびテストベクタ)」だけの情報から第三者によるハードウェア実装が可能であることを確認する目的で、FPGA 環境において各アルゴリズムの暗号化機能の実装検証を行った。

今回の実装は動作検証が主目的であるため、各アルゴリズムの性質を考慮した特別な回路規模縮小や動作速度向上などの工夫は行なわず、ストレート・フォワードなアーキテクチャをとって

いる。そのため、必ずしも最適な実装が行なわれているとは言えず、今回の結果に基づいて各アルゴリズムの回路実装効率を公平に比較評価することはできない。従って、Triple DES との相対比較はもとより、回路規模と動作速度の数値も公表しないこととした。ただし、いずれの暗号アルゴリズムも FPGA 実装環境において 33MHz 動作を確認している。

Data Randamize Clock 数の評価結果を次表に示す。

表 3.27: ストリーム暗号の HW 実装評価結果

ストリーム暗号	Data Randamize Clock 数	Total Clock 数
MUGI	6	104
MULTI-S01	70	140

FPGA 実装環境の概要については、「3.1.3 ハードウェア実装評価方法」を参照されたい。

3.3　個別暗号技術の評価

3.3.1　CIPHERUNICORN-E

3.3.1.1　技術概要

CIPHERUNICORN-E は、1998 年に日本電気株式会社 (NEC) が開発したブロック長 64 ビット、鍵長 128 ビットの 64 ビットブロック暗号であり、NEC より提案された。暗号の基本構造は 16 段の Feistel 型暗号である [1]。

この暗号の特徴は、暗号の基本となるラウンド関数での拡大鍵探索を難しくすることで安全性を高めることを意図して、本流部と一時鍵生成部とで構成される極めて複雑なラウンド関数を利用している点である。また、多くの暗号の設計方針とは異なり、ラウンド関数をブラックボックスとみなして、設計者が定めた初等統計量評価を行う暗号強度評価支援システム [2] により有意な相関関係が見出せないラウンド関数を設計することを主要な設計方針としている。その結果、ラウンド関数における初等統計量評価では、すべての項目について、データ攪拌の偏りは検出されなかったとしている。実装面では、ソフトウェア、ハードウェアとも実装可能であり、特に 32 ビットプロセッサで高速に処理できるように設計したと述べている。

3.3.1.2　技術仕様

ブロック長 64 ビット、鍵長 128 ビット、16 段 Feistel 型構造を採用した 64 ビットブロック暗号であり、2 段ごとに L 関数が挿入される。鍵スケジュール部は、秘密鍵を攪拌しながら、2,624 ビットの拡大鍵を生成する。

■データランダム化部　ラウンド関数は、拡大鍵 (関数鍵 (function key) とシード鍵 (seed key)) 32 ビット ×4 (合計 128 ビット) を用いた 32 ビット入出力関数であり、S-box、32 ビット算術加算、シフト演算により構成される。なお、この関数は全単射関数ではない。関数内部では、32 ビットの入力データは、本流部 (main stream) と一時鍵生成部 (temporary key generation) に分岐し、関数鍵は本流部に、シード鍵は一時鍵生成部にそれぞれ入力される。さらに、一時鍵生成部で入力データとシード鍵から生成された一時鍵が本流部に挿入され、最終的に 32 ビットの出力データが得られる。また、本流部の構成の一部は、一時鍵の値によって変化するデータ依存関数となっている。補助関数である L 関数は、拡大鍵 64 ビット ×2 (合計 128 ビット) を用いた、64 ビット入出力関数である。ビット単位の論理積として構成された鍵依存線形変換関数となっている。

■鍵スケジュール部　鍵スケジュール部は、ST 関数をラウンド関数とする Feistel 型構造をしており、秘密鍵を攪拌しながら、各 ST 関数から 2 または 4 個の 32 ビットの拡大鍵を出力する。ST 関数は、ラウンド関数と同じ T 関数を利用する。

■設計方針 差分攻撃や線形攻撃はラウンド関数での攪拌偏りを利用して鍵情報を推定することから、ラウンド関数で攪拌偏りが検出できない構造にするとの設計方針のもと、ラウンド関数をブラックボックスとみなして評価を行う暗号強度評価支援システムにより、以下の条件を満たすようにラウンド関数の設計を行っている。

- 高い確率で成立する入力ビットと出力ビットの関係が存在しない
- 高い確率で成立する出力ビット間の関係が存在しない
- 高い確率で成立する入力ビットの変化と出力ビットの変化の関係が存在しない
- 高い確率で成立する鍵ビットの変化と出力ビットの変化の関係が存在しない
- 高い確率で 0 あるいは 1 となる出力ビットが存在しない

3.3.1.3 その他

暗号強度評価支援システムによって同じように設計された暗号として、128 ビットブロック暗号である CIPHERUNICORN-A がある。

標準化関連として、ISO/IEC 9979 アルゴリズム公開登録が行われている。

3.3.1.4 安全性評価結果

■総評 CIPHERUNICORN-E のラウンド関数の構成は非常に複雑であり、差分攻撃や線形攻撃を始めとする、理論的な解読技術に対する安全性を正確に評価・解析することは困難である。このため、CIPHERUNICORN-E は 2000 年度の CRYPTREC Report 2000 において継続的な評価が必要であるとの総合評価を受けた。そこで、CRYPTREC Report 2001 において、以下の観点から安全性評価を継続的に実施した。

- 差分特性確率の観点からみた差分攻撃に対する安全性
- 線形特性確率の観点からみた線形攻撃に対する安全性
- その他の攻撃法に対する安全性

CRYPTREC Report 2000 では、概ね適切な考慮に基づいてラウンド関数の構成を簡略したモデル (mF 関数) で、12 段以上で最大差分確率の上界が 2^{-64} を下回ることが示されている。また線形特性確率においては、やはりラウンド関数を簡略したモデル (mF^* 関数) で、8 段でその上界が $2^{-70.72}$ となり、2^{-64} を下回ることが示されている。さらに CRYPTREC Report 2001 では 4 名 (チーム) による評価者が差分特性確率および線形特性確率について、各々適切と考慮する手法に基づいてラウンド関数および暗号全体の評価を実施した結果、いずれも仕様段数である 16 段よりも小さい段数で各々の上界値が 2^{-64} を十分下回るという結果が示されている。これらの評価結果はいずれも CIPHERUNICORN-E のラウンド関数に何らかの近似を施したラウンド関数に基づいて算出されたものである。しかし、多数の評価者が異なる手法による近似を利用しながらほぼ同じ安全性評価結果を得られたことから、CIPHERUNICORN-E の差分攻撃や線形攻撃に対

する安全性は、少なくとも今回見積もられた評価結果と同程度以上であると期待される。

また、上記以外の攻撃法については、CRYPTREC Report 2000 で示すように現時点までに問題となるような点は特に発見されていない。

以上の結論を総合すると、現在までに CIPHERUNICORN-E の安全性について問題点は見つかっていない。

■初等統計量評価　5 段以上で暗号出力と乱数との識別が不可能となることを確認した。さらに、ラウンド関数に対する初等統計量評価のすべての項目について良好な結果を得ているなど、初等的な乱数性に関しては優れていると判断される。なお、データ攪拌偏りが検出できないようにラウンド関数を設計したとしているが、このように設計されたラウンド関数が、ランダム関数とほぼ同じ特性をもつことを意味しているわけではないことに注意せよ。

■理論的解読法ごとの安全性評価

差分攻撃に対する安全性:　ラウンド関数の構成が複雑であり、直接的に評価することが困難な場合、適切な仮定を置くことによってラウンド関数を簡略化した暗号モデルを考え、そのモデル上での安全性を議論することがある。これは、実際の暗号が、適切な仮定をもとにした簡略化モデルでの安全性と同程度以上の安全性を有していると一般に期待されるためである。

CRYPTREC Report 2000 では、(1) 算術加算を排他的論理和に置換、(2)Y 関数は 32 ビットデータの上位 1 バイトへ入力ビットを集約する処理に置換など、おおむね適切な考慮に基づいてラウンド関数の構成を簡略化した mF 関数を利用したモデルで安全性の評価を行った。その結果、少なくとも 12 段以上で最大差分特性確率の上界が 2^{-64} を下回ることが示されている。

CRYPTREC Report 2001 では 4 名 (チーム) の評価者により以下の評価結果が示されている。

評価者 1:　ラウンド関数の最大差分特性確率の上界が 2^{-21} であり、13 段での最大差分特性確率の上界が 2^{-126} であると指摘した。これにより仕様段数である 16 段の CIPHERUNICORN-E は差分攻撃では解読不可能であると示している。

評価者 2:　ラウンド関数の最大差分特性確率の上界は自己評価書の結果と同じく 2^{-12} であるが、アルゴリズム全体としての上界は 2^{-72} であると指摘した。仕様段数である 16 段の CIPHERUNICORN-E は差分攻撃では解読不可能であるという結論は同じである。

評価者 3:　ラウンド関数の最大差分特性確率の上界が 2^{-14} であると指摘した。これにより 15 段での最大差分特性確率の上界が 2^{-98} であると示している。その結果として、仕様段数である 16 段の CIPHERUNICORN-E は差分攻撃では解読不可能であるという結論を示している。

評価者 4:　ラウンド関数の最大差分特性確率の上界が 2^{-16} であると示し、10 段以上あれば 2^{-64} を下回ることを示している。仕様段数である 16 段の CIPHERUNICORN-E が差分攻撃では解読不可能であるという結論は同じである。

以上の評価結果を総合的に判断すると、様々な異なる近似モデルでの安全性評価結果のいず

れについても仕様段数である 16 段よりも小さい段数で上界値が 2^{-64} を十分下回ることから、CIPHERUNICORN-E は差分攻撃に対して安全であるとことが期待される。

線形攻撃に対する安全性: CRYPTREC Report 2000 では、ラウンド関数を簡略したモデル (mF* 関数) で、ラウンド関数の最大線形特性確率の上界値が $2^{-17.68}$ となり、8 段でその上界が $2^{-70.72}$ となり、2^{-64} を下回ることが示されている。また自己評価書では、ラウンド関数を簡略したモデル (mF 関数) で、ラウンド関数の最大線形特性確率の上界値は $2^{-63.90}$ と示されている。

CRYPTREC Report 2001 年では 4 名 (チーム) の評価者により以下の評価結果が示されている。

評価者 1: ラウンド関数の最大線形特性確率の上界が $2^{-24.64}$ であり、13 段での最大線形特性確率の上界が $2^{-147.84}$ であると指摘した。これにより仕様段数である 16 段の CIPHERUNICORN-E は線形攻撃では解読不可能であると示している。

評価者 2: ラウンド関数の最大線形特性確率の上界が mF 関数を用いて 2^{-62} となると指摘した。仕様段数である 16 段の CIPHERUNICORN-E は線形攻撃では解読不可能であるという結論は同じである。

評価者 3: ラウンド関数の最大線形特性確率の上界が mF 関数を用いて $2^{-27.3}$ であると指摘した。これにより 15 段での最大線形特性確率の上界が $2^{-191.2}$ であると示している。その結果として、仕様段数である 16 段の CIPHERUNICORN-E は線形攻撃では解読不可能であるという結論を示している。

評価者 4: ラウンド関数の最大線形特性確率の上界が 2^{-16} であると示し、10 段以上あれば 2^{-64} を下回ることを示している。仕様段数である 16 段の CIPHERUNICORN-E が線形攻撃では解読不可能であるという結論は同じである。

以上の評価結果を総合的に判断すると、様々な異なる近似モデルでの安全性評価結果のいずれについても仕様段数である 16 段よりも小さい段数で上界値が 2^{-64} を十分下回ることから、CIPHERUNICORN-E は線形攻撃に対して安全であるとことが期待される。

高階差分攻撃、補間攻撃に対する安全性: これらの攻撃法に対する安全性は、自己評価書でもおおむね適切な考慮に基づく評価がなされており、また詳細評価においても、特に問題となるような点は発見されなかった。

鍵衝突攻撃に対する安全性: 鍵スケジュール部の構成上、鍵衝突は起こらないと考えられる。

■弱鍵の存在 鍵の値によっては、L 関数があることによって、Feistel 型暗号で重要な左右データの入れ替えが行われず、実効段数が減少することがある。したがって、利用する秘密鍵に、そのような弱鍵が発生しないことを確認した上で利用することが望ましい。

3.3.1.5　ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.28、表 3.29 のとおりである。

表 3.28: CIPHERUNICORN-E のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	26,232 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	“/O2 /Oy-” (実行速度) を指定	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,435 / 1,438	1,424 / 1,426
2 回目	1,434 / 1,444	1,422 / 1,425
3 回目	1,436 / 1,440	1,422 / 1,425

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	11,848 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	“-v -fast”を指定	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,462 / 1,469	1,462 / 1,468
2 回目	1,462 / 1,468	1,462 / 1,468
3 回目	1,462 / 1,469	1,462 / 1,468

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	13,552 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	“-O4” を指定	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,575 / 1,583	1,566 / 1,579
2 回目	1,575 / 1,583	1,568 / 1,582
3 回目	1,575 / 1,583	1,568 / 1,580

表 3.29: CIPHERUNICORN-E の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	13,552 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/O4 を指定	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	2,421 / 2,426	2,406 / 2,453
2 回目	2,418 / 2,428	2,406 / 2,424
3 回目	2,420 / 2,424	2,410 / 2,414

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	11,848 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-v -fast	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	2,882 / 2,892	2,936 / 2,944
2 回目	2,882 / 2,890	2,935 / 2,944
3 回目	2,883 / 2,890	2,935 / 2,944

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	13,552 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-O4 を指定	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	2,381 / 2,393	2,621 / 2,634
2 回目	2,381 / 2,390	2,619 / 2,635
3 回目	2,381 / 2,390	2,623 / 2,634

また、応募者からは以下の自己評価が報告されている。

プラットフォーム	:	Pentium III (866MHz), RAM 256MB
OS およびコンパイラ	:	Windows NT4.0, Visual C++ 6.0 SP5
使用言語	:	ANSI C (インラインアセンブラ有)
鍵スケジュール	:	993 cycles/key
暗号化	:	1,409 cycles/block
復号	:	1,423 cycles/block
鍵スケジュール + 暗号化	:	2,411 cycles
鍵スケジュール + 復号	:	2,402 cycles

3.3.1.6　ハードウェア実装評価結果

以下のブロック図 (図 3.1,3.2) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.30) を示す。なお、アルゴリズムに含まれる多数の乗算は、FPGA にハードマクロとして用意されている 18 ビット乗算器による繰返し処理によって実現されているため、他のアルゴリズムに比べて多くの clock 数を要している。

Key Setup Clock 数	35
Data Randomize Clock 数	265
実装鍵ビット数	128

表 3.30: CIPHERUNICORN-E の HW 実装評価結果

また、応募者からは、ASIC および FPGA 実装に関して以下の自己評価が報告されている。

ASIC プロセス	:	NEC 0.25μm CMOS ASIC Design Library
速度優先実装	:	108.00 Mbps, 1,034.3 Kgates
規模優先実装	:	39.00 Mbps, 966.2 Kgates
FPGA デバイス	:	ALTERA EP20K600EBC652-1
速度優先実装	:	21.76 Mbps, 9,013 cells + 118 ESB
規模優先実装	:	17.03 Mbps, 6,486 cells + 67 ESB

参考文献

[1] 角尾幸保、久保博靖、宮内宏、中村勝洋、「統計的手法により安全性が評価された暗号」、1998 年暗号と情報セキュリティシンポジウム SCIS’98, 4.2.B, 1998.

[2] 角尾幸保、太田良二、宮内宏、中村勝洋、「分散型暗号強度評価支援システム」、2000 年暗号と情報セキュリティシンポジウム, SCIS2000, A53, 2000.

[3] 鶴丸豊広、酒井康行、反町亨、松井充, 「64 ビットブロック暗号に対するタイミング攻撃」、2003 年暗号と情報セキュリティシンポジウム SCIS2003, 2D-3, 2003.

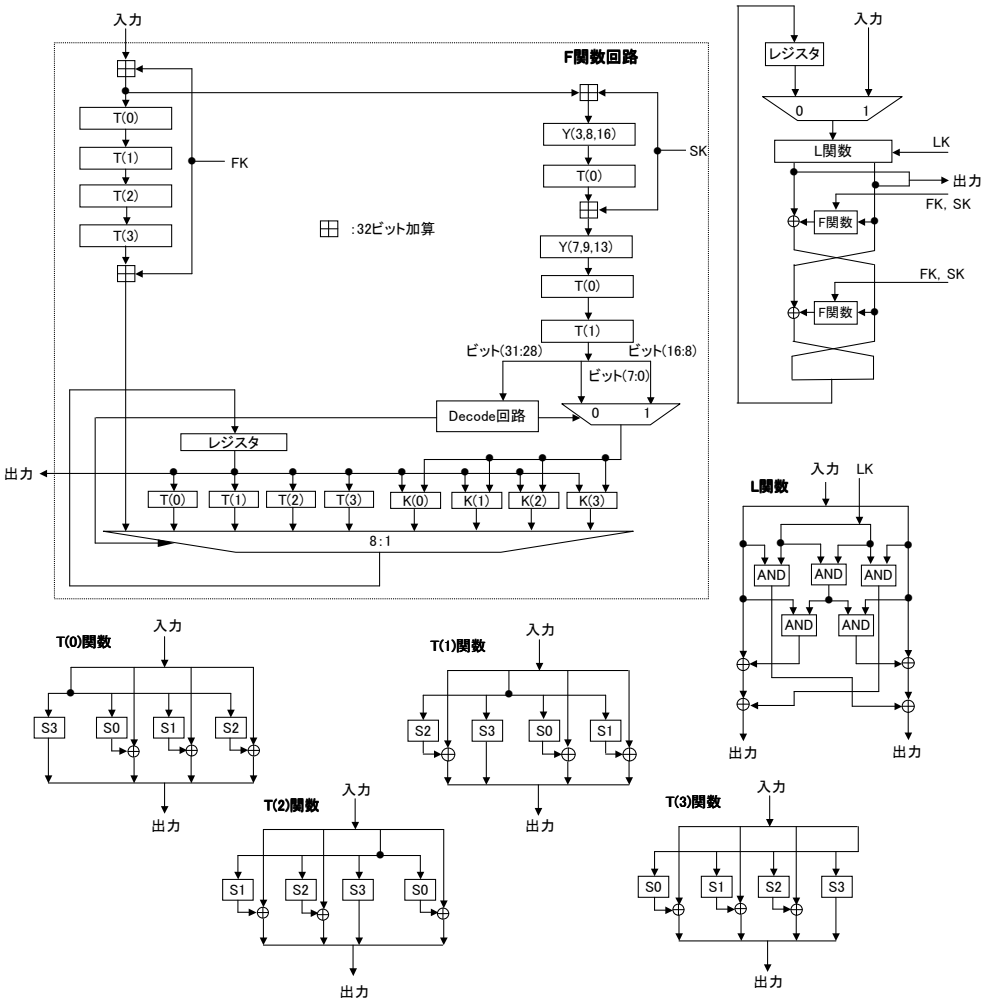

図 3.1: CIPHERUNICORN-E 暗号化回路ブロック図

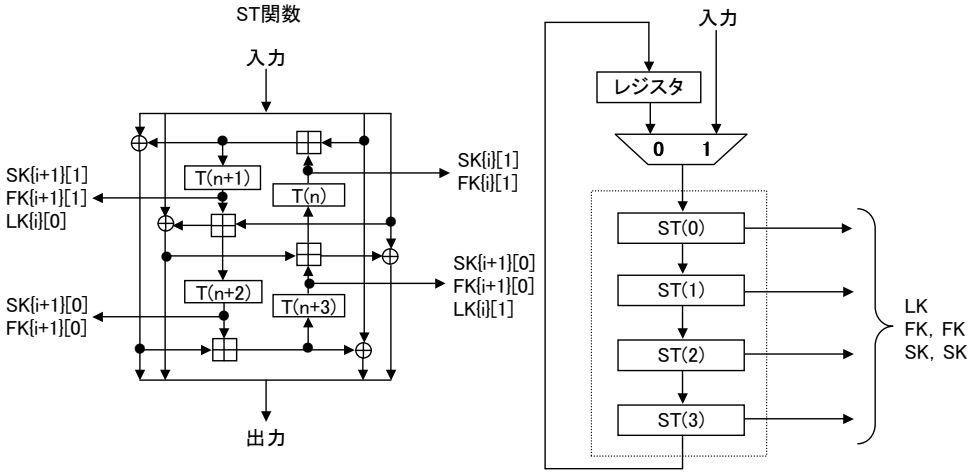

図 3.2: CIPHERUNICORN-E 鍵生成回路ブロック図

3.3.2　Hierocrypt-L1

3.3.2.1　技術概要

Hierocrypt-L1 は 2000 年 9 月 8 日に情報処理学会・コンピュータセキュリティ研究会において、株式会社東芝により提案された共通鍵ブロック暗号である。8 ビット S-box を 8 個並列に並べたバイト換字層 (S) と GF(2^8) 上の 4×4 MDS 行列を 2 個並列に並べたバイト置換層 (MDS_L)、GF(2^{32}) の 2×2 MDS 行列からなるバイト置換層 (MDS_H)、鍵加算層 (K) から構成される。ラウンド関数の一段は S から始まり MDS_L、K、S、MDS_H と続き K で終わる。最終段は S から始まり MDS_L、K、S と続き K で終わる。暗号化処理は K から始まり、ラウンド関数を 5 段繰り返した後、最終段の処理を一段行う。

技術のポイント　入れ子型 SPN の採用による計算効率と安全性の両立

3.3.2.2　技術仕様

■入出力鍵サイズ

- 入出力長: 64 ビット
- 鍵長: 128 ビット
- 段数: 6 段
- 構造: 入れ子型 SPN

■設計方針

- 主要な共通鍵暗号攻撃法に対して十分強く、主要なプラットフォーム上で高速で動作し、実装サイズもコンパクトになることを目標としている。
- 計算効率と安全性の両立を高めるため、データランダム化部には SPN 構造を再帰的に利用した入れ子型 SPN 構造を採用している。
- S-box は、ガロア体上のべき乗関数を基本として、差分攻撃・線形攻撃に対する耐性に関する最適化を行なっている。さらに、べき乗関数をビット置換とアフィン変換で挟むことにより代数的攻撃法の適用を困難にしている。
- 拡散層は、符号理論を用いて活性 S-box 数の下限が大きな値を取るものを多数生成して候補とし、安全性と実装効率の条件で絞り込んでいる。
- 鍵スケジュール部は、64 ビット Feistel 型構造を基本構造とし、中間出力を組み合わせて拡大鍵を生成する。復号時にも on-the-fly での鍵設定の初期遅延が小さくなるよう、中間鍵列が途中で逆転して戻ってくる折り返し型の構造を採用している。

3.3.2.3 その他

Hierocrypt-L1 は 128 ビットブロック暗号 Hierocrypt-3 とほぼ同一の構造を持つ。実装の仕方によっては、両者とも、データランダム化部のみの復号処理速度が暗号化のそれより僅かに遅いことがある (表 3.31、表 3.32 を参照)。

3.3.2.4 安全性評価結果

現時点 (2003 年 3 月) では、いくつかの安全性に関する解析結果が知られているが、いずれの結果も、Hierocrypt-L1 の安全性について、決定的な欠陥を示したものではない。

現在知られている Hierocrypt-L1 に対する最も効果的な攻撃法は SQUARE 攻撃であり、6 段 (12 層) 中 3.5 段 (7 層) まで攻撃可能であることが報告されている [1, 11]。より詳しくは、Ferguson らの手法 [4] に Type1 の拡張 [3] を適用することで、6 段 (12 層) 中 3 段 (6 層) の攻撃が可能であり [10]、さらにもう一段分の拡大鍵を推定することにより、3.5 段 (7 層) の攻撃が可能であることが確認されている [11]。その際に必要な選択平文数は 2^{37} であり、計算量は 2^{110} 回の暗号化処理と同等である。SQUARE 攻撃、およびその拡張は、特に Rijndael に類似の暗号に対して効果的な攻撃であり、今後も注視する必要があると思われる。しかし、現在のところ、脅威となる攻撃方法は見つかっていない。

設計者による自己評価書では、SQUARE 攻撃以外にも、共通鍵暗号のさまざまな攻撃法に対する安全性の評価が行われている。特に、差分攻撃・線形攻撃については、信頼性の高い評価を行っており、証明可能安全性については、継続的に証明を改良した結果を発表している [11, 12, 13, 16]。最新の結果 (2003 年 3 月現在) によれば、Hierocrypt-L1 は、各段の鍵が独立かつ一様との仮定の下で、Hong らの手法 [9] を用いて差分攻撃・線形攻撃に対して証明可能な安全性を持ち、最大差分確率・最大線形確率は 2 段以降 2^{-48} を超えない [16]。なお、3 段以上の段数において最大差分確率・最大線形確率が 2^{-48} より確実に小さな値をとるか否かは現在のところ明らかになっていない。ただし、ブロック長が 64 ビットのブロック暗号では、最大差分確率・最大線形確率の下限は 2^{-64} である。また、最大差分特性確率・最大線形特性確率は 2 段 (4 層) で 2^{-90} を超えない [13]。したがって、Hierocrypt-L1 は差分攻撃・線形攻撃に対して十分な強度を持つと考えられる。

この他、Truncated 差分に関しては、自己評価書において、6 段中 2.5 段 (5 層) でランダム置換と区別できなくなることが確認されている。

高階差分攻撃に対しては、同様に自己評価書において、S-box の代数次数が 7 次であること、代数構造を複雑にするため S-box の入力側でのビット置換が行われていること、また、拡散層に MDS 行列を用い S-box との組合せたときの多項式表現の項数の最大化を行なっていることから、効率的な高階差分が発見される可能性は極めて低いであろうと予想されている。ただし、多項式表現の項数は (比較的大きな値をとってはいるものの) 最大値でないことが詳細評価で確認されている。また、補間攻撃に関する検討 [6, 7]、実験的な乱数検定結果 [2] などが新しい結果が発表された。鍵スケジュール部についても、拡大鍵間にいくつかの線形関係式があることが確

認されている [5, 8]。

しかし、これらの結果は、直ちに Hierocrypt-L1 の安全性を脅かすものではない。Hierocrypt-L1 の設計における SPN 構造、S-box、MDS の選択はこれまでの暗号学の研究結果を踏まえたものであり、致命的な欠点は無いと考えられる。

■実装攻撃に対する安全性　Hierocrypt-L1 に対する実装攻撃として、ある種の特殊な条件のもとで、キャッシュメモリのヒット・ミスヒットの処理時間差を使ったタイミング攻撃を行うことにより、すべての秘密鍵が導出された例が報告されている [17]。本攻撃法は実装法や使用環境に依存した攻撃法であるので、Hierocrypt-L1 のアルゴリズム自体の安全性に致命的な欠点をもたらすものではなく、利用環境下における適切な実装攻撃対策を施せば実用上十分な安全性が確保されると考えられる。したがって、この種のタイミング攻撃に対する脅威がある環境において Hierocrypt-L1 を利用する場合には、このような実装攻撃に対する防御策を注意深く講じることが望まれる。防御策としては、有意な処理時間差が計測できないようにすることなどが挙げられる。実装攻撃の一般的概要、および対策法の詳細については、第 6 章も参照すること。

3.3.2.5　ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.31、表 3.32 のとおりである。

備考　UltraSPARC IIi と Alpha 21264 の測定において、(カッコ) 内の値は応募者による測定プログラムの改変した場合の測定値。測定プログラムは汎用性を持たせるため巨大なバッファ領域を確保しているが、その領域を必要な分だけ取るように改変した。速度評価の主旨を違えるような改変は行っていないことは確認済み。

また、応募者からは以下の自己評価が報告されている。

プラットフォーム	:	Mobile Pentium II (600MHz), 192MB
OS およびコンパイラ	:	Windows 2000 SP2, Sun JDK 1.3.1 without JCE
使用言語	:	Java
鍵スケジュール (暗号化)	:	1,125 cycles/key
暗号化	:	1,198 cycles/block
復号	:	1,249 cycles/block

表 3.31: Hierocrypt-L1 のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ (486 命令)	
プログラムサイズ	52,982 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	VC++6.0 速度優先オプション使用	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	199 / 201	204 / 206
2 回目	199 / 201	204 / 206
3 回目	200 / 201	204 / 205

UltraSPARC IIi (400MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	24,496 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -native -fast -xarch=v9 -xCC	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	378 (332) / 380 (336)	500 (304) / 504 (307)
2 回目	378 (332) / 380 (336)	500 (304) / 504 (308)
3 回目	378 (332) / 380 (336)	500 (304) / 504 (308)

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	84,328 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -O3	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	210 (179) / 214 (182)	210 (179) / 212 (182)
2 回目	210 (179) / 214 (182)	210 (179) / 212 (182)
3 回目	210 (179) / 213 (182)	210 (179) / 212 (182)

表 3.32: Hierocrypt-L1 の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ (486 命令)	
プログラムサイズ	52,982 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	VC++6.0 速度優先オプション使用	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	374 / 375	616 / 618
2 回目	374 / 377	616 / 617
3 回目	374 / 375	616 / 618

UltraSPARC IIi (400MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	24,496 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -native -fast -xarch=v9 -xCC	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	718 (616) / 721 (620)	1,203 (1,014) / 1,215 (1,031)
2 回目	718 (616) / 721 (619)	1,203 (1,012) / 1,215 (1,030)
3 回目	718 (616) / 721 (620)	1,203 (1,015) / 1,215 (1,031)

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	84,328 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -O3	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	390 (386) / 394 (389)	625 (617) / 654 (648)
2 回目	390 (386) / 394 (389)	625 (617) / 653 (648)
3 回目	390 (386) / 394 (389)	625 (617) / 653 (648)

■IC カード実装他 応募者からは、8 ビットプロセッサ Z80 (5MHz) における実装例 (Z80 シミュレータおよび東芝製 IC カード JT6N55) が以下のように示されている。なお、処理時間には暗号化・復号処理に加え必要な鍵生成処理も含む。

実装環境	Z80 シミュレータ				JT6N55		
評価項目	ROM [bytes]	RAM [bytes]	stack [bytes]	処理時間 [states]	ROM [bytes]	RAM [bytes]	処理時間 [states]
暗号化	2,228	25	16	18,384	2,447	26	19,399
復号	3,200	25	16	21,588	—	—	—
暗復共用	4,196	—	—	—	—	—	—

3.3.2.6 ハードウェア実装評価結果

以下のブロック図 (図 3.3,3.4,3.5) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.33) を示す。

Key Setup Clock 数	15
Data Randomize Clock 数	12
実装鍵ビット数	128

表 3.33: Hierocrypt-1 の HW 実装評価結果

また、応募者からは、ASIC および FPGA 実装に関して以下の自己評価が報告されている。

ASIC プロセス : 0.25μm CMOS ASIC Design Library
速度優先実装 : 1,081 Mbps, 81.2 Kgates
規模優先実装 : 135.0 Mbps, 9.9 Kgates

ASIC プロセス : 0.13μm CMOS ASIC Design Library
速度優先実装 : 1,568 Mbps, 54.9 Kgates

FPGA デバイス : ALTERA Max+plus II ver. 9.6
速度優先実装 : 51.0 Mbps, 11.0 Kcells

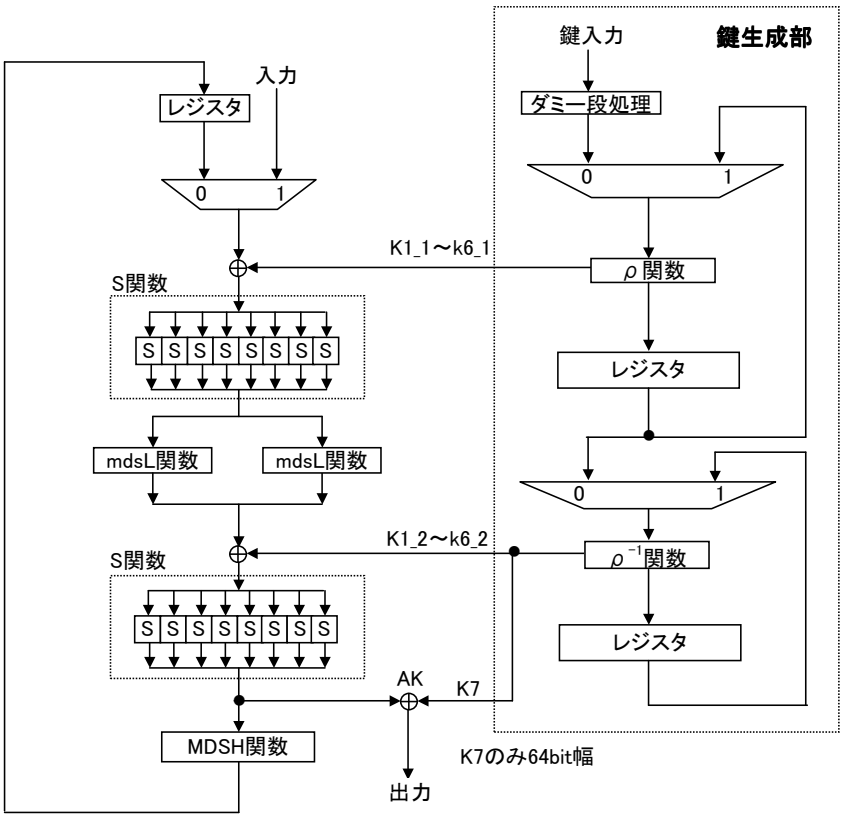

図 3.3: Hierocrypt-L1 暗号化回路・鍵生成部回路 (右側点線内) ブロック図

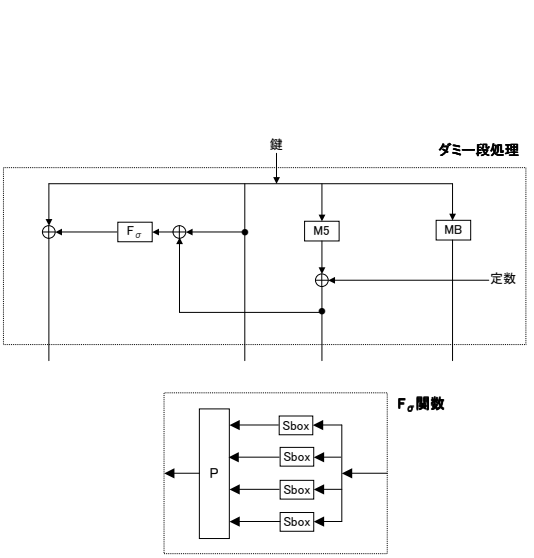

図 3.4: ダミー段処理・F_σ 関数内部ブロック図

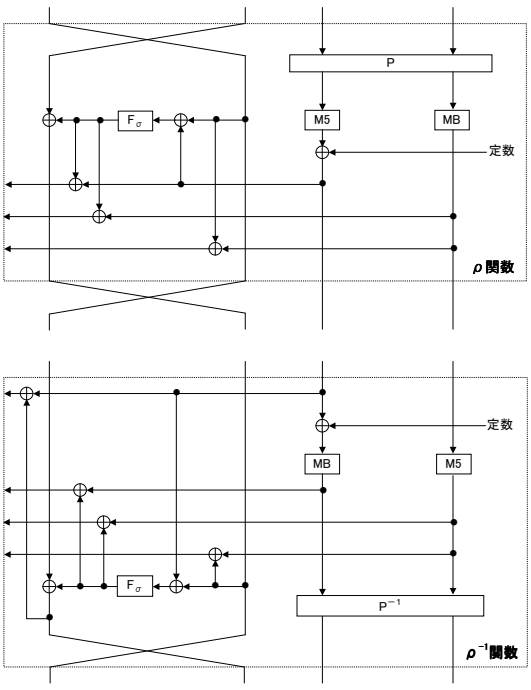

図 3.5: ρ・ρ^{-1} 関数内部ブロック図

参考文献

[1] P. S.L.M. Barreto, V. Rijmen, J. Nakahara Jr., B. Preneel, J. Vandewalle, and H. Y. Kim, “Improved Square Attacks Against Reduced-Round Hierocrypt,” Fast Software Encryption, 8th International Workshop, FSE 2001, LNCS 2355, pp.173–182, Springer-Verlag, 2001.

[2] Y. Braziler, “The statistical evaluation of the NESSIE submission Hierocrypt-L1,” Public reports of NESSIE project, NES/DOC/TEC/WP3/022/1, available at http://www.cosic.esat.kuleuven.ac.be/nessie/reports/.

[3] J. Daemen, L. R. Knudsen, and V. Rijmen, “The block cipher Square,” Fast Software Encryption, 4th International Workshop, FSE’97, LNCS 1267, pp.149–165, 1997.

[4] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, “Improved Cryptanalysis of Rijndael,” Fast Software Encryption, 7th International Workshop, FSE 2000, LNCS 1978, pp.213–230, available at http://www.counterpane.com/rijndael.html, 2000.

[5] S. Furuya and V. Rijmen, “Observations on Hierocrypt-3/L1 Key-scheduling Algorithms,” Proceedingds of the second open NESSIE Workshop, 2001.

[6] 古屋聡一、櫻井幸一、「SPN 構造をもつブロック暗号の代数近似について」、第 4 回コンピュータセキュリティシンポジウム (CSS2001) 講演予稿集, CSS2001, 6B-1, 2001.

[7] 古屋聡一、櫻井幸一、「Sudan の Reed-Solomon 符号復号アルゴリズムを使ったブロック暗号の補間攻撃」、電子情報通信学会技術研究報告, COMP2002-22, 2002.

[8] 金丸昌司、白井太三、阿部譲司、「Hierocrypt-L1/3 の鍵スケジュール解析」、電子情報通信学会技術研究報告, ISEC2002-91, 2002.

[9] S. Hong, S. Lee, J. Lim, J. Sung, and D. Cheon, “Provable Security against Differential and Linear Cryptanalysis for the SPN Structure,” Fast Software Encryption, 7th International Workshop, FSE 2000, LNCS 1978, pp.273–283, 2001.

[10] 村谷博文、大熊建司、佐野文彦、本山雅彦、川村信一、「64 ビット版 Hierocrypt の提案」、情報処理学会研究報告, CSEC11-9, 2000.

[11] 大熊建司、佐野文彦、村谷博文、本山雅彦、川村信一、「ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 の安全性について」、2001 年暗号と情報セキュリティシンポジウム SCIS2001 講演予稿集, SCIS2001 11A-4, 2001.

[12] 大熊建司、佐野文彦、清水秀夫、川村信一、「入れ子型 SPN 構造の証明可能安全性に関する補足事項」、2002 年暗号と情報セキュリティシンポジウム SCIS2002 講演予稿集, SCIS2002 5B-2, 2002.

[13] K. Ohkuma, H. Shimizu, F. Sano, and S. Kawamura, “Security Assessment of Hierocrypt and Rijndael against the Differential and Linear Cryptanalysis (Extended Abstract),” IACR’s ePrint archive, 2001/070, available at http://eprint.iacr.org/.

[14] B. Preneel, B. Van Rompay, L. Granboulan, G. Martinet, S. Murphy, R. Shipsey, J. White, M. Dichtl, P. Serf, M. Schafheutle, E. Biham, O. Dunkelman, M. Ciet, J-J. Quisquater, F. Sica, L. Knudsen, and H. Raddum,“NESSIE Phase I: Selection of Primitives,” NESSIE deliverables,

available at http://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/.

[15] B. Van Rompay, V. Rijmen, and J. Nakahara Jr., “A first report on CS-Cipher, Hierocrypt, Grand Cru, SAFER++, and SHACAL,” Public reports of NESSIE project, NES/DOC/KUL/WP3/006/1, available at http://www.cosic.esat.kuleuven.ac.be/nessie/reports/.

[16] F. Sano, K. Ohkuma, H. Shimizu, and S. Kawamura, “On the Security of Nested SPN Cipher against the Differential and Linear Cryptanalysis,” IEICE TRANS. FUNDAMENTALS, vol.E86-A, No.1, pp.37–46, 2003.

[17] 鶴丸豊広、酒井康行、反町亨、松井充、「64 ビットブロック暗号に対するタイミング攻撃」、2003 年暗号と情報セキュリティシンポジウム SCIS2003 講演予稿集, SCIS2003 2D-3, 2003.

3.3.3 MISTY1

3.3.3.1 技術概要

MISTY1 は 1996 年に三菱電機株式会社が開発したブロック長 64 ビット、鍵長 128 ビットの共通鍵ブロック暗号であり [11, 12]、今回、三菱電機から提案された。MISTY1 は Feistel 型構造と鍵依存線形変換を用いており、Feistel 型構造の内部関数には、変形 Feistel 型構造を再帰的に組合せた関数が用いられている。この構造によって、MISTY1 は差分攻撃・線形攻撃に対する証明可能安全性を持つことが示されている [9, 10]。実装面では IC カード向け 8 ビットプロセッサから 64 ビット RISC プロセッサまであらゆる分野に適した暗号であり、とりわけレジスタが多いプロセッサでは Bitslice 実装法によってソフトウェアによる高速処理が実現できる点は大きな特徴である (Alpha プロセッサでは 68 cycles/block)[13, 14, 15, 16, 17]。さらにハードウェアでは 10 Kgates 以下という極めて小さいサイズで実装可能であることも特徴の一つである [1, 2]。暗号の発表以来 5 年以上が経過しており、豊富な利用実績をもつ暗号である。

3.3.3.2 技術仕様

■大まかな構造

- ブロック長 64 ビット、鍵長 128 ビットのブロック暗号である。
- Feistel 型構造と鍵依存線形変換 (FL 関数) を用いており、Feistel 型構造の内部関数には、変形 Feistel 型構造を再帰的に組合せた関数が使われている。
- 段数は 4 の倍数の範囲で可変であり、推奨段数は 8 段である。

■設計方針

- 安全性に関する何らかの数値的な根拠を持つこと。特にブロック暗号の汎用的で強力な解読法である差分攻撃と線形攻撃に対する証明可能安全性の理論を用い、再帰構造を用いることで小さく安全な関数から大きく安全な暗号が構成されている。
- プロセッサの種類によらずソフトウェアで実用的な性能を達成すること。できる限り多くのアプリケーションで利用可能な暗号をめざし、特定のプロセッサでのみ高速処理が可能となるような命令を用いず、あらゆるプロセッサで適度な高速性と小型化が実現できる基本的な命令のみが採用されている。また IC カードでの実装を考慮しワークメモリサイズが小さくなるよう設計されている。
- ハードウェア上で十分な高速性を実現すること。算術演算はハードウェアでの速度低下につながることがあるので採用せず、論理演算とテーブル参照だけからアルゴリズム全体を構成されている。またテーブルの設計においてはハードウェアで最適化されるように考慮されている。

3.3.3.3 その他

MISTY1 が発表される 1 年前に開発者によって、MISTY1 で用いる変形 Feistel 型構造とその安全性に関する理論、ならびにこの構造を用いたブロック暗号の具体例が複数示されている。それらに暗号名は無いが、その一つ、Algorithm 1 として記述された暗号が MISTY1 のベースになっていると考えられる [9, 10]。MISTY1 と同時に、同じく差分攻撃・線形攻撃に対する証明可能安全性を持つ 64 ビットブロック暗号 MISTY2 が発表されている [11, 12]。

標準化関連として、ISO/IEC 9979 アルゴリズム公開登録、RFC 2994 (Informational)、ISO/IEC 18033-3 (Committee Draft)、NESSIE に掲載されている。また、MISTY1 を携帯電話用にカスタマイズしたアルゴリズムとして KASUMI が 3GPP を中心にして開発され、2000 年 3 月に次世代携帯電話 (W-CDMA) における秘匿と完全性アルゴリズムのコア部分として採用されている [26]。

3.3.3.4 安全性評価結果

■総評 MISTY1 のデータランダム化部は 3 段で最大差分確率・最大線形確率が 2^{-56} 以下になることが理論的に証明されており、差分攻撃・線形攻撃については十分安全であると考えられ、さらに詳細評価での解析の結果、データランダム化部、鍵スケジュール部への従来型攻撃に対する安全性についても問題はないと判断される。

■データランダム化部に対する安全性 差分攻撃、線形攻撃、丸め差分攻撃、カイ 2 乗攻撃、分割攻撃、高階差分攻撃、補間攻撃、不能差分攻撃、mod n 攻撃、非全単射攻撃、Luby-Rackoff 流ランダム性について解析した結果、標準仕様の MISTY1 について攻撃が有効となるものは認められなかった。なお、MISTY1 の推奨段数は 8 段であるが、5 段以下 (FL 関数を省いた場合には 7 段以下) であれば、改良型高階差分攻撃を用いて鍵の全数探索より少ない計算量で拡大鍵の一部が推定可能であるとの結果が報告されている [21, 22, 23, 20]。また計算機を用いた解読実験では、5 段 MISTY1 (FL 関数あり) の拡大鍵の一部が約 1 時間で導出されたとの結果が発表されている [5] (表 3.34 参照)。さらに、不能差分を用いた攻撃 [8]、Integral Cryptanalysis (Square 攻撃) を用いた攻撃 [7, 6]、複数変数による補間多項式を用いた攻撃 [19, 5] 等が発表されている。しかしこれらの攻撃はいずれも段数を減らすなど標準仕様と異なるものに対する解析結果であり、現時点では標準仕様の MISTY1 の安全性に影響を与える結果は知られていない。

■鍵スケジュール部に対する安全性 鍵の全数探索、弱鍵・準弱鍵、関連鍵攻撃、スライド攻撃について解析した結果、一部の解読法についてはその有効性を否定しきることはできないものの、暗号全体の安全性を脅かすに至る攻撃法は認められなかった。

MISTY1 の鍵スケジュール部は 128 ビットを 16 ビット単位に分割した 8 個の鍵変数 $K_1, K_2, \ldots, K_8$ について、各段で異なる順序に並べ替えたものを用いている。そのため、これら 16 ビットの値について $K_1 = K_2 = \cdots = K_8$ が成り立つ場合には、すべての段の拡大鍵が等しくなる。こ

表 3.34: 攻撃段数と解読必要計算量

	MISTY1		FL 関数を除いた MISTY1	
段数	データ量	計算量	データ量	計算量
4 段	$2^{8.4}$	2^{85}	2^{4}	$2^{7.2}$
5 段	2^{22}	2^{33}	$2^{10.5}$	2^{17}
6 段	—	—	2^{12}	2^{93}
7 段	—	—	2^{39}	2^{129}

の性質より、全秘密鍵 2^{128} 個中 $K_1 = K_2 = \cdots = K_8$ を満たす 2^{16} 個の秘密鍵は、MISTY1 から FL 関数を除いた暗号について、スライド攻撃に対する弱鍵となるものと考えられる。ただし、この攻撃は、FL 関数を含めた暗号に適用するのは困難であること、さらに弱鍵として考えられる鍵の個数が全体に比べ非常に少ないことから、MISTY1 の安全性に対する脅威とはならないと考えられる。

■実装攻撃に対する安全性 MISTY1 に対する実装攻撃として、ある種の特殊な条件のもとで、キャッシュメモリのヒット・ミスヒットの処理時間差を使ったタイミング攻撃を行うことにより、すべての秘密鍵が導出された例が報告されている [25]。

本攻撃法は実装法や使用環境に依存した攻撃法であるので、MISTY1 のアルゴリズム自体の安全性に致命的な欠点をもたらすものではなく、利用環境下における適切な実装攻撃対策を施せば実用上十分な安全性が確保されると考えられる。したがって、この種のタイミング攻撃に対する脅威がある環境において MISTY1 を利用する場合には、このような実装攻撃に対する防御策を注意深く講じることが望まれる。防御策としては、有意な処理時間差が計測できないようにすることなどが挙げられる。実装攻撃の一般的概要、および対策法の詳細については、第 6 章も参照すること。

3.3.3.5 ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.35、表 3.36 のとおりである。

復号の処理時間が暗号化処理時間に比べ最大で 5clocks 程度の増減が見られるが、概ね同じ値であった。

また、応募者からは以下の自己評価が報告されている。

プラットフォーム	:	Pentium III (800MHz)
OS およびコンパイラ	:	Windows98
使用言語	:	アセンブラ
鍵スケジュール	:	230 cycles/key
暗号化	:	207 cycles/block
鍵スケジュール (Bitslice 実装)	:	46 cycles/key
暗号化 (Bitslice 実装)	:	169 cycles/block

プラットフォーム	:	Alpha 21264 (667MHz)
OS およびコンパイラ	:	UNIX
使用言語	:	アセンブラ
鍵スケジュール	:	200 cycles/key
暗号化	:	197 cycles/block
鍵スケジュール (Bitslice 実装)	:	17 cycles/key
暗号化 (Bitslice 実装)	:	71 cycles/block

表 3.35: MISTY1 のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	21,353 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション		
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	213 / 215	208 / 210
2 回目	213 / 215	208 / 210
3 回目	213 / 214	209 / 211

Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	15,632 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション		
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	203 / 205	206 / 208
2 回目	203 / 206	206 / 208
3 回目	203 / 205	206 / 208

表 3.36: MISTY1 の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	17,681 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション		
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	357 / 358	350 / 351
2 回目	357 / 358	350 / 351
3 回目	357 / 358	350 / 351

Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	10,088 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション		
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	334 / 338	337 / 340
2 回目	337 / 338	337 / 340
3 回目	334 / 338	337 / 340

■IC カード実装他 応募者からは、組み込み用 16 ビットマイコン M16C (20MHz) による実装例、8 ビットマイコン H8/300 (3.57MHz) による実装例が示されている [17]。また、佐野らにより 8 ビットプロセッサ Z80 (5MHz) での実装例が報告されている [18]。

プロセッサ	暗号化 [cycles/block]	鍵スケジュール [cycles/key]	ROM [bytes]	RAM [bytes]
M16C	1,877	743	3,400	64
H8/300	6,018	1,240	1,934	43
Z80	25,486 (鍵スケジュール込)		1,598	44

3.3.3.6　ハードウェア実装評価結果

以下のブロック図 (図 3.6,3.7) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.37) を示す。なお、今回の実装においては、評価基板の仕様に適合させるため、中間でラッチを挿入している。そのため、ブロック図での実装より Data Randomize Clock 数が増加している。

Key Setup Clock 数	1
Data Randomize Clock 数	16
実装鍵ビット数	128

表 3.37: MISTY1 の HW 実装評価結果

また、応募者からは、ASIC および FPGA 実装に関して以下の自己評価が報告されている。なお、処理回路には暗号化・復号処理部および鍵スケジュール部のすべてが含まれる。この他にも、ASIC 実装例として [1, 2]、FPGA 実装例として [3, 4] が発表されている。

ASIC プロセス　:　Mitsubishi Electric 0.18μm CMOS ASIC Design Library
速度優先実装　:　2,800.9 Mbps, 71.11 Kgates
規模優先実装　:　70.2 Mbps, 5.39 Kgates

FPGA デバイス　:　Xilinx XCV1000EBG560-8
速度優先実装　:　455.8 Mbps, 3,593 units
規模優先実装　:　250.9 Mbps, 1,462 units
パイプライン実装　:　13,330.6 Mbps, 6,432 units

FPGA デバイス　:　Xilinx Vertex 1000E
速度優先実装　:　497.0 Mbps, 3,770 slices
規模優先実装　:　281.5 Mbps, 1,372 slices

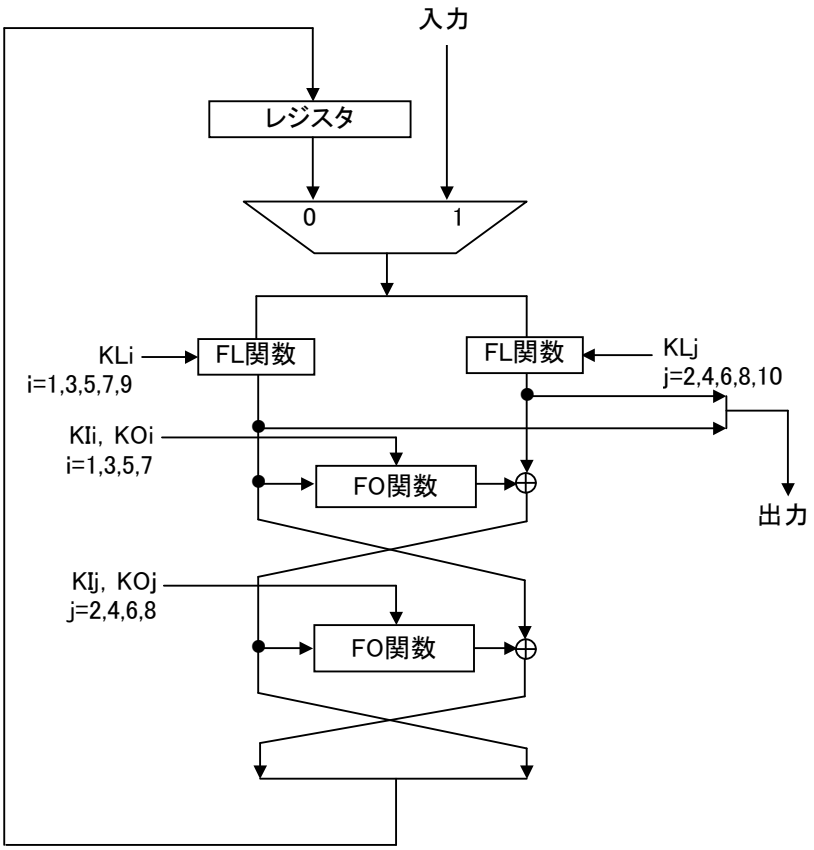

図 3.6: MISTY1 暗号化回路ブロック図

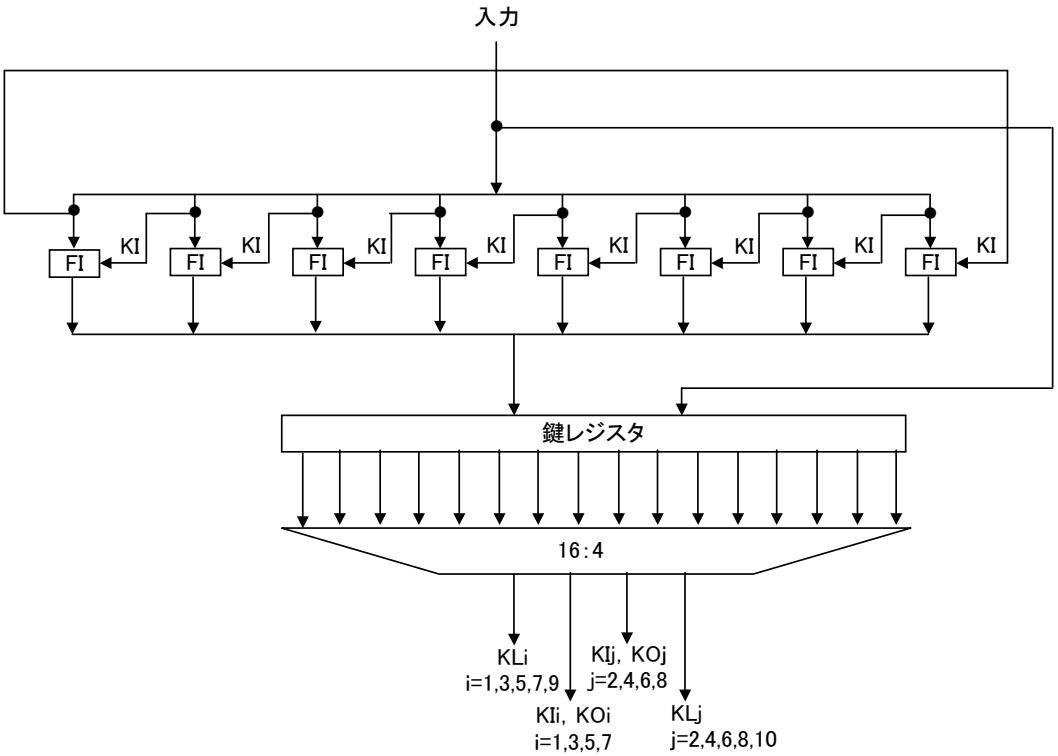

図 3.7: MISTY1 鍵生成回路ブロック図

参考文献

[1] 市川哲也、加藤潤二、松井充、「秘密鍵暗号 MISTY1 の H/W 実装における一方法」、Proceedings of SCIS98, SCIS98-9.1.A, 1998.

[2] 市川哲也、反町亨、松井充、「秘密鍵暗号 H/W 設計に関する考察」、Proceedings of SCIS97, SCIS97-9.D, 1997.

[3] 市川哲也、反町亨、粕谷智巳、松井充、「NESSIE 提案ブロック暗号のハードウェア実装について (I)」、Proceedings of SCIS2002, SCIS2002-12C-3, 2002.

[4] 市川哲也、反町亨、粕谷智巳、「FPGA を用いたブロック暗号のハードウェア実装評価」、Proceedings of SCIS2003, SCIS2003-12D-3, 2003.

[5] 秦野康生、田中秀磨、金子敏信、「MISTY1 の高階差分攻撃」、Proceedings of SCIS2002, SCIS2002-13A-5, 2002.

[6] I. Kim, Y. Yeom, and H. Kim, “Square Attacks on the Reduced-Round MISTY1,” Proceedings of SCIS2002, SCIS2002-13A-2, 2002.

[7] L. Knudsen and D. Wagner, “Integral Cryptanalysis,” Pre-proceedings of the International workshop of Fast Software Encryption 2002, Lecture Notes in Computer Science 2365, Springer Verlag, pp.112–127, 2002.

[8] U. Kuehn, “Improved cryptanalysis of MISTY1,” Proceedings of the International workshop of Fast Software Encryption 2002, pp.61–75, Lecture Notes in Computer Science 2365, Springer Verlag, 2002.

[9] 松井充、市川哲也、反町亨、時田俊雄、山岸篤弘、「差分解読法と線型解読法に対する証明可能安全性をもつ実用ブロック暗号」、Proceedings of SCIS 1996 SCIS96-4C, 1996.

[10] M. Matsui, “New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis,” Proceedings of the 3rd international workshop of Fast Software Encryption, Lecture Notes in Computer Science 1039, pp.205–218, Springer Verlag, 1996.

[11] 松井充、「ブロック暗号アルゴリズム MISTY」、信学技報 ISEC96-11, 1996.

[12] M. Matsui, “New Block Encryption Algorithm MISTY,” Proceedings of the 4-thinternational workshop of fast software encryption, Lecture Notes in Computer Science 1267, Springer Verlag, pp.54–68, 1997.

[13] 中嶋純子、松井充、「MISTY のソフトウエアによる高速実装について (I)」、信学技報 ISEC97-12, 1997.

[14] 中嶋純子、松井充、「MISTY のソフトウエアによる高速実装について (II)」、Proceedings of SCIS 98, SCIS98-9.1.B, 1998.

[15] J. Nakajima and M. Matsui, “Fast Software Implementation of MISTY1 on Alpha Processors,” IEICE Trans. Funcationals, Vol E82-A, No.1 January 1999.

[16] 中嶋純子、松井充、「MISTY のソフトウエアによる高速実装について (III)」、信学技報 ISEC2000-81, 2000.

[17] 中嶋純子、松井充、「共通鍵暗号 MISTY1 の最適なソフトウエア実装について」、Proceedings of SCIS 2001, 13A-3, 2001.

[18] F. Sano, M. Koike, S. Kawamura, and M. Shiba, “Performance Evaluation of AES Finalists on the High-End Smart Card,” 3rd AES Conference, New York, 2000.

[19] 渋谷香士、小林篤、下山武司、辻井重男、「MISTY1 の内部関数の多項式表現と暗号解析への応用」、Proceedings of SCIS2002, SCIS2002-10A-3, 2002.

[20] 白田麻貴子、杉尾信行、秦野康生、田中秀麿、金子敏信、「MISTY1 における上一段消去法の適用に関する一考察」、Proceedings of SCIS 2003 6D-3, pp.457–462. 2003.

[21] H. Tanaka, M. Hisamatsu, and T. Kaneko, “Higher Order Differential Attack of MISTY1 without FL functions,” JWIS’98, ISEC98-66, pp.143–150, 1998.

[22] 田中秀磨、石井周志、金子敏信、「霞と MISTY の強度評価に関する一考察」、Proceedings of SCIS 2001 12A-1, pp.647–652. 2001.

[23] 田中秀磨、金子敏信、「FL 関数の無い 6 段 MISTY1 の攻撃について」、信学技報 ISEC2002-41, 2002.

[24] 田中秀磨、杉尾信行、金子敏信、「鍵スケジュールを考慮したブロック暗号に対する攻撃に関する一考察」、Proceedings of SCIS 2003 5D-3, pp.363–368. 2003.

[25] T. Tsunoo, E. Tsujihara, K. Minematsu, and H. Miyauchi, “Cryptanalysis of Block Ciphers Implemented on Computers with Cache,” ISITA, Xi’an, PRC, October 7-11, 2002.

[26] 3GPP, KASUMI, ETSI/SAGE Specification, 1999.
(http://www.etsi.org/dvbandca/3GPP/3gppspecs.htm)

3.3.4　Triple DES

3.3.4.1　技術概要

Triple DES[14] は、1979 年に IBM の Tuchman により提案された。1977 年に米国政府標準 (FIPS PUB: Federal Information Processing Standards Publications) の暗号として認定された共通鍵ブロック暗号である DES (Data Encryption Standard) [12] の組合せ暗号であり、DES を 3 回繰り返すことにより暗号強度を高めている。現在 Triple DES は、Triple Data Encryption Algorithm (TDEA) として米国の ANSI (American National Standards Institute) X9.52 に 7 種類の利用モードとともに規定され、FIPS 化 (FIPS PUB 46-3) も行われている [1, 12]。

3.3.4.2　技術仕様

Triple DES は、Feistel 型暗号である DES を 3 回繰り返す構造をとっているため、DES と同じく 64 ビット入出力サイズの共通鍵ブロック暗号に分類される。平文を P、暗号文を C、鍵を K、鍵 K による暗号化および復号処理を E_K および D_K とすると、暗号化処理および復号処理は次のように表すことができる。

暗号化　$C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$
復号　　$P = D_{K_1}(E_{K_2}(D_{K_3}(C)))$

この時、鍵 K_1、K_2、K_3 の取り方で次の三種類のオプションがとられる [1]。

(1) K_1、K_2、K_3 が独立
(2) K_1 と K_2 が独立で、$K_1 = K_3$
(3) $K_1 = K_2 = K_3$

特に (3) は、3 つの鍵をすべて同じとすることで、通常の “(Single) DES” との間で互換性がとられている。一般に (1) は “3-key Triple DES”、(2) は “2-key Triple DES” と呼ばれる。DES の鍵長が 56 ビットであることから、(1)~(3) の鍵長は各々 168 ビット、112 ビット、56 ビットである。

Triple DES の暗号利用モードは ANSI X9.52 の中で、ISO 8372 で規定される 64 ビットブロック暗号の利用モード (ECB、CBC、CFB、OFB) をベースに拡張した利用モード (TECB、TCBC、TCFB、TOFB) と、その他 (TCBC-I、TCFB-P、TOFB-I) の計 7 つの暗号利用モードが規定されている [1]。

3.3.4.3　その他

■周辺状況　発表当初から DES の 56 ビットという鍵長は短く、鍵の全数探索に対して安全ではないという懸念が出されていた [16]。そのため DES をカスケード接続して使用することにより鍵長を増やすことが議論された結果、生まれたのが Triple DES である。DES を 2 回ではなく 3

回繰り返しているのは、中間一致攻撃 (meet-in-the-middle attack) [4] を避けるためである。実際に DES は発表から 20 年後の 1997 年に米国 RSA Laboratories が主催する解読コンテスト (DES Challenge I) で解読に成功し、さらに DES Challenge III (1999 年) において約 22 時間で解読されたという報告がある [17]。米国では Triple DES の FIPS 化が完了しており、米国政府機関だけでなく、一般の DES ユーザの間でも Triple DES に移行する動きは更に拡大することが予想される。

■SSL/TLS における DES の安全性 SSL [13]/TLS [6] における DES は、40bit-key (Single) DES、56bit-key (Single) DES、168bit-key Triple DES (3-key Triple DES) の 3 種類がデータの秘匿の目的で利用される。なお、40bit-key (Single) DES とは、(Single) DES において鍵長を通常の 56 ビット鍵から 40 ビット鍵に短縮化したものである。いずれも暗号利用モードとしては CBC モードが用いられる。

まず、(Single) DES に関しては、後述するように鍵の全数探索に対して、もはや十分な安全性があるとは言えなくなった。従って、SSL/TLS における DES として、40bit-key (Single) DES と 56bit-key (Single) DES の利用は安全性の面から避けるべきである。

次に、Triple DES に関しては、まず SSL/TLS の bulk encryption として Triple DES を選択する際に注意しなければならないのは、2^{32} ブロック以上を同じセッション鍵を用いて暗号化することである。SSL/TLS において Triple DES は CBC モードで利用され、かつブロック長は 64 ビットであるため、2^{32} ブロック以上を同じセッション鍵を用いて暗号化すれば、暗号文一致攻撃により暗号文から平文に関する 1 ビットの情報が漏れる可能性が高くなる。ただし、ブロック長 64 ビットの 2^{32} ブロック分は 32GB であるため、セッション鍵の更新を適宜行うことによりこの問題は避けられる。

また後述するように 168bit-key (3-key) Triple DES は 2^{56} 組の選択平文/暗号文対を用いて学術的な意味での解読が可能である。そのため、次に注意しなければならないのは 2^{56} ブロック以上を同じセッション鍵を用いて暗号化することであるが、この攻撃に要する計算量は $2^{108.2}$ 程度と非常に大きく、また、ブロック長 64 ビットの 2^{56} ブロック分も 512PB と非常に大きいためこの攻撃に対する脅威は無視してよいであろう。

■標準化関連情報 Triple DES は、FIPS PUB 46-3 のほか、ANSI X9.52-1998、ANSI X9.65 (Working Draft)、ISO/IEC 18033-3 (Committee Draft)、SSL3.0 および RFC 2246: TLS1.0 (Proposed Standard) などに掲載されている。また、NIST は、2002 年 11 月に (Triple) DES に関する標準化情報を発表している。それによれば、2004 年に現行の FIPS PUB 46-3 を廃止し、FIPS PUB 46-4 に改訂する予定である。FIPS PUB 46-4 になると、鍵長 56 ビット以下の (Single) DES は正式に米国政府標準暗号から除外されることになるため、今後の DES の利用にあたっては、3-key Triple DES を採用するよう、強く勧告している。

3.3.4.4 安全性評価結果

■総評 Triple DES ((1) 3-key Triple DES、(2) 2-key Triple DES、(3) (Single) DES) についてこれまでに報告されてきた主な安全性の評価結果を、表 3.38 に示す。(Single) DES は代表的な Short Cut Method である差分攻撃や線形攻撃に対して、鍵の全数探索よりも効率よく解読可能 (すなわち学術的な意味で解読可能) であることが報告されている。また、(Single) DES の鍵の全数探索に対する計算量 2^{56} に関しては、解読コンテスト (DES Challenge III) で約 22 時間で解読に成功したという報告もあり [17]、もはや現実的な意味で解読可能な領域に達していると言える。2-key Triple DES および 3-key Triple DES は代表的な Short Cut Method である差分攻撃や線形攻撃に対しては安全であると言えるが、組合せ暗号であることに着目した中間一致攻撃によって、鍵の全数探索よりも効率よく解読可能であることが報告されている。特に 2-key Triple DES は、2^{57} の計算量 (選択平文数 2^{56}) で学術的な意味で解読可能であるが、これは鍵の全数探索の 2 倍ほどの計算量であるため、もはや現実的な意味でも解読可能な領域に達しつつあると言える。一方、3-key Triple DES も $2^{108.2}$ の計算量 (選択平文数 2^{56}) で学術的な意味で解読可能であるが、これは現在の計算機の計算能力からすると、現実的な意味では当面の間は安全であると考えられる。以上の結果をまとめると、3-key Triple DES であれば当面の間の使用は問題ないと言える。

■Brute Force Method に対する安全性 Triple DES (2-key Triple DES、3-key Triple DES) に関しては鍵の全数探索に対して現時点で十分安全であると考えられている。(Single) DES の 56 ビット鍵に関しては、1997 年に米国 RSA Laboratories が主催する解読コンテスト (DES Challenge I) で解読に成功し、さらに DES Challenge III (1999 年) において約 22 時間で解読されたという報告があり [17]、もはや十分な安全性があるとは言えないようになった。一方、Triple DES は組合せ暗号であるため、ある条件の下では、鍵長が拡大するほどには実質的な安全性は向上しないことが示されている。代表的な例として、Merkle と Hellman が提案した選択平文攻撃では、2-key Triple DES の場合は 2^{57} (鍵の全数探索では 2^{112})、3-key Triple DES の場合は 2^{112} (鍵の全数探索では 2^{168}) と、鍵の全数探索に対して大幅に計算量を削減することが可能であることが示されている [11]。ただし、本解読法においては、解読成功確率を 50% とした場合、必要となる選択平文数が 2^{55} であり、平文と鍵のペアを記憶するのに必要な外部記憶媒体が 4.03×10^{10} Gbits と膨大になるほか、必要な情報を通信回線経由で入手するためにも困難を伴うことが指摘されており、現時点では本解読法が現実の脅威となる可能性は低いと考えられている [8]。なお、Lucks は、Merkle と Hellman による選択平文攻撃の処理回数を削減する解読方法を提案しており、3-key Triple DES に対して、2^{108} 程度の計算量で解読できると報告している [9]。ただし、Lucks による解読法もまた、必要となる記憶媒体等から現実の脅威となる可能性は現時点では低いと考えられている。また、2-key Triple DES に関しては、Oorschot と Wiener が Merkle と Hellman による選択平文攻撃をもとに拡張した既知平文攻撃を提案し、既知平文数 N に対して $120 - N$ ビットの記憶媒体を用意すれば $2^{120-log_2 N}$ という計算量で解読できると指摘している [15]。ただし、本解読法も現実的な脅威となるには今後数十年かかると予想されている [9]。

表 3.38: Triple DES の主要な安全性評価結果 (解読に必要な計算量*1)

	Single DES	2-key Triple DES	3-key Triple DES
● Brute Force Method			
鍵の全数探索	2^{56}	2^{112}	2^{168}
Merkle-Hellman 中間一致攻撃 [Lucks による攻撃]	—	2^{57} (選択平文数 2^{56}) [—]	2^{112} (選択平文数 2^{56}) [$2^{108.2}$]
Oorshot-Wiener 既知平文攻撃	—	$2^{120-log_2 N}$ 既知平文数 N	—
● Short Cut Method			
差分攻撃	2^{37} (選択平文数 2^{47})	最大差分特性確率 $2^{-162.3}$ 以下*2	(同左)
線形攻撃	2^{42} (選択平文数 2^{43})	最大線形特性確率 $2^{-134.7}$ 以下*3	(同左)
関連鍵攻撃*4	—	—	$2^{56} \sim 2^{72}$ (選択平文 1) (選択鍵ペア 1)

*1 解読に必要となる Triple DES (または DES) の暗号化または復号処理の回数

*2 Triple DES を 48 段の DES とみなし、16 段 DES の最大差分特性確率 $2^{-54.1}$ より求めた上界値

*3 Triple DES を 48 段の DES とみなし、16 段 DES の最大線形特性確率 $2^{-44.9}$ より求めた上界値

*4 関連鍵攻撃は、攻撃が成立する条件が非常に限定されていることから実際の脅威にはならないとみられている

■Short Cut Method に対する安全性　Short Cut Method の代表的なものとして差分攻撃と線形攻撃がある。DES は、差分攻撃によって、2^{47} 個の選択平文によって 2^{37} の計算量をもって解読可能であることが示されている [3]。また、線形攻撃によって、2^{43} 個の既知平文によって 2^{42} の計算量をもって解読可能であることが示されている [10]。従って、Triple DES を 48 段の DES とみなした場合、既に明らかにされている DES の最大差分特性確率 (16 段で $2^{-54.1}$) と最大線形特性確率 (16 段で $2^{-44.9}$) から見積られる Triple DES の最大差分特性確率と最大線形特性確率は十分に小さいことから、攻撃に必要な選択平文数・既知平文数が膨大となるため、ブロック長 64 ビットの下で理論的に作成可能な 2^{64} 個すべての平文/暗号文ペアを利用したとしても、効率的に鍵の候補を絞り込むことができないと考えられる。また、関連鍵攻撃によって、3-key Triple DES が Merkle と Hellman の選択平文攻撃よりも少ない計算量によって解読可能であるとの研究成果が報告されている [7]。具体的には、1 組の選択平文/暗号文ペアとある特定の関係を有する 1 組の鍵ペアを利用することによって、$2^{56} \sim 2^{72}$ 回程度の計算量で解読できることが示されている。しかしこの攻撃法は、2-key Triple DES には適用できないほか、攻撃可能な環境は極め

て限定されているため、実際の脅威となるとの見方は少ないと言える。

■実装攻撃に対する安全性　DES に対する実装攻撃として、ある種の特殊な条件のもとで、異なるメッセージに対する暗号化処理に伴う消費電力差を使った DPA 攻撃を行うことにより、すべての秘密鍵が導出された例が報告されている [18]。この手法を用いて Triple DES のすべての秘密鍵も DPA 攻撃によって導出可能であると思われる (詳細は第 6 章参照)。

また、ある種の特殊な条件のもとで、キャッシュメモリのヒット・ミスヒットの処理時間差を使ったタイミング攻撃を行うことにより、すべての秘密鍵が導出された例も報告されている [19]。

これらの攻撃法は実装法や使用環境に依存した攻撃法であるので、Triple DES のアルゴリズム自体の安全性に致命的な欠点をもたらすものではなく、利用環境下における適切な実装攻撃対策を施せば実用上十分な安全性が確保されると考えられる。したがって、このような実装攻撃に対する脅威がある環境において Triple DES を利用する場合には、これらの実装攻撃に対する防御策を注意深く講じることが望まれる。防御策としては、有意な消費電力量や処理時間差が計測できないようにすることなどが挙げられる。実装攻撃の一般的概要、および対策法の詳細については、第 6 章も参照すること。

3.3.4.5　ソフトウェア実装評価結果

Triple DES のソフトウェア実装評価として、表 3.39、表 3.40 に CRYPTREC2000 での評価結果を示す。これによれば PC 環境 (Pentium III) での Triple DES のデータランダム化部処理速度は最速値で 854 cycles/block を達成する。またその後、2002 年の SCIS2002 において、CRYPTREC2000 における評価とほぼ同様な PC 環境 (Pentium III) で、様々な高速化実装技術を駆使することで、データランダム化部処理速度において最速値 763 cycles/block を達成したとの研究報告が発表されている [2]。

表 3.39: Triple DES のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	44,385 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション		
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	854 / 856	854 / 856
2 回目	854 / 857	854 / 856
3 回目	854 / 856	854 / 857

表 3.40: Triple DES の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	44,679 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション		
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,963 / 1,967	1,971 / 1,975
2 回目	1,967 / 1,971	1,971 / 1,975
3 回目	1,963 / 1,967	1,971 / 1,975

3.3.4.6 ハードウェア実装評価結果

以下のブロック図 (図 3.8,3.9) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.41) を示す。

Key Setup Clock 数	1
Data Randomize Clock 数	48
実装鍵ビット数	128

表 3.41: Triple DES の HW 実装評価結果

また、Triple DES のハードウェア実装評価としては、ASIC による実装結果が報告されている [5, 20]。以下にそれらの評価結果を示す。

ASIC プロセス : Mitsubishi Electric 0.35μm CMOS ASIC Design Library
速度優先実装 : 407.4 Mbps, 148.1 Kgates
(暗号化&復号部:124.9 Kgates/鍵スケジュール部:23.2 Kgates)

ASIC プロセス : 0.18μm CMOS ASIC Design Library
速度優先実装 : 646.5 Mbps, 13.7 Kgates
規模優先実装 : 170.3 Mbps, 5.7 Kgates

ASIC プロセス : 0.13μm CMOS ASIC Design Library
速度優先実装 : 1,066.7 Mbps, 17.0 Kgates
規模優先実装 : 334.2 Mbps, 5.5 Kgates

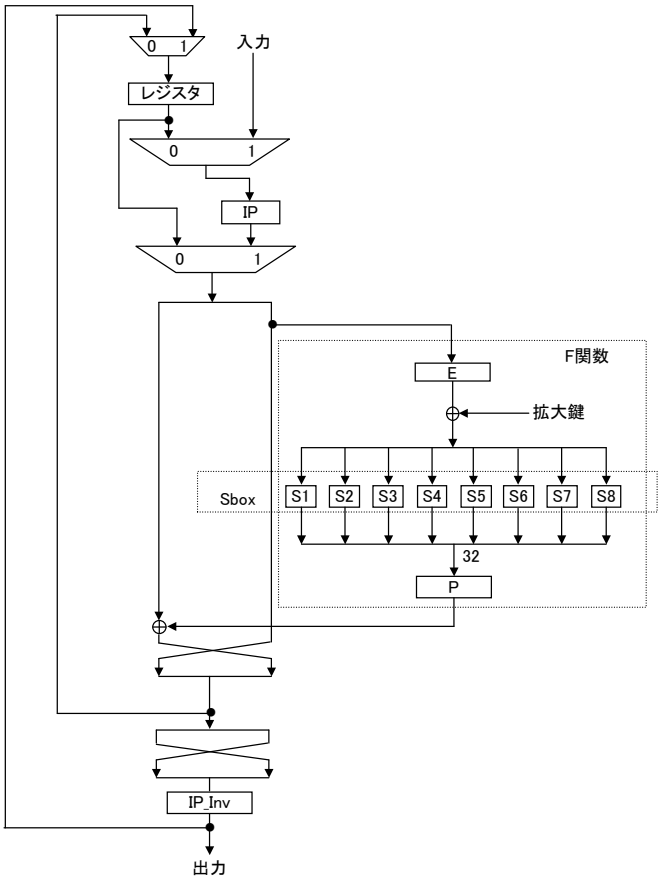

図 3.8: Triple DES 暗号化回路ブロック図

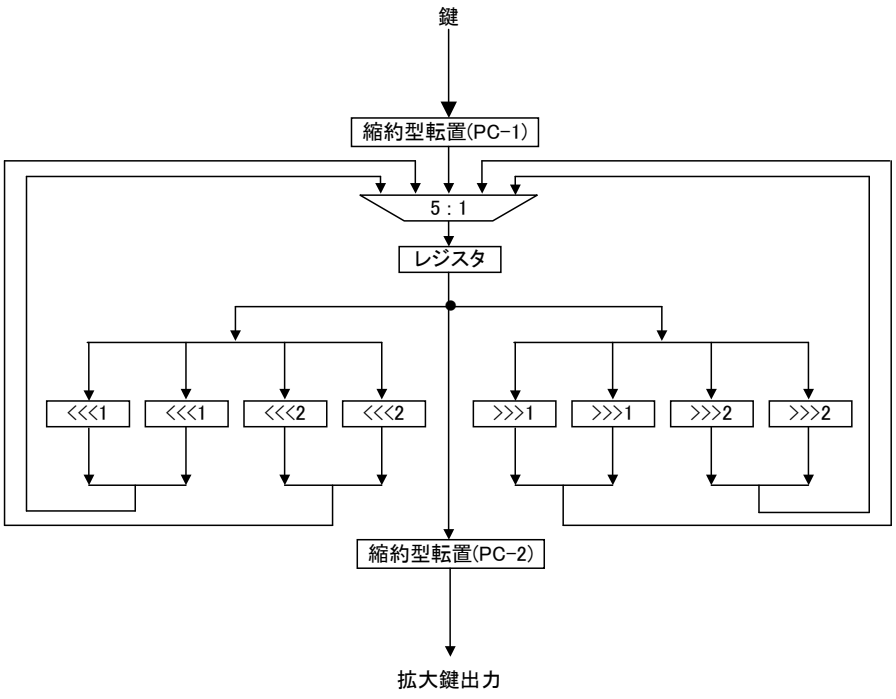

図 3.9: Triple DES 鍵生成回路ブロック図

参考文献

[1] American National Standards Institute, *Triple Data Encryption Algorithm Modes of Operation (X9.52-1998)*, 1998.

[2] 青木和麻呂、「Pentium III 上の Triple DES 実装最適化記」、2002 年暗号と情報セキュリティシンポジウム, SCIS2002, 12C-2, 2002.

[3] E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES," In *Advances in Cryptology -Proceedings of CRYPTO92*, Vol. 740 of *LNCS*, pp.487–496. Springer-Verlag, 1993.

[4] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, Vol. 10, No. 6, pp.74–84, June 1977.

[5] T. Ichikawa, T. Kasuya, and M. Matsui, " Hardware evaluation of the AES finalists," In *The Third AES Candidate Conference*, pp.279–285. the National Institute of Standards and Techonlogy, Gaithersburg, MD, April 13-14 2000.

[6] IETF, *The TLS Protocol Version 1.0*, *RFC2246*, 1999. http://www.ietf.org/ rfc/rfc2246.txt.

[7] J. Kelsey, B. Schneier, and D. Wagner, "Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple DES," In *Advances in Cryptology CRYPTO96*, Vol. 1109 of *LNCS*, pp.237–251. Springer-Verlag, 1996.

[8] K. Kusuda and T. Matsumoto, " *A Strength Evaluation of the Data Encryption Standard*," No. 97-E-5 in IMES Discussion Paper. Institute for Monetary and Economic Studies, Bank of Japan, 1997.

[9] S. Lucks, " Attacking triple DES," In *proceedings of Fast Software Encryption '98*, Vol. 1372 of *LNCS*, pp.239–253, 1998.

[10] M. Matsui, " Linear cryptanalysis method for DES cipher," In *Advances in Cryptology - Proceedings of EUROCRYPT'93*, Vol. 765 of *LNCS*, pp.386–397. Springer-Verlag, 1994.

[11] R. C. Merkle and M. Hellman, "On the security of multiple encryption," *Communications of the ACM*, Vol. 24, No. 7, pp.465–467, 1981.

[12] National Institute of Standards and Technology, *Data Encryption Standard (Federal Information Processing Standards Publication 46-3)*, 1999.

[13] Netscape Communications, *SSL 3.0 SPECIFICATION*, 1996. http://home.netscape.com/eng/ssl3/draft302.txt.

[14] W. Tuchman, " Hellman presents no shortcut solutions to DES," *IEEE Spectrum*, Vol. 16, No. 7, pp.40–41, 1979.

[15] P. C. van Oorschot and M. J. Wiener, "A known plaintext attack on two-key triple encryption," In *Advanced in Cryptology -Proceedings of EUROCRYPT'90*, Vol. 473 of *LNCS*, pp.318–325. Springer-Verlag, 1990.

[16] 谷口、太田、大久保、「Triple DES を巡る最近の標準化動向について」、金融研究, 第 18 巻別冊第 1 号. 日本銀行金融研究所, 1999.

[17] 宇根、太田、「共通鍵暗号を取り巻く現状と課題 —DES から AES へ—」、金融研究, 第 18

巻第 2 号. 日本銀行金融研究所, 1999.

[18] P.Kocher, J.Jaffe, B.Jun, “Differential Power Analysis,” in Proceedings of Advances in Cryptology –CRYPTO’99, Springer-Verlag, 1999, pp.388–397.

[19] 鶴丸、酒井、反町、松井、「64 ビットブロック暗号に対するタイミング攻撃」、2003 年暗号と情報セキュリティシンポジウム SCIS2003, 2D-3, 2003.

[20] 佐藤、森岡、「AES/Camellia/Triple-DES のハードウェア性能比較」、2003 年暗号と情報セキュリティシンポジウム SCIS2003, 12D-1, 2003.

3.3.5 Advanced Encryption Standard (AES)

3.3.5.1 技術概要

AES は基本的に、1998 年に J. Daemen (Proton World International) と V. Rijmen (Katholieke Universiteit Leuven) によって AES (Advanced Encryption Standard) プロジェクトに提案された共通鍵ブロック暗号 Rijndael であり、ブロック長・鍵長ともに 128、192、256 ビットが利用可能である [1]。Rijndael は、AES での公開の議論を経て、2000 年 10 月に NIST (National Institute of Standards and Technology) によって AES winner に選定され [2]、2001 年 11 月に AES (Advanced Encryption Standard) として FIPS PUB 197 (Federal Information Processing Standards Publications 197) に制定された。Rijndael は、ブロック長・鍵長、繰り返し段数の可変なパラメータを持つが、FIPS PUB 197 においては、AES の仕様としてそれらを限定している。

3.3.5.2 技術仕様

AES の主な設計方針は、(1) 既存の攻撃法に対して十分な安全性を確保する、(2) 様々なハードウェアにおいて実装可能とする、(3) 安全性に関する分析が容易になるようにアルゴリズムの構造をシンプルにする、である。AES は SPN 型暗号で、データブロックはラウンド関数内で 8 ビット単位で変換される。アルゴリズムの段数はブロック長と鍵長に依存し、128 ビットブロックの場合は、鍵長が 128、192、256 ビットに対応して、10 段、12 段、14 段となる。ラウンド関数は三種類の変換部によって構成されており、線形変換層 (ビットシフト等)、非線形変換層 (換字変換)、拡大鍵変換層 (拡大鍵との排他的論理和) を用いて変換が行われる。鍵スケジュール部では、ブロック長と同じ長さの拡大鍵が $(r + 1)$ 個 (r は段数) 生成される。鍵スケジュール部の変換には、データランダム化部のビットシフトと換字変換が利用される。

3.3.5.3 その他

Rijndael は同じ設計者を提案者に含む、SHARK[3] および SQUARE[4] という暗号の後継暗号であると考えられる。

標準化関連として、FIPS PUB 197 のほか、ISO/IEC 18033-3 (Committee Draft)、RFC 3268: AES Ciphersuites for TLS (Proposed Standard)、RFC 3394: Advanced Encryption Standard (AES) Key Wrap Algorithm (Informational)、IETF S/MIME (Internet Draft)、IETF IPsec (Internet Draft)、NESSIE、WAP/WTLS 1.0、TV-Anytime Forum Specification S-7 など多数に掲載されている。

3.3.5.4 安全性評価結果

128 ビットブロック暗号の AES の安全性について現在まで報告されてきた公知文献等の主な評価結果をまとめると次のようになる。

- 128、192、256 ビット鍵の仕様どおりの AES を解読可能な攻撃法は発見されていない。
- 128 ビット鍵の場合、10 段のうち 6 段あるいは 7 段まで解読可能な攻撃法が発見されて

いる。

- 192 ビット鍵の場合、12 段のうち 7 段まで解読可能な攻撃法が発見されている。
- 256 ビット鍵の場合、14 段のうち 7 段、8 段あるいは 9 段まで解読可能な攻撃法が発見されている。

以上の結果、NIST は AES の報告書において Rijndael はその安全性において及第点に達している (adequate セキュリティマージンを持つ) と報告し [2]、AES として FIPS PUB 197 に制定した。以下、これらに関してもう少し詳しく述べる。

(1) AES 提案時の提案者による自己評価報告

Rijndael の提案者は AES への提案時に、Rijndael の差分攻撃、線形攻撃、Truncated 差分攻撃、Square 攻撃、補間攻撃、弱鍵、鍵関連攻撃について考察し、すべてのブロック長と鍵長の組合せにおいて、鍵の全数探索よりも効率のよい解読法は存在しないと述べている [1]。具体的には、差分攻撃と線形攻撃に対しては、差分特性確率および線形特性確率において、4 段で確率 2^{-150} を越えるパスは存在しないとし、十分安全であるとしている。また、Truncated differentials については、6 段以上において鍵の全数探索より効率の良い解読法はないと述べている。更に、Square 攻撃 [4] に関しては、4 段、5 段、6 段の Rijndael に対して適用可能であることを示し、7 段以上において鍵の全数探索より効率の良い解読法は見つかっていないと述べている。その他、補間攻撃、弱鍵、鍵関連攻撃などの攻撃法は、Rijndael には適用困難であると示している。

(2) AES 提案後の安全性評価結果

AES に提案後、多くの研究者によって Rijndael の安全性に関する研究報告が行われた。それらのうち主なものを以下に示す。

- Collision attack の適用で、192 ビット鍵および 256 ビット鍵の Rijndael の場合には、2^{32} の選択平文を用いて 7 段まで解読可能であることが報告されている [5]。
- Square 攻撃を 192 ビット鍵および 256 ビット鍵に適用することで、2^{32} の選択平文を用いて 7 段の Rijndael が解読可能であることが報告されている [6]。
- Square 攻撃を改良し、128 ビット鍵の場合は 7 段まで、256 ビット鍵の場合は 8 段まで解読可能な攻撃法が報告されている [7]。ただしこの解読法で必要な選択平文数は、ほぼ全数にあたる $2^{128} - 2^{119}$ となっている。
- 鍵関連攻撃によって 256 ビット鍵の Rijndael が 9 段まで解読可能であると報告されている [7]。

以上のように、現在までフルスペックの Rijndael を解読可能な攻撃法は見つかっていない。NIST は、これらの公開評価報告にもとづき、Rijndael はその安全性において及第点に達している (adequate セキュリティマージンを持つ) と報告している [2]。

さらに、その後も暗号関連学会において、安全性の研究が引き続き行われ、1, 2 の平文で解読できる可能性を示唆する論文 [19] も報告されている。この論文では、AES の他 Camellia に対しても言及しているが、上述の研究報告に比べ、安全側の推定評価であり、現時点では、これら暗号の安全性に問題は無いと判断する。しかしながら、今後の暗号研究の進展には、継続的に注意

を払って行く必要があると考える。

■実装攻撃に対する安全性 AES に対する実装攻撃として、ある種の特殊な条件のもとで、暗号化処理に伴う、消費電力量を使った電力解析攻撃を行うことによりすべての秘密鍵が導出された例 [20] や、処理時間差を使ったタイミング攻撃を行うことによりすべての秘密鍵が導出された例 [21] が報告されている。

また、ある種の特殊な条件のもとで、キャッシュメモリのヒット・ミスヒットの処理時間差を使ったタイミング攻撃を行うことにより、すべての秘密鍵が導出された例も報告されている [22]。

これらの攻撃法は実装法や使用環境に依存した攻撃法であるので、AES のアルゴリズム自体の安全性に致命的な欠点をもたらすものではなく、利用環境下における適切な実装攻撃対策を施せば実用上十分な安全性が確保されると考えられる。したがって、このような実装攻撃に対する脅威がある環境において AES を利用する場合には、これらの実装攻撃に対する防御策を注意深く講じることが望まれる。防御策としては、有意な消費電力量や処理時間差が計測できないようにすることなどが挙げられる。実装攻撃の一般的概要、および対策法の詳細については、第 6 章も参照すること。

3.3.5.5 ソフトウェア実装評価結果

Rijndael のソフトウェア実装評価としては、幾つかの評価環境 (CPU、言語、他) のもとでの実装結果が多数報告されている [2]。なお、以下の評価結果にある鍵セットアップ時間は暗号化または復号時間を含まないことに注意せよ。

表 3.42: 64 ビットプロセッサでの暗号化・復号処理クロック数 [cycles/block]

	F		G		H	I
鍵長	暗号化	復号	暗号化	復号	暗号化	暗号化
128-bit	168	168	125	126	490	293

表 3.43: 64 ビットプロセッサでの鍵セットアップクロック数 [cycles/key]

	F	G
鍵長	暗号化	暗号化
128-bit	239	148

F: Hewlett-Packard PA-RISC, ASM. Source: Ref. [14], Appendix A.

G: Hewlett-Packard IA-64, C. Source: Ref. [14], Appendix A., Ref. [15]

H: Compaq Alpha 21164A 500MHz, C. Source: Ref. [13], Table 1.

I: Compaq Alpha 21264, C. Ref. [16], Table 1.

表 3.44: 32 ビットプロセッサでの暗号化・復号処理クロック数 [cycles/block]

	A	B		C		D		E
鍵長	暗号化	暗号化	復号	暗号化	復号	暗号化	復号	暗号化
128-bit	237	1,276	1,276	805	784	362	358	7,770
192-bit	—	—	—	981	955	428	421	—
256-bit	—	—	—	1,155	1,121	503	492	—

表 3.45: 32 ビットプロセッサでの鍵セットアップクロック数 [cycles/key]

	B		C		D	
鍵長	暗号化	復号	暗号化	復号	暗号化	復号
128-bit	17,742	18,886	1,289	1,724	215	1,334
192-bit	—	—	2,000	2,553	215	1,591
256-bit	—	—	2,591	3,255	288	1,913

A: Intel Pentium II, C. Source: Ref.[10],Table 1.

B: Linux/GCC-2.7.2.2/Pentium 133MHz MMX, C. Source: Ref. [11], Table 3

C: Intel Pentium III 600MHz, C.Ref. [8], 5.1, Table 6 (128blocks)

D: Intel Pentium II/III, C. Source: Ref. [12], Table 1.

E: Ultra SPARC-I, W/JDK1.2, JIT, Java. Ref. [13], Table 2.

表 3.46: 8 ビットプロセッサでの処理クロック数

	J		K
鍵長	鍵セットアップ cycles/key	暗号化 cycles/block	鍵セットアップ + 暗号化 cycles
128-bit	10,318	9,464	25,494

J: Motorola 6805 CPU Core, C. Ref. [17], Table 3.

K: Z80 CPU+coprocessor. Ref. [18], Table 8.

3.3.5.6　ハードウェア実装評価結果

以下のブロック図 (図 3.10,3.11) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.47) を示す。

Key Setup Clock 数	10
Data Randomize Clock 数	11
実装鍵ビット数	128

表 3.47: AES の HW 実装評価結果

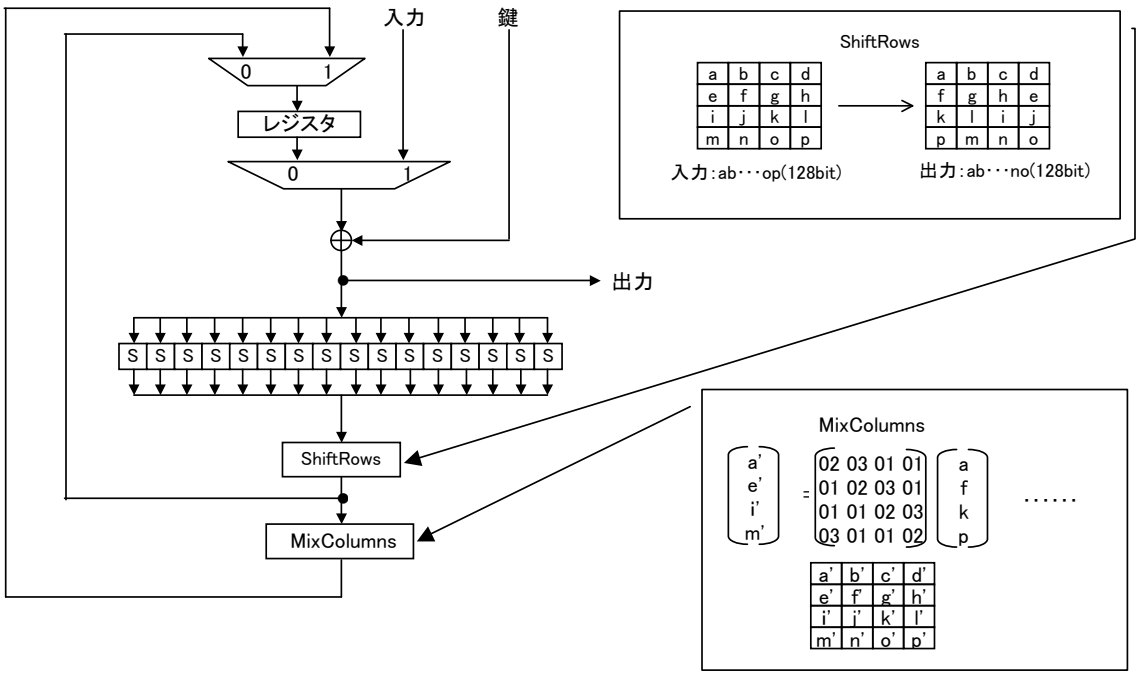

図 3.10: AES 暗号化回路ブロック図

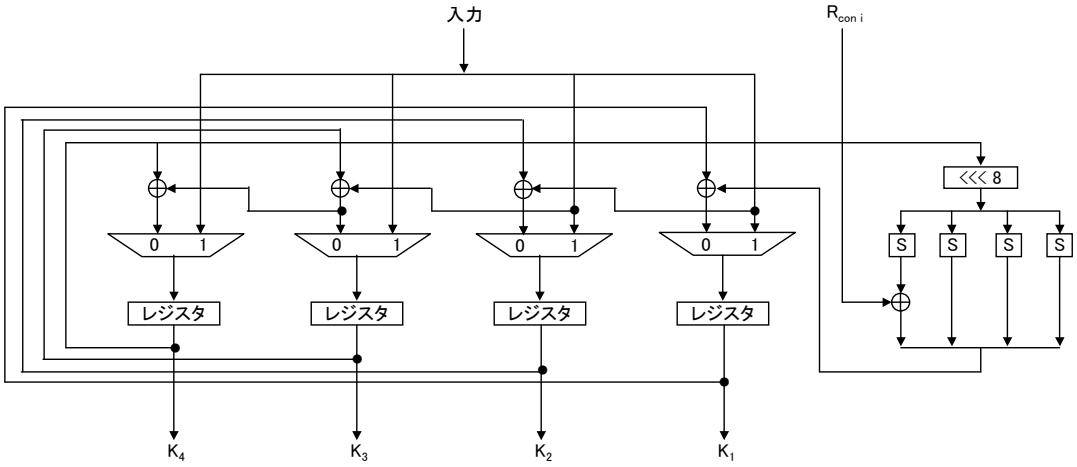

図 3.11: AES 鍵生成回路ブロック図

また、Rijndael のハードウェア実装評価としては、ASIC, FPGA ともに多数の実装例が報告されている [2, 9, 23, 25]。以下に、その一例を示す。

ASIC プロセス　：　Mitsubishi Electric 0.35μm CMOS ASIC Design Library
速度優先実装　：　1,950.03 Mbps, 612.8 Kgates
(暗号化&復号部:518.5 Kgates/鍵スケジュール部:93.7 Kgates)

ASIC プロセス　：　0.18μm CMOS ASIC Design Library
速度優先実装　：　2,245.6 Mbps, 33.9 Kgates
規模優先実装　：　235.2 Mbps, 5.3 Kgates

ASIC プロセス　：　0.13μm CMOS ASIC Design Library
速度優先実装　：　3,459.5 Mbps, 36.9 Kgates
規模優先実装　：　311.1 Mbps, 5.4 Kgates

FPGA プロセス　：　Xilinx Vertex 3200E
速度優先実装　：　591.7 Mbps, 31,239 slices
規模優先実装　：　512.6 Mbps, 4,052 slices

また、興味深い実装例として、AES とほぼ同じ構成要素で作られている暗号である Camellia との共有ハードウェア・アーキテクチャが発表されている [24]。この実装例による処理性能は以下のとおりである。

ASIC プロセス　：　0.13μm CMOS ASIC Design Library
速度優先実装　：　(AES+Camellia) 24.7 Kgates
(AES) 794.1 Mbps, (Camellia) 1,118.9 Mbps
規模優先実装　：　(AES+Camellia) 16.3 Kgates
(AES) 458.8 Mbps, (Camellia) 646.6 Mbps

参考文献

[1] J. Daemen and V. Rijmen, AES proposal: Rijndael, AES algorithm submission, September 3, 1999, http://csrc.nist.gov/encryption/aes/rijndael/Rijndael-ammended.pdf.

[2] J. Nechvatal, et al., Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000, http://csrc.nist.gov/encryption/aes/round2/r2report.pdf.

[3] V. Rijmen, et al., “The Cipher SHARK,” 3rd Fast Software Encryption, LNCS 1039, pp.99–112, Springer-Verlag, 1996.

[4] J. Daemen, L. Knudsen, and V. Rijmen, “The Block Cipher Square,” 4th Fast Software Encryption, FSE97, LNCS 1267, pp.28–40, Springer-Verlag, 1997.

[5] H. Gilbert and M. Miner, “A collision attack on 7 rounds of Rijndael,” in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, April 13-14, 2000, pp.230–241.

[6] S. Lucks, “Attacking Seven Rounds of Rijndael Under 192-bit and 256-bit Keys,” in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14, 2000, pp.215–229.

[7] N. Ferguson, et al., “Improved Cryptanalysis of Rijndael,” in the preproceedings of the Fast Software Encryption Workshop 2000, April 10-12, 2000.

[8] L. Bassham, “Efficiency Testing of ANSI C implementations of Round 2 Candidate Algorithms for the Advanced Encryption Standard,” in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14, 2000, pp.136–148.

[9] T. Ichikawa, T. Kasuya, and M. Matsui, “Hardware Evaluation of the AES Finalists,” in The Third AES Candidate Conference, printed by the National Institute of Standards and Techonlogy, Gaithersburg, MD, April 13-14, 2000, pp.279–285.

[10] K. Aoki and H. Lipmaa, “Fast Implementations of AES Candidates,” in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14, 2000, pp.106–120.

[11] E. Biham, “A Note on Comparing the AES Candidates,” in The Second AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, March 22-23, 1999, pp.85–92.

[12] B. Gladman, AES Second Round Implementation Experience, AES Round2 public comment, May 15, 2000

[13] O. Baudron, et al., “Report on the AES Candidates,” in The Second AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, March 22-23, 1999, pp.53–67.

[14] J. Worley, et al., “AES Finalists on PA-RISC and IA-64: Implementations & Performance,” in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburgs, MD, April 13-14, 2000, pp.57–74.

[15] J. Worley, E-mail comments, AES Round 2 public comment, May 15, 2000, available at AES home page.

[16] R. Weiss and N. Binkert, “A comparison of AES Candidate on the Alpha 21264,” in The Third AES candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14, 2000, pp.75–81.

[17] G. Keating, “Performance analysis of AES candidates on the 6805 CPU,” AES Round 2 public comment, April 15, 1999, available at AES home page.

[18] F. Sano, et al., “Performance Evaluation of AES Finalists on the High-End Smart Card,” in The Third AES Candidate Conference, printed by the National Institute of Standards and technology, Gaithersburg, MD, April 13-14, 2000, pp.82–89

[19] N. T. Courtois and J. Pierprzyk, “Cryptanalysis of Block Ciphers with Overdefined Systems of

Equations,” http://eprint.iacr.org/2002/044/.

[20] S. Chari, C. Jutla, J. R. Rao, P. Rohatgi, “A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards,” in The Second AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, March 22-23, 1999.

[21] F. Koeune, J. J. Quisquater, “A timing attack against Rijndael,” UCL Crypto Group Technical Report CG-1999/1, 1999.

[22] 角尾、久保、茂、辻原、宮内、「S-box におけるキャッシュ遅延を利用した AES へのタイミング攻撃」、SCIS 2003, 3D-1, 2003.

[23] 佐藤、森岡、「AES/Camellia/Triple-DES のハードウェア性能比較」、SCIS 2003, 12D-1, 2003.

[24] 佐藤、森岡、「AES と Camellia の共有ハードウェア・アーキテクチャ」、SCIS 2003, 12D-2, 2003.

[25] 反町、市川、粕谷、「FPGA を用いたブロック暗号のハードウェア実装評価」、SCIS 2003, 12D-3, 2003.

3.3.6 Camellia

3.3.6.1 技術概要

Camellia は日本電信電話株式会社と三菱電機株式会社によって共同開発された 128 ビットブロック長の共通鍵暗号であり、2000 年に学会発表された [3]。鍵長は 128、192、256 ビットの 3 通りである。その基本構造は 18 段 (128 ビット鍵長) もしくは 24 段 (192 ビット鍵長、256 ビット鍵長) の Feistel 型構造であり、6 段ごとに FL/FL^{-1} 関数が挿入されて、規則的な繰り返し構造とならないように配慮されている。安全性と実装性とのバランスを重視した設計であり、ソフトウェアとハードウェアの両面での効率的な実装を目指している。とくにハードウェア実装ではゲート数当りの暗号化/復号処理速度およびゲート数で現時点の世界最小クラスに属する。また、鍵スケジュール部も簡単な構造であるので鍵変更速度も高速であるという特長も有している。想定されるアプリケーションとしては、高速暗号通信からローコスト IC カードまでの幅広い分野が考えられる。

3.3.6.2 技術仕様

ラウンド関数の構成要素の基本部分は S-box と排他的論理和から、また関数の構成要素は論理和、論理積、排他的論理和およびローテーションからなっている。算術演算は一切用いていない。これにより、長いクリティカルパスを排除し回路規模の小型化を実現している。また拡大鍵の生成関数の設計においては、on-the-fly 鍵生成が可能な設計を用いている。

■データランダム化部

128 ビット鍵の場合　データランダム化部は 18 段の Feistel 型構造と FL/FL^{-1} 関数により構成されている。Feistel 型構造における 64 ビット出力の F 関数は、同じく 64 ビット出力の S 関数と P 関数との合成であり、S 関数では 4 種類の 8 ビット入出力の S-box からなっている。P 関数は 8 ビットの線形写像を 8 個並列に実行したものである。FL/FL^{-1} 関数は 2 層あり、第 6 段と第 12 段の直後に挿入されている。64 ビット出力の FL/FL^{-1} 関数では論理和、論理積、1 ビット巡回シフト、排他的論理和が用いられている。FL/FL^{-1} 関数における MISTY と Camellia の違いは 1 ビット巡回シフトの導入である。第 1 段の直前と最終段の直後において、初期および最終排他的論理和が行われている。鍵スケジュール部では 128 ビットの秘密鍵 K から、64 ビットの拡大鍵を 26 個生成する。(拡大鍵生成手順の一部はデータランダム化部と同一)

データランダム化部では、平文と 2 つの拡大鍵を連接したものとの排他的論理和が計算され、それを 2 等分する。そして以下の演算を $r = 1 \sim 18$ まで実行する (但し、$r = 6, 12$ は除外)。

$$L_r = R_{r-1} \oplus F(L_{r-1}, k_r)$$
$$R_r = L_{r-1}$$

$r = 6, 12$ の場合は FL/FL^{-1} 関数が一部用いられる。これは規則的な繰り返し構造とならないようにするために挿入されたものである。最後に 2 個の拡大鍵との排他的論理和が行われる。

192 ビット鍵と 256 ビット鍵の場合 データランダム化部は 24 段の Feistel 型構造と FL/FL^{-1} 関数とからなる。FL/FL^{-1} 関数は 3 層あり第 6 段、12 段、18 段の直後に挿入される。第 1 段の直前と最終段の直後において拡大鍵との排他的論理和演算がなされる。

■復号関数 Camellia の復号は、拡大鍵の順番を逆順にすれば暗号化と同様の処理で行われる。

■鍵スケジュール部 鍵スケジュール部では 2 つの 128 ビットデータおよび 4 つの 64 ビットデータを用いる。これらの値を用いて 2 つの 128 ビットデータ K_a と K_b を生成する。但し、K_b の方は 192 あるいは 256 ビット鍵の場合のみ使用する。拡大鍵は中間的な鍵を循環シフトさせた値の左あるいは右半分の値になっている。鍵スケジュール部は簡単な構造を有し、暗号化処理の一部分を共用している。また動的な拡大鍵生成 (on-the-fly 鍵生成) が可能で、そのとき暗号化/復号を問わずほぼ同じ効率で拡大鍵は生成される。拡大鍵生成のためのメモリ使用量も小さい。(128 ビット鍵で約 32 bytes の RAM、192 ビット鍵・256 ビット鍵で約 64 bytes の RAM)

■安全性設計 主要な攻撃法として考えられている差分攻撃、線形攻撃、丸め差分攻撃に対して十分な耐性を持つように、つまり最大差分特性確率・最大線形特性確率の上界値の見積もりから本暗号の安全性設計を行っている。その他、高階差分攻撃、補間攻撃、関連鍵攻撃、不能差分攻撃、スライド攻撃などに対する耐性を設計段階で考慮している。

3.3.6.3 その他

Camellia [3] の開発設計において、いくつかの暗号技術が NTT の暗号技術 E2 [1] と三菱電機の暗号技術 MISTY [2] を土台にして開発設計されている。例えば、ラウンド関数 (F 関数) や線形変換関数 (P 関数) の設計指針は E2 の F 関数/P 関数の設計指針を踏襲している。また FL/FL^{-1} 関数の設計指針は MISTY の FL 関数の設計指針を踏襲している。主たる暗号設計の変更点は PC 上、IC カード (Smart Card)、ハードウェアでの実装性能の向上にあると考えられる。

標準化関連として、ISO/IEC 18033-3 (Committee Draft)、NESSIE、IETF RFC (Internet Draft)、IETF TLS (Internet Draft)、IETF S/MIME (Internet Draft)、TV-Anytime Forum Specification S-7 に掲載されている。

3.3.6.4 安全性評価結果

詳細評価結果によれば、本暗号の安全性に重大な問題点は見出されていない。特に差分攻撃や線形攻撃に対しては、7、8 段程度が実際の攻撃可能段数になるであろうと考えられ、実用的な意味で安全性を満たしていると判断できる (なお、truncated 差分経路探索を行った結果、FL/FL^{-1} 関数を除いた 7 段変形 Camellia に対して攻撃に有効な特性が見出されている [4])。

また、その安全性に対する考察も引き続きなされており、評価の精密化が進展しさらに攻撃法の改良も進み 10 段程度の攻撃可能段数が得られている (例えば、高階差分攻撃と選択暗号文攻撃の組み合わせで 11 段 Camellia の暗号化が解読可能 [10]) が、その安全性に特段の問題点が生じている訳ではない [9, 6, 7, 8, 13, 11, 10]。詳細評価結果の概要は以下のとおりである。

- FL/FL^{-1} 関数を除いた変形 Camellia の 5 段において、バイト多項式による解析によって選択平文 2 文で 5 段目の拡大鍵 1 byte を 1 つに絞り込めることがある。
- 全単射なラウンド関数を用いていることから、FL/FL^{-1} 関数を除いた変形 Camellia 6 段で鍵の全数探索よりも少ない計算量で鍵推定が可能であろう。
- 2 つの差分を利用するブーメラン攻撃を適用することにより、FL/FL^{-1} 関数を除いた変形 Camellia 8 段が、鍵の全数探索より少ない計算量で鍵を推定することが可能であろう。また Camellia にとってはブーメラン攻撃が最も有効な解析手法であると考えられる。
- 鍵スケジュール部の特性として秘密鍵 5 bytes と中間鍵 6 bytes から不明な秘密鍵 1 byte を計算できる場合が存在した。
- 差分攻撃や線形攻撃の他、丸め差分攻撃・丸め線形攻撃、高階差分攻撃、不能差分攻撃、補間攻撃、線形和攻撃、スライド攻撃などに関しても、安全性に関する問題は見つかっていないと判断できる。

■実装攻撃に対する安全性 Camellia に対する実装攻撃として、ある種の特殊な条件のもとで、キャッシュメモリのヒット・ミスヒットの処理時間差を使ったタイミング攻撃を行うことにより、すべての秘密鍵が導出された例が報告されている [12]。

本攻撃法は実装法や使用環境に依存した攻撃法であるので、Camellia のアルゴリズム自体の安全性に致命的な欠点をもたらすものではなく、利用環境下における適切な実装攻撃対策を施せば実用上十分な安全性が確保されると考えられる。したがって、この種のタイミング攻撃に対する脅威がある環境において Camellia を利用する場合には、このような実装攻撃に対する防御策を注意深く講じることが望まれる。防御策としては、有意な処理時間差が計測できないようにすることなどが挙げられる。実装攻撃の一般的概要、および対策法の詳細については、第 6 章も参照すること。

3.3.6.5 ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.48、表 3.49 の通りである。

また、応募者からは以下の自己評価が報告されている。

プラットフォーム	:	Pentium III (1GHz), 512MB
OS およびコンパイラ	:	Windows 2000, IBM Java Compiler 1.2.2, Java VM 1.2.2
使用言語	:	Java
鍵スケジュール	:	9,091 cycles/key
暗号化	:	793 cycles/block

表 3.48: Camellia のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	29,285 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/G6 /ML /O2 /Ob2 /Og /Oi /Ot /Ox /Oy /Gr /I	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	326 / 327	326 / 328
2 回目	326 / 327	326 / 327
3 回目	326 / 327	326 / 327

UltraSPARC IIi (400MHz)		
言語	アセンブラ	
プログラムサイズ	15,240 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-fast - xtarget=ultra -xarch=v9a	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	355 / 360	355 / 357
2 回目	355 / 358	355 / 358
3 回目	355 / 357	355 / 357

Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	31,552 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-O -arch ev6	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	282 / 288	282 / 288
2 回目	282 / 289	282 / 288
3 回目	282 / 288	282 / 289

表 3.49: Camellia の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	20,110 bytes (暗号化/鍵スケジュール含む) 20,236 bytes (復号/鍵スケジュール含む)	
コンパイラオプション	/G6 /ML /O2 /Ob2 /Og /Oi /Ot /Ox /Oy /Gr /I	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	467 / 487	474 / 493
2 回目	467 / 487	474 / 494
3 回目	467 / 487	474 / 493

UltraSPARC IIi (400MHz)		
言語	アセンブラ	
プログラムサイズ	23,992 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-fast -xcrossfile -xtarget=ultra -xarch=v9a	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	403 / 408	403 / 407
2 回目	403 / 407	403 / 407
3 回目	403 / 408	403 / 408

Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	25,792 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-O -arch ev6	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	448 / 454	435 / 439
2 回目	448 / 454	435 / 439
3 回目	448 / 455	435 / 439

■IC カード実装　Z80 による IC カード実装評価を実施した。表 3.50 は、128 ビット鍵を利用したときの鍵スケジュール部 + データランダム化部処理時間測定結果である。

表 3.50: Camellia の Z80 上での鍵スケジュール部 + データランダム化部処理時間測定結果

	ROM [bytes]	RAM [bytes]	Stack [bytes]	処理時間 [states]
暗号化	1,023	48	12	35,951
復号	1,042	48	12	37,553
暗復共用	1,268	—	—	—

また、提案者よりのデータを以下に載せる。なお、128 ビット鍵を利用している。

プロセッサ	暗号化 [cycles/block]	鍵スケジュール [cycles/key]	ROM [bytes]	RAM [bytes]
8051	10,217 (鍵スケジュール込)		990	32
Z80	28,382	5,146	1,698	62
H8/3113	4,100	2,380	—	208
MC68HC705B16	9,900	7,500	—	208
MC68HC908AB32	8,430	5,679	—	208
M32Rx/D	1,236	642	8,684	44

3.3.6.6　ハードウェア実装評価結果

以下のブロック図 (図 3.12,3.14,3.15,3.13) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.51) を示す。

Key Setup Clock 数	1
Data Randomize Clock 数	20
実装鍵ビット数	128

表 3.51: Camellia の HW 実装評価結果

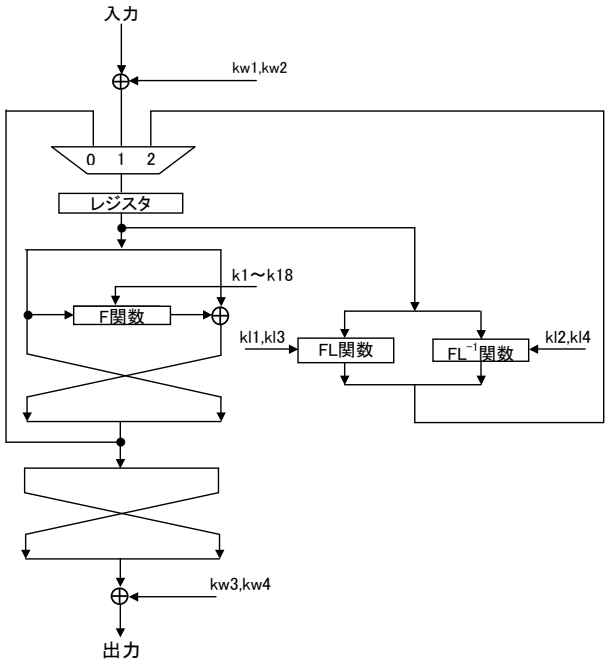

図 3.12: Camellia 暗号化回路ブロック図

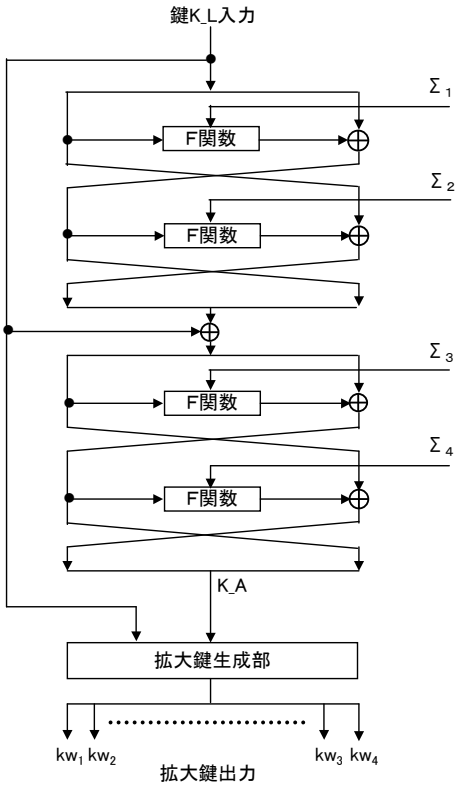

図 3.13: Camellia 鍵生成部回路ブロック図

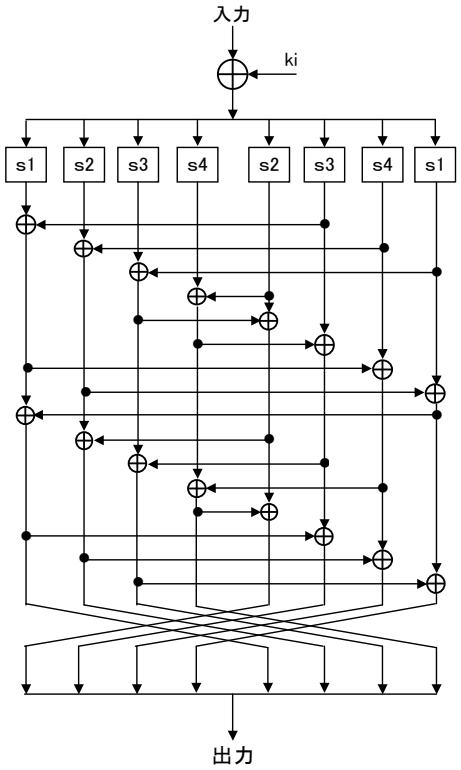

図 3.14: F 関数内部ブロック図

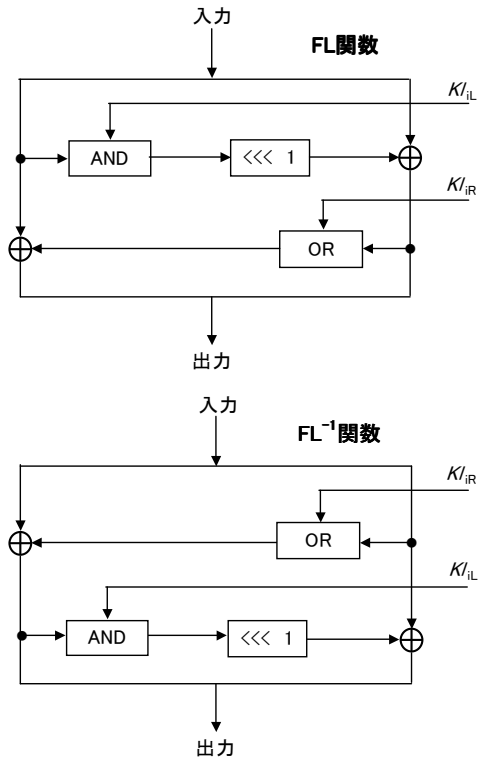

図 3.15: FL・FL^{-1} 関数内部ブロック図

また、応募者からは、ASIC および FPGA 実装に関して以下の自己評価が報告されている。なお、処理回路には暗号化・復号処理部および鍵スケジュール部 (128 ビット鍵) のすべてが含まれる。

ASIC プロセス　：　Mitsubishi Electric 0.18μm CMOS ASIC Design Library
速度優先実装　：　3,200 Mbps, 355.1 Kgates
規模優先実装　：　177.7 Mbps, 8.1 Kgates

FPGA デバイス　：　Xilinx XC4000XL
規模優先実装　：　77.3 Mbps, 1,296 CLBs

FPGA デバイス　：　Xilinx VertexE
速度優先実装　：　401.9 Mbps, 9,426 slices
規模優先実装　：　227.4 Mbps, 1,780 slices
パイプライン実装　：　6,750.0 Mbps, 9,692 slices

最近、Camellia に関する実装技術に関する検討がなされていて、回路規模および処理性能での改善が見られている [5, 14, 16]。

ASIC プロセス　：　0.18μm CMOS ASIC Design Library
速度優先実装　：　1.422.2 Mbps, 31.1 Kgates
規模優先実装　：　204.6 Mbps, 6.3 Kgates

ASIC プロセス　：　0.13μm CMOS ASIC Design Library
速度優先実装　：　2,154.9 Mbps, 29.8 Kgates
規模優先実装　：　325.8 Mbps, 6.5 Kgates

FPGA プロセス　：　Xilinx Vertex 3200E
速度優先実装　：　369.0 Mbps, 8,957 slices
規模優先実装　：　223.7 Mbps, 1,678 slices

また、興味深い実装例として、Camellia とほぼ同じ構成要素で作られている暗号である AES との共有ハードウェア・アーキテクチャが発表されている [15]。この実装例による処理性能は以下のとおりである。

ASIC プロセス　：　0.13μm CMOS ASIC Design Library
速度優先実装　：　(AES+Camellia) 24.7 Kgates
(Camellia) 1,118.9 Mbps, (AES) 794.1 Mbps
規模優先実装　：　(AES+Camellia) 16.3 Kgates
(Camellia) 646.6 Mbps, (AES) 458.8 Mbps

参考文献

[1] M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta, and T. Matsumoto, “A New 128-bit Block Cipher E2,” Technical Report ISEC98-12, IEICE, 1998

[2] M. Matsui, “New Block Encryption Algorithm MISTY,” In E. Biham, editor, Fast Software Encryption — 4th International Workshop, FSE97, Vol.1267, LNCS, pp.54–68, 1997

[3] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakjima, and T. Tokita, “Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms,” Seventh Annual Workshop on Selected Areas in Cryptography, SAC2000, pp.41–54, 2000 (日本語版は、「128 ビットブロック暗号 Camellia」、信学技報 ISEC2000-6, 2000 年 5 月)

[4] 渋谷、下山、辻井、「Byte-Oriented な暗号に対する Truncated Linear Attack」、SCIS2001, pp.591–596, Jan. 2001

[5] 佐藤、森岡、張、「128 ビットブロック暗号 Camellia の小型ハードウエア・アークテクチャ」、SCIS2002, pp.595–598, Jan. 2002

[6] Y. Yeom, S. Park, and I. Kim, “On the Security of CAMELLIA agaist the Square Attaack,” FSE2002, Feb. 2002

[7] T. Shirai, S. Kanamaru, and G. Abe, “Improved Upper Bound of Differential and Linear Characteristic Probability for Camellia,” FSE2002, Feb. 2002

[8] 武田、金子、「Camellia の制御型高階差分攻撃に関する一考察」、SCIS2002, pp.915–919, Jan. 2002

[9] T. Kawabata and T. Kaneko, “A Study on Higher Order Differential Attack of Camellia,” 2nd NESSIE workshop, Sept. 2001

[10] 秦野、関根、金子、「Camellia の高階差分攻撃 (II)」、信学技報, ISEC2002-2, pp.5–12, May 2002

[11] T. Shirai, “Differential, Linear, Boomerang and Rectangle Cryptanalysis of Reduced-Round Camellia,” 3rd NESSIE workshop, Nov.2002

[12] 角尾、洲崎、齋藤、川端、宮内、「S-box におけるキャッシュ遅延を利用した Camellia へのタイミング攻撃」、SCIS2003, pp.179–184, Jan. 2003

[13] Y. Yeom, S. Park and I. Kim, “A Study of Integral Type Cryptanalysis on Camellia,” SCIS2003, pp453–456, Jan. 2003

[14] 佐藤、森岡、「AES/Camellia/Triple-DES のハードウェア性能比較」、SCIS 2003, 12D-1, 2003.

[15] 佐藤、森岡、「AES と Camellia の共有ハードウェア・アーキテクチャ」、SCIS 2003, 12D-2, 2003.

[16] 反町、市川、粕谷、「FPGA を用いたブロック暗号のハードウェア実装評価」、SCIS 2003, 12D-3, 2003.

3.3.7　CIPHERUNICORN-A

3.3.7.1　技術概要

CIPHERUNICORN-A は、2000 年に日本電気株式会社 (NEC) が開発したブロック長 128 ビット、鍵長 128、192、256 ビットの 128 ビットブロック暗号 [1] であり、NEC より提案された。この暗号の基本構造は 16 段の Feistel 型暗号である。

最大の特徴は、暗号の基本となるラウンド関数での拡大鍵探索を難しくすることによって安全性を高めることを意図して、本流部 (main stream) と一時鍵生成部 (temporary key generation) とで構成される極めて複雑なラウンド関数を利用している点である。また、ラウンド関数をブラックボックスとみなし、設計者が定めた初等統計評価を行う暗号強度評価支援システム [2] によって有意な相関関係が見出せないように、ラウンド関数を設計することを主要な設計方針としており、この点が最近の多くの主要な暗号の設計方針と大きく異なる。

応募者によれば、ラウンド関数における初等統計評価においても、すべての評価項目でデータ攪拌の偏りは検出されなかったとしている。実装面では、32 ビットプロセッサ上でより高速に処理できるように設計したと述べている。

3.3.7.2　技術仕様

ブロック長 128 ビット、鍵長 128、192、256 ビット、16 段 Feistel 型構造を採用した 128 ビットブロック暗号であり、AES と同じインタフェースを有する。鍵スケジュール部では、秘密鍵を攪拌しながら、2,304 ビット分の拡大鍵 (32 ビット拡大鍵 72 個) を生成する。

■ラウンド関数 (データランダム化部)

- 4 個の 32 ビット拡大鍵 (関数鍵とシード鍵各 2 個) を用いた 64 ビット入出力関数であり、4 個の S-box (T 関数)、32 ビット算術加算、32 ビット定数算術乗算およびローテーション (A3 関数) の組合せにより構成される。
- 全単射関数ではない。
- 64 ビットの入力データは本流部と一時鍵生成部に分岐し、関数鍵は本流部に、シード鍵は一時鍵生成部にそれぞれ入力される。
- 一時鍵生成部では入力データとシード鍵から一時鍵が生成される。
- 生成された一時鍵は本流部に挿入され、最終的に 64 ビットの出力データが得られる。また、本流部の構成の一部は一時鍵の値によって変化するデータ依存関数となっている。

■鍵スケジュール部

- MT 関数をラウンド関数とする拡張 Feistel 型構造をしており、秘密鍵を攪拌しながら、各 MT 関数から 32 ビットの中間鍵を出力する。

- MT 関数は、ラウンド関数と同じ T0 関数および 32 ビット定数算術乗算の組合せにより構成される。
- 72 個の中間鍵を生成した後、その順番を入れなおして各段における拡大鍵とする。

■設計方針　差分攻撃や線形攻撃はラウンド関数でのデータ攪拌の偏りを利用して鍵情報を推定することから、ラウンド関数においてデータ攪拌の偏りが検出できない構造にすることを CIPHERUNICORN-A の本質的な設計方針としている。そこで、ラウンド関数をブラックボックスとみなして初等統計評価を行う暗号強度評価支援システムを利用して、以下の条件を満たすようにラウンド関数の設計を行っている。

- 高い確率で成立する入力ビットと出力ビットの関係が存在しない
- 高い確率で成立する出力ビット間の関係が存在しない
- 高い確率で成立する入力ビットの変化と出力ビットの変化の関係が存在しない
- 高い確率で成立する鍵ビットの変化と出力ビットの変化の関係が存在しない
- 高い確率で 0 あるいは 1 となる出力ビットが存在しない

3.3.7.3　その他

暗号強度評価支援システムによって同じように設計された暗号として、64 ビットブロック暗号である CIPHERUNICORN-E がある。

3.3.7.4　安全性評価結果

■総評　CIPHERUNICORN-A のラウンド関数の構成は非常に複雑であり、差分攻撃や線形攻撃をはじめとする、理論的な解読技術に対する安全性を正確に評価・解析することは困難である。このため、CRYPTREC Report 2000 での継続的な評価が必要との総合評価を受け、CRYPTREC Report 2001 では CIPHERUNICORN-A に対する 3 段消去攻撃を想定し、差分攻撃や線形攻撃に対して 13 段で十分な安全性を有しているかという観点から安全性評価を継続的に実施した。

CRYPTREC Report 2000 では、おおむね適切な考慮に基づいてラウンド関数の構成を簡略化した mF 関数を利用したモデルでは、少なくとも 15 段以上で最大差分特性確率の上界が、また 14 段以上で最大線形特性確率の上界がそれぞれ 2^{-128} を下回ることが示されている。さらに、CRYPTREC Report 2001 では、応募者ならびに 4 人 (チーム) による評価者が、各々独自に適切と考慮する手法に基づいてラウンド関数および暗号全体の評価を実施した結果、一部の評価を除き、いずれも 13 段における最大差分特性確率の上界は 2^{-100} 以下、最大線形特性確率の上界は 2^{-128} 前後と見積もられている。これらの評価結果はいずれも CIPHERUNICORN-A のラウンド関数そのものではなく、何らかの近似を施していたラウンド関数に基づいて算出されたものである。しかし、多数の評価者が異なる手法による近似を利用していながらほぼ同じ安全性評価結果が得られたことから、CIPHERUNICORN-A の差分攻撃や線形攻撃に対する安全性は、少なくとも今回見積もられた評価結果と同程度以上であると期待される。したがって、3 段消去攻撃を想

定した場合の差分攻撃や線形攻撃に対して、学術的に攻撃不可能であるとまでは証明されないものの、現実にはほぼ不可能であろうと推定される。

また、上記以外の解読法については、CRYPTREC Report 2000 で示すように現時点までに問題となるような点は特に発見されていない。

加えて、安全性に関する新しい指摘として、すべての拡大鍵の値が (秘密鍵の鍵長に関わりなく) 秘密鍵の上位 32 ビットと同一になるという、非自明と考えられる弱鍵が少なくとも一つ存在することが示された。もっとも、現時点では、2^{128} 個の秘密鍵 (128 ビット秘密鍵の場合) のうちの一つが弱鍵として指摘されているだけであるので、この指摘だけで安全性に重大な問題が生じたということではない。

2002 年度は応募者らによって新たな安全性評価結果が公開された [3]。文献 [3] では、これまで評価に用いていたラウンド関数を簡略したモデル (mF 関数) に対して、定数乗算部分を近似しないモデル (mF' 関数) を用いており、13 段の差分特性確率および線形特性確率の上界が 2^{-128} 以下となることから 3 段消去攻撃に対して 16 段の CIPERUNICORN-A が安全であると示していた。ただし、差分特性確率の評価に関しては予測値が含まれており、応募者はこの予測値の正当性に関して調査を継続する必要があると指摘していた。

またその後、上記予測値を計算機実験によって求められた計算値に更新した安全性評価結果が公開された [6]。文献 [6] では、計算機を用いた差分特性確率の上界も 13 段において 2^{-128} 以下になると示している。これらの評価結果は、これまでの評価と同様に実際のラウンド関数ではなく簡易モデル (mF' 関数) を用いた評価である。

以上の結論を総合すると、今までのところ、CIPHERUNICORN-A の安全性について、重大な問題点は見つかっていない。

■初等統計量評価　ラウンド関数に対する初等統計量評価のすべての項目について良好な結果を得ているなど、乱数性に関してはおおむね良好と判断される。ただし、データ攪拌偏りが検出できないようにラウンド関数を設計したとしているが、このように設計されたラウンド関数がランダム関数とほぼ同じ特性をもつことを意味しているわけではない。例えば、自己評価書では本流部、一時鍵生成部のどちらか一方でも十分な攪拌が行われていると述べているが、入力データや鍵の値によっては高い確率で複数個の T 関数の効果が打ち消しあい、どちらか一方だけでは十分な攪拌が行われていない場合があるとの指摘もある。

■理論的解読法ごとの安全性評価

差分攻撃に対する安全性:　ラウンド関数の構成が複雑であり、直接的に評価することが困難な場合、適切な考慮に基づいてラウンド関数を簡略化した暗号モデルを考え、そのモデル上での安全性を議論することがある。これは、実際の暗号が適切な考慮に基づく簡略化モデルでの安全性と同程度以上の安全性を有していると一般に期待されるためである。

CRYPTREC Report 2000 では、(1) 算術加算を排他的論理和に置換、(2) 定数乗算は 32 ビットデータの上位 1 バイトへ入力ビットを集約する処理に置換、(3)A3 関数は truncated vector 単位

でのローテーション処理に置換、などによって簡略化した mF 関数を利用したモデルで安全性の評価を行った。その結果、少なくとも 15 段以上で最大差分特性確率の上界が 2^{-128} を下回ることが示されている。

CRYPTERC Report 2001 では別の観点による近似手法に基づく評価を以下のように実施した。

評価者 1: mF 関数を利用したモデルでの安全性を再評価し、その結果、定数乗算の近似処理が不完全であったことを発見した。また、この近似処理を完全に行った場合、mF 関数での最大差分特性確率の上界が 2^{-7}、13 段での最大差分特性確率の上界が 2^{-56} までしか示せないと指摘した。ただし、本来、定数乗算は入力データに依存して差分特性確率になんらかの影響を与え、安全性向上に寄与すると期待されるが、ここでは定数乗算の近似処理において差分特性確率に影響を与えないもの (設計者に対して不利な評価) として安全性の評価を実施していることに注意を要する。

応募者: 2002 年 1 月の暗号技術評価ワークショップのランプセッションで応募者が発表した新しい安全性自己評価に関して、さらに詳細な報告を検討する必要があると認められたため、応募者に対して追加レポートの提出を要求した。この追加レポートによれば、定数乗算による差分特性確率への影響度を実験的に調査 (継続中) しており、評価者 1 が指摘したようなケースにおける定数乗算では差分特性確率に対して 2^{-6} の影響を少なくとも与えるとしている。また、mF 関数での最大差分特性確率の上界が 2^{-13}、13 段での最大差分特性確率の上界が 2^{-104} となるとも述べている。この新しい評価結果に関して検討した結果、評価者 4 も定数乗算の効果を 2^{-7} と見積もっていることなどを考慮すると、ここでの評価結果は妥当なものと考えられる。

評価者 2: 本流部のみで構成されるラウンド関数とした時のモデルに対し、6 段繰返し表現 (最大差分特性確率 2^{-56}) によって安全性評価を実施している。この結果、13 段での最大差分特性確率の上界が 2^{-119} となることを示している。

評価者 3: A3 関数および定数乗算の効果を完全に除外した場合について安全性評価を行っている。その結果、ラウンド関数での最大差分特性確率の上界が $2^{-14.4}$、13 段での最大差分特性確率の上界が $2^{-115.2}$ となることを示している。

評価者 4: T 関数での効果のほかに、本流部の算術加算と A3 関数の (実験的に調査した) 効果および一時鍵生成部の定数乗算の効果をそれぞれ加味すると、本流部、一時鍵生成部、ラウンド関数の最大差分特性確率の上界はそれぞれ 2^{-14}、2^{-7}、2^{-21} となる。これより、13 段での最大差分特性確率の上界が 2^{-126} となることを示している。

以上の評価結果を総合的に判断すると、様々な異なる近似モデルでの安全性評価いずれについても 13 段での最大差分特性確率の上界が 2^{-100} 以下と見積もられ、また実際の CIPHERUNICORN-A についても同程度以上の安全性を有するであろうと期待される。したがって、3 段消去攻撃を想定した場合の差分攻撃に対して、学術的に攻撃不可能であるとまでは証明されないものの、現実にはほぼ不可能であろうと推定される。

2002 年度は応募者らによって新たな安全性評価結果が公開された [3]。文献 [3] では、これまで評価に用いていたラウンド関数を簡略したモデル (mF 関数) に対して、定数乗算部分を近似しないモデル (mF' 関数) を用いて評価しており、13 段の差分特性確率の上界が 2^{-128} 以下となる

ことから 3 段消去攻撃に対して 16 段の CIPERUNICORN-A が安全であると示していた。この評価結果は、これまでの評価と同様に実際のラウンド関数ではなく簡易モデル (mF' 関数) を用いた評価である。また、評価に関しては一部予測値が含まれていた。従って予測値に対して引き続き調査を継続し、その正当性を確認する必要があることを応募者自らも指摘していた。またその後、上記予測値を計算機実験によって求められた計算値に更新した安全性評価結果が公開された [6]。文献 [6] では、計算機を用いた差分特性確率の上界も 13 段において 2^{-128} 以下になると示している。

線形攻撃に対する安全性: ここでは、いずれの評価者も mF 関数を利用したモデルをベースに評価を実施している。

評価者 1: ラウンド関数での最大線形特性確率の上界が $2^{-21.37}$、13 段での最大線形特性確率の上界が $2^{-128.2}$ となることを示している。

評価者 3: ラウンド関数での最大線形特性確率の上界が $2^{-21.68}$、13 段での最大線形特性確率の上界が $2^{-130.1}$ となることを示している。

評価者 4: 応募者による S-box の最大線形特性確率の評価が正しいと仮定した場合、ラウンド関数での最大線形特性確率の上界が $2^{-13.9}$、13 段での最大線形特性確率の上界が $2^{-83.4}$ となることを示している。なお、評価者 4 の検査では、応募者の評価と矛盾する結果が出ており、ラウンド関数での線形特性確率の上界が今回の評価よりも高くなる可能性を否定していない。その一方、A3 関数、定数乗算および一時鍵生成部の影響をほとんど考慮していないことも合わせて注意を要する。そのため、実際には差分攻撃に対する耐性よりも強いと期待されるとも述べている。

以上の評価結果を総合的に判断すると、線形攻撃に対する耐性は差分攻撃に対する耐性よりも強いと期待され、具体的な評価としては 13 段での最大差分特性確率の上界が 2^{-128} 程度以下と推定される。したがって、3 段消去攻撃を想定した場合の線形攻撃による攻撃はほぼ不可能であろうと考えられる。

2002 年度は応募者らによって新たな安全性評価結果が公開された [3]。文献 [3] では、これまで評価に用いていたラウンド関数を簡略したモデル (mF 関数) に対して、定数乗算部分を近似しないモデル (mF' 関数) を用いて評価しており、13 段の線形特性確率の上界が 2^{-128} 以下となることから 3 段消去攻撃に対して 16 段の CIPERUNICORN-A が安全であると示している。この評価結果は、これまでの評価と同様に実際のラウンド関数ではなく簡易モデル (mF' 関数) を用いた評価である。

高階差分攻撃、補間攻撃、スライド攻撃、mod n 攻撃に対する安全性: これらの解読法に対しては特に問題となるような点は発見されなかった。

■鍵スケジュール部に対する安全性評価

弱鍵存在の指摘: 鍵スケジュール部における中間鍵の取り出しを以下のように行っている。ここで、すべてのシンボルは 32 ビットデータを表すものとし、128 ビット鍵は (A, B, C, D)、192 ビット鍵は (A, B, C, D, E, F)、256 ビット鍵は (A, B, C, D, E, F, G, H) を入力とする。

入力: $(A, B, C, D, \ldots, y)$
以下を指定回数繰り返す
　$(A^*, B^*) \leftarrow MT(A, B)$
　$A \leftarrow B^*, B \leftarrow C, C \leftarrow D, \ldots, y \leftarrow A^*$
　指定箇所で中間鍵出力: A

この鍵スケジュール部では、入力が $(A, B, B, B, \ldots, B)$ であるとき、もし $(B, A) \leftarrow MT(A, B)$ を満たすならば、繰り返し中のデータは (何回繰り返そうとも) 常に $(A, B, B, B, \ldots, B)$ のままである。つまり、どの指定箇所での中間鍵もすべて A となり、中間鍵生成のための鍵スケジュール部が実効的にまったく作用していない状態となる。

このような条件を満たす入力を計算した結果、A = `0x61db99c8`、B = `0x9f3d61c8` のときに $(B, A) \leftarrow MT(A, B)$ を満たすことが判明した。つまり、秘密鍵が (`0x61db99c8`, `0x9f3d61c8`, `0x9f3d61c8`, `0x9f3d61c8`, ..., `0x9f3d61c8`) であるとき、すべての中間鍵が秘密鍵の上位 32 ビットと同一な値 `0x61db99c8` となる。また、拡大鍵は中間鍵の順番だけを入れ替えて生成することから、すべての拡大鍵が同じ値 `0x61db99c8` となることをも意味する。

本来の鍵スケジュール部の役割に照らし合わせ、CIPHERUNICORN-A の鍵スケジュール部の構成から推測するに、この種の秘密鍵は非自明な弱鍵であると考えるほうが自然である。なお、現時点で判明している弱鍵はこの一つ (一種類) だけであり、この指摘だけで安全性に重大な問題が生じたということではない。

鍵関連攻撃に対する安全性: 鍵スケジュール部の構成上、鍵関連攻撃に対して安全であると考えられる。

3.3.7.5 ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.52、表 3.53 のとおりである。

暗号化および復号処理、鍵生成まで含めた暗号化および復号処理のすべての測定項目について、今回応募された 128 ビットブロック暗号のなかで、測定プラットフォームによらずに処理速度が最も遅いグループである。また、Pentium III 上ではすべての測定項目について Triple DES と同程度である。

表 3.52: CIPHERUNICORN-A のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	3,984 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/O2 /Oy- (実行速度) を指定	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,569 / 1,574	1,574 / 1,578
2 回目	1,570 / 1,574	1,574 / 1,577
3 回目	1,570 / 1,574	1,574 / 1,578

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	5,644 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-v -fast	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	2,273 / 2,282	2,302 / 2,326
2 回目	2,273 / 2,282	2,309 / 2,327
3 回目	2,273 / 2,282	2,310 / 2,327

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	8,472 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-O4	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,834 / 1,843	1,769 / 1,782
2 回目	1,828 / 1,842	1,769 / 1,782
3 回目	1,828 / 1,842	1,769 / 1,782

表 3.53: CIPHERUNICORN-A の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	4,306 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/O2 /Oy- (実行速度) を指定	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	4,788 / 4,822	4,799 / 4,931
2 回目	4,788 / 4,814	4,798 / 4,815
3 回目	4,787 / 4,830	4,806 / 4,814

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	5,644 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-v -fast	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	7,970 / 8,160	8,802 / 9,025
2 回目	7,961 / 8,164	8,817 / 9,034
3 回目	7,900 / 8,161	8,823 / 9,028

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	8,552 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-O4	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	4,610 / 4,623	5,071 / 5,092
2 回目	4,610 / 4,628	5,071 / 5,100
3 回目	4,610 / 4,624	5,071 / 5,095

また、応募者からは以下の自己評価が報告されている。

プラットフォーム	Pentium III (866MHz), RAM 256MB		
OS およびコンパイラ	Windows NT4.0, Visual C++ 6.0 SP5		
使用言語	ANSI C (インラインアセンブラ有)		
測定項目	128 ビット鍵	192 ビット鍵	256 ビット鍵
鍵スケジュール [cycles/key]	3,219	4,032	3,518
暗号化 [cycles/block]	1,565	1,565	1,565
復号 [cycles/block]	1,559	1,559	1,559
鍵スケジュール + 暗号化 [cycles]	4,780	5,593	5,079
鍵スケジュール + 復号 [cycles]	4,791	5,604	5,090

3.3.7.6　ハードウェア実装評価結果

以下のブロック図 (図 3.16,3.17,3.18) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.54) を示す。なお、アルゴリズムに含まれる多数の乗算は、FPGA にハードマクロとして用意されている 18 ビット乗算器による繰返し処理によって実現されているため、他のアルゴリズムに比べて多くの clock 数を要している。

Key Setup Clock 数	156
Data Randomize Clock 数	126
実装鍵ビット数	128

表 3.54: CIPHERUNICORN-A の HW 実装評価結果

また、応募者からは、ASIC および FPGA 実装に関して以下の自己評価が報告されている。なお、ASIC 実装では 128 ビット鍵のみ利用可能であり、FPGA 実装ではすべての鍵長が選択可能である。

ASIC プロセス　：　NEC 0.25μm CMOS ASIC Design Library
速度優先実装　：　170.60 Mbps, 325.3 Kgates
規模優先実装　：　86.80 Mbps, 290.4 Kgates

FPGA デバイス　：　ALTERA EP20K1500EFC33-1
速度優先実装　：　44.33 Mbps, 7,072 cells + 66 ESB

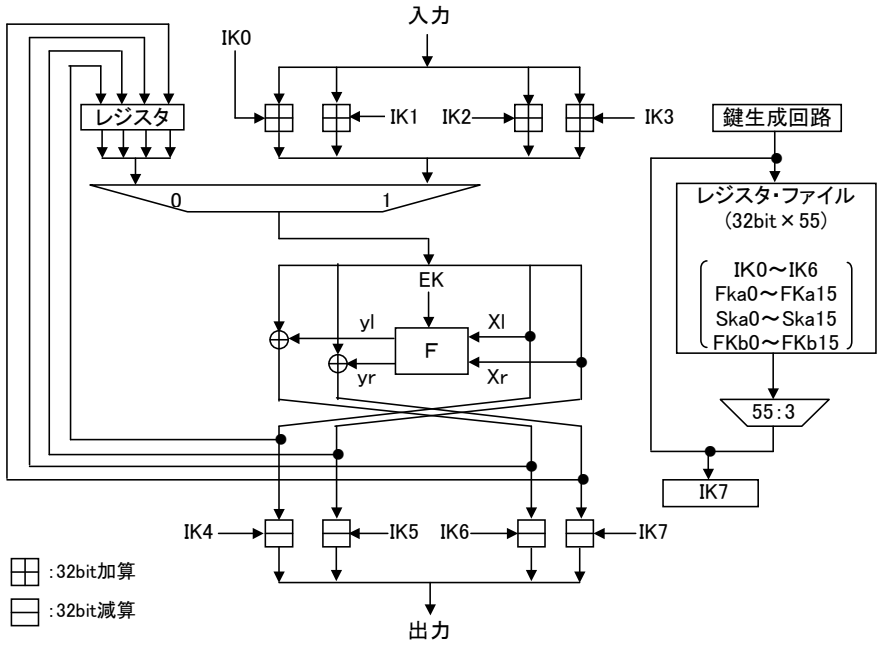

図 3.16: CIPHERUNICORN-A 暗号化回路ブロック図

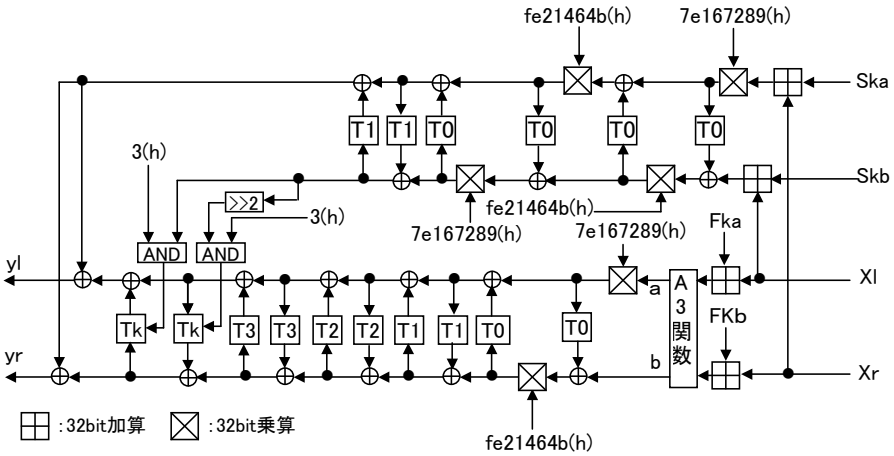

図 3.17: F 関数内部ブロック図

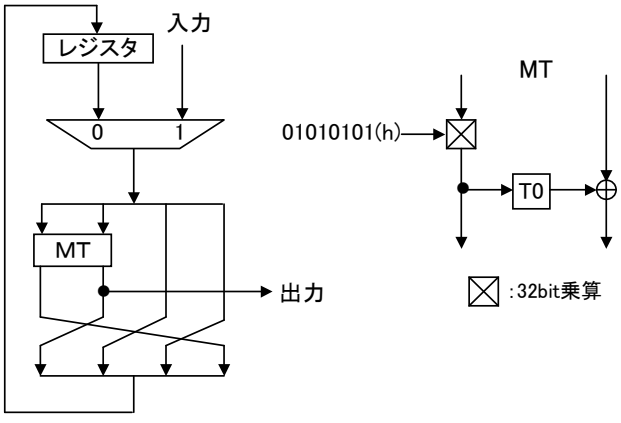

図 3.18: CIPHERUNICORN-A 鍵生成回路ブロック図

参考文献

[1] 角尾幸保、久保博靖、宮内宏、中村勝洋、「128 ビットブロック暗号 CIPHERUNICORN-A」、2000 年暗号と情報セキュリティシンポジウム SCIS2000, A18, 2000.

[2] 角尾幸保、太田良二、宮内宏、中村勝洋、「分散型暗号強度評価支援システム」、2000 年暗号と情報セキュリティシンポジウム SCIS2000, A53, 2000.

[3] 角尾幸保、久保博靖、山田真紀、洲崎智保、宮内宏、「CIPHERUNICORN-A の差分解読／線形解読に対する安全性について」、信学技報, ISEC2002-42 (2002-07).

[4] 青木和麻呂、古屋聡一、 盛合志帆、「CIPHERUNICORN-A 実装に対する乗算の時間差を使ったタイミング攻撃」、2003 年暗号と情報セキュリティシンポジウム SCIS2003, 4D-3, 2003.

[5] 角尾幸保、洲崎智保、久保博靖、辻原悦子、宮内宏、「S-box におけるキャッシュ遅延を利用した CIPHERUNICORN-A へのタイミング攻撃」、2003 年暗号と情報セキュリティシンポジウム SCIS2003, 4D-4, 2003.

[6] 角尾幸保、久保博靖、茂真紀、洲崎智保、宮内宏、「CIPHERUNICORN-A の差分解読／線形解読に対する安全性について (II)」、2003 年暗号と情報セキュリティシンポジウム SCIS2003, 5D-1, 2003.

3.3.8 Hierocrypt-3

3.3.8.1 技術概要

Hierocrypt-3 は、2000 年に、情報処理学会・コンピュータセキュリティ研究会において、株式会社東芝により提案された共通鍵ブロック暗号 [11] である。ブロック長は 128 ビットであって、三つの鍵長 (128、192、256 ビット) をサポートする。鍵長と同程度に期待される安全性と、効率的なソフトウェア実装およびハードウェア実装を目指して設計された暗号アルゴリズムであるが、特に IC カードやミドルウェアでの暗号化の高速性を重視している。

3.3.8.2 技術仕様

- 主要な共通鍵暗号攻撃法に対して十分強く、主要なプラットフォーム上で高速で動作し、実装サイズもコンパクトになることを目標とした。
- 計算効率と安全性の両立を高めるため、データランダム化部には SPN 構造を再帰的に利用した入れ子型 SPN 構造を採用した。
- 入れ子型 SPN 構造は非常に簡潔であり、十分な安全性を維持しつつ、構成要素もある程度独立に設計できる。さらに、ブロック長の変化にも柔軟に対応できる。
- S-box は、ガロア体上のべき乗関数を基本とし、差分攻撃・線形攻撃に対する耐性に関する最適化を行なった。さらに、べき乗関数をビット置換とアフィン変換で挟んで代数的攻撃法の適用を困難にした。
- 拡散層は、符号理論を用いて活性 S-box 数の下限が大きな値を取るものを多数生成して候補とし、安全性と実装効率の条件で絞り込んだ。
- 鍵スケジュール部は、128 ビット Feistel 型構造を基本構造とし、中間出力を組み合わせて拡大鍵を生成する。復号時にも on-the-fly での鍵設定の初期遅延が小さくなるよう、中間鍵列が途中で逆転して戻ってくる折り返し型の構造を採用した。
- 段数は鍵長に依存し、鍵長 128、192、256 ビットに対し各々 6、7、8 段である。

3.3.8.3 その他

Hierocrypt は東芝が開発した共通鍵ブロック暗号のファミリーに付けられた名前。このファミリーには、ブロック長 128 ビットの Hierocrypt-3 とブロック長 64 ビットの Hierocrypt-L1 があり、いずれもデータランダム化部が入れ子型 SPN 構造と呼ばれる SPN 構造の一種で設計されているという共通点がある。

3.3.8.4 安全性評価結果

現時点 (2003 年 3 月) では、いくつかの安全性に関する解析結果が知られている。しかし、どの安全性評価も、安全性に関する決定的な欠点となる結果を示したものではない。

設計者による自己評価書では、共通鍵暗号のさまざまな攻撃手法についての安全性の検討結果

を行っている。特に差分攻撃・線形攻撃については信頼性の高い評価を行っている。また、新しい評価技術 [8, 9] に対し継続的に、新しい評価手法を取り入れた、あるいは改良した評価による安全性評価結果の更新を行っている [15, 14, 18]。最新 (2003 年 1 月時点) の結果によると、Hierocrypt-3 はいくつかの仮定のもとで、差分攻撃・線形攻撃に対する証明可能な安全性がある。具体的には 2 段分のラウンド関数を差分近似・線形近似した場合の差分確率・線形確率に上限を与えることができ、それぞれ 2^{-96} であることが知られている [18]。これらにより Hierocrypt-3 が差分攻撃・線形攻撃に対して十分強固であると考えることができる。

Hierocrypt-3 の設計者が最も注目する攻撃法のひとつである SQUARE 攻撃については、3.5 段での解読可能性が指摘されている [1]。これは、設計者の当初の見解である、「Rijndael よりも少ない段数 (2.5 段) で SQUARE 攻撃に対して安全である」という結論とは若干異なる (応募者による SCIS2001 における発表)。しかし、Hierocrypt-3 は 6 段以上で使われる仕様となっており、この仕様での安全性に直接の脅威を与えるものではない。

また、(意味が多少あいまいであるが) 仕様書中の記述「安全面に関しては、拡大鍵間の単純な依存関係によって、鍵の全数探索による探索範囲が実質的に狭くなることが無いようにすることである」の「拡大鍵間の単純な依存関係」としていくつかの線形関係式が得られている [4, 7]。

また、アバランシュ性の検証では、鍵スケジュール部、ラウンド関数で偏りがあることが示された。また設計指針の一つとして、「MDS 行列と S-box を組合せたときの多項式表現の項数が最大であること」があったが、これに反して評価結果として、多項式表現の項数は (比較的大きな値をとってはいるものの) 最大値でないことが確認されている。その他、補間攻撃に関する検討 [5, 6]、不能差分攻撃に関する検討 [10]、実験的な乱数検定結果 [2] などの新しい結果が発表された。

しかし、これら評価どれもが仕様どおりの Hierocrypt-3 の安全性を脅かすものではない。安全性に関する結果をまとめた情報は NESSIE プロジェクトのいくつかの文書で確認することができる [17, 16]。SPN 構造、S-Box 評価、MDS などほとんどの要素技術がこれまでの暗号学の研究結果を踏まえた設計となっており、個々については今後の明らかかつ致命的な欠点は起らないと考えられる。

最後に、設計指針とアルゴリズムは直感的、理論的に結びつくものであり、設計者が落とし戸を意図的に組み込んだとは考えにくい。

3.3.8.5　ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.55、表 3.56 のとおりである。

備考　UltraSPARC IIi と Alpha 21264 の測定において、(カッコ) 内の値は応募者による測定プログラムの改変した場合の測定値。測定プログラムは汎用性を持たせるため巨大なバッファ領域を確保しているが、その領域を必要な分だけ取るように改変した。速度評価の主旨を違えるような改変は行っていないことは確認済み。

表 3.55: Hierocrypt-3 のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ (MMX 命令)	
プログラムサイズ	68,832 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	VC++6.0 Win32 Release (Default)	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	404 / 406	426 / 428
2 回目	404 / 406	426 / 428
3 回目	404 / 406	426 / 428
UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	38,936 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -native -fast -xarch=v8plusa -xCC (暗号化) cc -native -fast -xarch=v9 -xCC (復号)	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	511 (471) / 554 (473)	759 (612) / 826 (616)
2 回目	510 (471) / 556 (473)	758 (612) / 826 (616)
3 回目	510 (471) / 555 (473)	757 (612) / 826 (616)
Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	58,152 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -O3	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	420 (399) / 424 (406)	427 (386) / 429 (393)
2 回目	420 (399) / 424 (406)	427 (386) / 430 (394)
3 回目	420 (399) / 423 (407)	427 (386) / 430 (393)

表 3.56: Hierocrypt-3 の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ (MMX 命令)	
プログラムサイズ	68,832 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	VC++6.0 Win32 Release (Default)	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	726 / 728	1,345 / 1,358
2 回目	726 / 729	1,344 / 1,357
3 回目	726 / 728	1,346 / 1,358

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	38,936 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -native -fast -xarch=v8plusa -xCC (暗号化) cc -native -fast -xarch=v9 -xCC (復号)	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	823 (761) / 828 (822)	2,673 (2,612) / 2,684 (2,627)
2 回目	823 (761) / 828 (821)	2,671 (2,611) / 2,683 (2,627)
3 回目	824 (761) / 828 (823)	2,670 (2,610) / 2,683 (2,627)

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	58,152 byte (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	cc -O3	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	675 (668) / 679 (672)	1,130 (1,130) / 1,142 (1,141)
2 回目	675 (668) / 678 (673)	1,130 (1,130) / 1,142 (1,142)
3 回目	675 (668) / 679 (672)	1,130 (1,130) / 1,142 (1,142)

また、応募者からは以下の自己評価が報告されている。

プラットフォーム	:	Mobile Pentium II (600MHz), 192MB
OS およびコンパイラ	:	Windows 2000 SP2, Sun JDK 1.3.1 without JCE
使用言語	:	Java
鍵スケジュール (暗号化)	:	2,243 cycles/key
暗号化	:	2,814 cycles/block
復号	:	3,033 cycles/block

■IC カード実装 Z80 による IC カード実装評価を実施した。表 3.57 は、128 ビット鍵を利用したときの鍵スケジュール部 + データランダム化部処理時間測定結果である。

表 3.57: Hierocrypt-3 の Z80 上での鍵スケジュール部 + データランダム化部処理時間測定結果

	ROM [bytes]	RAM [bytes]	Stack [bytes]	処理時間 [states]
暗号化	2,577	73	8	49,919
復号	3,662	73	8	71,782
暗復共用	4,746	—	—	—

3.3.8.6 ハードウェア実装評価

以下のブロック図 (図 3.19,3.20,3.21) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.58) を示す。

Key Setup Clock 数	15
Data Randomize Clock 数	12
実装鍵ビット数	128

表 3.58: Hierocrypt-3 の HW 実装評価結果

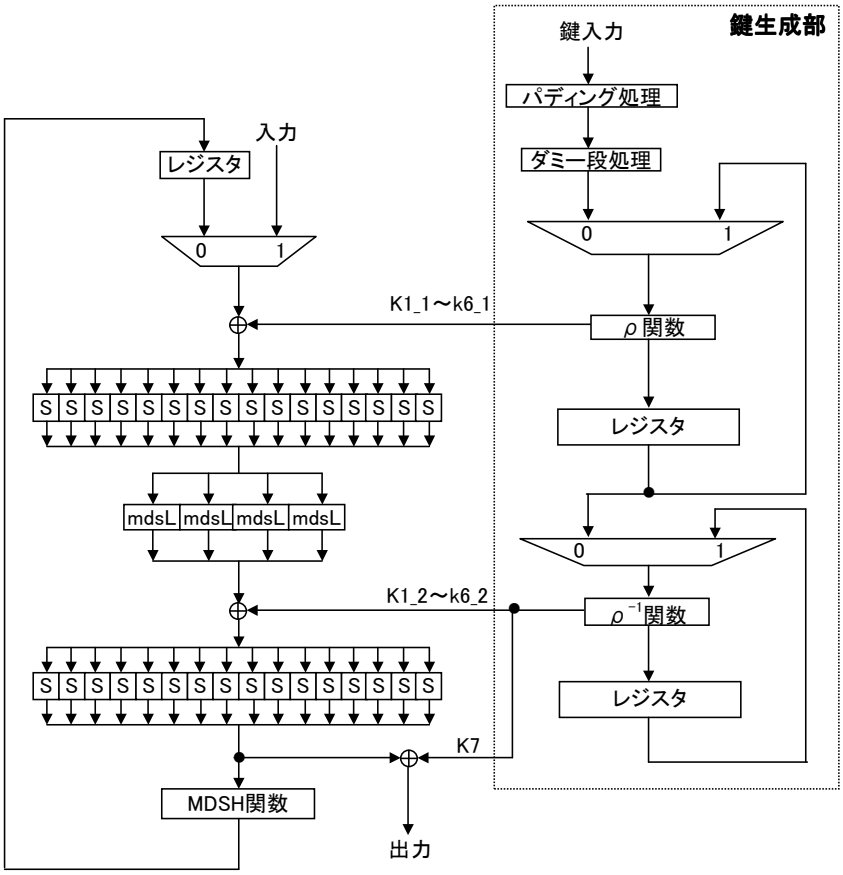

図 3.19: Hierocrypt-3 暗号化回路・鍵生成部回路 (右側点線内) ブロック図

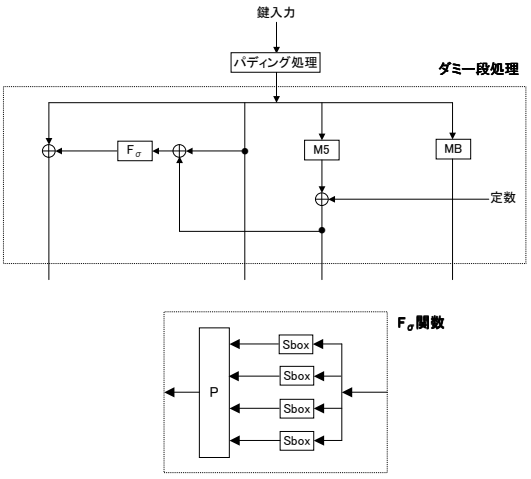

図 3.20: ダミー段処理・F_σ 関数内部ブロック図

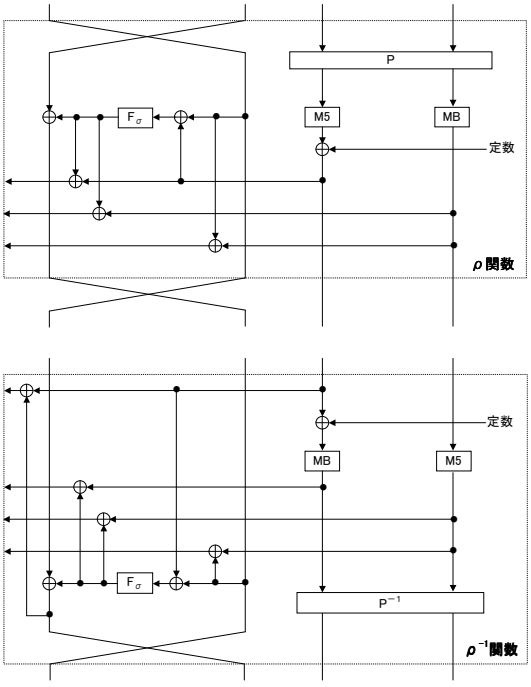

図 3.21: ρ・ρ^{-1} 関数内部ブロック図

また、応募者からは、ASIC および FPGA 実装に関して以下の自己評価が報告されている。

ASIC プロセス	:	0.25μm CMOS ASIC Design Library
速度優先実装	:	2,067 Mbps, 143.9 Kgates
規模優先実装	:	135 Mbps, 18.1 Kgates
ASIC プロセス	:	0.13μm CMOS ASIC Design Library
速度優先実装	:	3,082 Mbps, 111.8 Kgates
FPGA デバイス	:	ALTERA Max+plus II ver. 9.6
速度優先実装	:	52.6 Mbps, 22.7 Kcells
規模優先実装	:	4.1 Mbps, 6.3 Kcells

参考文献

[1] P. S.L.M. Barreto, V. Rijmen, J. Nakahara Jr., B. Preneel, J. Vandewalle, and H. Y. Kim, "Improved SQUARE Attacks Against Reduced-Round HIEROCRYPT," Fast Software Encryption, 8th International Workshop, FSE 2001, LNCS 2355, Springer-Verlag, 2001.

[2] Y. Braziler, "The statistical evaluation of the NESSIE submission Hierocrypt-3," Public reports of NESSIE project, NES/DOC/TEC/WP3/021/1, available at http://www.cosic.esat.kuleuven.ac.be/nessie/reports/.

[3] Y. Braziler, "The statistical evaluation of the NESSIE submission Hierocrypt-L1," Public reports of NESSIE project, NES/DOC/TEC/WP3/022/1, available at http://www.cosic.esat.kuleuven.ac.be/nessie/reports/.

[4] S. Furuya and V. Rijmen, "Observations on Hierocrypt-3/L1 Key-scheduling Algorithms," Proceedingds of the second open NESSIE Workshop, 2001.

[5] 古屋聡一、櫻井幸一、「SPN 構造をもつブロック暗号の代数近似について」、第 4 回コンピュータセキュリティシンポジウム (CSS2001) 講演予稿集, CSS2001, 6B-1, 2001.

[6] 古屋聡一、櫻井幸一、「Sudan の Reed-Solomon 符号復号アルゴリズムを使ったブロック暗号の補間攻撃」、電子情報通信学会技術研究報告, COMP2002-22, 2002.

[7] 金丸昌司、白井太三、阿部譲司、「Hierocrypt-L1/3 の鍵スケジュール解析」、電子情報通信学会技術研究報告, ISEC2002-91, 2002.

[8] L. Keliher, H. Meijer, and S. Tavares, "Improving the Upper Bound on the Maximum Average Linear Hull Probability for Rijndael," Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001 Toronto, LNCS 2259, Springer-Verlag, 2001.

[9] L. Keliher, H. Meijer, and S. Tavares, "Dual of New Method for Upper Bounding the Maximum Average Linear Hull Probability for SPNs," IACR's ePrint archive, 2001/033, available at http://eprint.iacr.org/.

[10] C. MJ. Kim and K. Kim, "Impossible Differential Cryptanalysis of Hierocrypt-3 Reduced to 3 Rounds," Proceedingds of the second open NESSIE Workshop, 2001.

[11] 村谷博文、大熊建司、佐野文彦、本山雅彦、川村信一、「64 ビット版 Hierocrypt の提案」、情報処理学会研究報告 CSEC11-9, 2000.

[12] 大熊建司、村谷博文、佐野文彦、川村信一、「Specification and Assessment of the block cipher Hierocrypt」、電子情報通信学会技術研究報告 IT99-102, ISEC99-141, SST99-150, 2000.

[13] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura, “The Block Cipher Hierocrypt,” Selected Areas in Cryptography, 7th Annual International Workshop, SAC 2000, LNCS 2012, Springer-Verlag, 2001.

[14] 大熊建司、佐野文彦、清水秀夫、川村信一、「入れ子型 SPN 構造の証明可能安全性に関する補足事項」、2002 年暗号と情報セキュリティシンポジウム SCIS2002 講演予稿集, SCIS2002 5B-2, 2002.

[15] K. Ohkuma, H. Shimizu, F. Sano, and S. Kawamura, “Security Assessment of Hierocrypt and Rijndael against the Differential and Linear Cryptanalysis (Extended Abstract),” IACR’s ePrint archive, 2001/070, available at http://eprint.iacr.org/.

[16] B. Preneel, B. Van Rompay, L. Granboulan, G. Martinet, S. Murphy, R. Shipsey, J. White, M. Dichtl, P. Serf, M. Schafheutle, E. Biham, O. Dunkelman, M. Ciet, J-J. Quisquater, F. Sica, L. Knudsen, and H. Raddum,“NESSIE Phase I: Selection of Primitives,” NESSIE deliverables, available at http://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/.

[17] B. Van Rompay, V. Rijmen, and J. Nakahara Jr., “A first report on CS-Cipher, Hierocrypt, Grand Cru, SAFER++, and SHACAL,” Public reports of NESSIE project, NES/DOC/KUL/WP3/006/1, available at http://www.cosic.esat.kuleuven.ac.be/nessie/reports/.

[18] F. Sano, K. Ohkuma, H. Shimizu, and S. Kawamura, “On the Security of Nested SPN Cipher against the Differential and Linear Cryptanalysis,” IEICE Trans. Fundamentals, Vol.E86-A, No.1, pp.37–46, Jan. 2003.

[19] F. Sano, K. Ohkuma, H. Shimizu, M. Motoyama, and S. Kawamura, “Efficient Implementation of Hierocrypt,” Proceedings of the second open NESSIE Workshop, 2001.

[20] 清水秀夫、佐野文彦、本山雅彦、大熊建司、川村信一、「SPN 型ブロック暗号の実装について」、電子情報通信学会技術研究報告, ISEC2001-55, 2001.

[21] 川幡剛嗣、角尾幸保、齊藤照夫、辻原悦子、宮内宏、「Hierocrypt-L1/-3 へのタイミング攻撃」、2003 年暗号と情報セキュリティシンポジウム SCIS2003 講演予稿集, SCIS2003 4D-2, 2003.

3.3.9 RC6

3.3.9.1 技術概要

RC6 は 1998 年に R. Rivest らにより発明され、公募に対し RSA セキュリティ株式会社として応募している可変ブロック長 (推奨 128 ビット) の共通鍵暗号である [1]。設計には、その前身である RC5 の思想を受け継ぎ、簡潔な構造で、高速で効率的な実装や広い範囲の解析が可能になることを目指している。具体的には、データ依存巡回シフトや、拡大鍵の整数加算などの演算を安全性確保の主軸とし、さらに、ラウンド関数内に乗算を用いることにより、一段あたりのデータの攪拌量を大きくし、安全性の向上、暗号化処理の効率化を目指している。

3.3.9.2 技術仕様

RC6 は広範なパラメータを持ち、正確には RC6-w/r/b と表現される。w はワードのビット長、r は段数、b は鍵のバイト長である。構造は、平文ブロックを 4 分割した変形 Feistel 型構造であり、ワード長 w の 4 倍の平文ブロック長を持つ。今回の応募は、ワード長 w = 32 bits、鍵長 b = 16、24、32 bytes で、段数 r = 20 を推奨値 (RC6-32/20/{16,24,32}) として提案している。テーブルを使用しておらず、コンパクトなソフトウェア実装が可能である。その本体部分は 176 bytes の鍵スケジュール部とほんの僅かな追加メモリで実装が可能である。ワード長が 32 ビットの場合、暗号アルゴリズムで使用される演算の、算術加減算、排他的論理和、算術乗算、左右巡回シフト演算は、いずれも、32 ビットワード単位であり 32 ビット CPU の演算を効率良く使用するアルゴリズムとなっている。速度面では、これら演算の処理速度の高さが、高速な実装に結びつく。

3.3.9.3 その他

RC6 は CRYPTREC 応募暗号であったが、CRYPTREC 事務局では、2002 年 10 月 16 日付書類で、応募者の RSA セキュリティ株式会社より、「知的財産権上の問題により、今後 RC6 の普及活動は行わない」との連絡を受領している。このため、CRYPTREC としての評価は、2002 年 10 月で終了する。

3.3.9.4 安全性評価結果

RC6 は AES 提案暗号として、評価を受け、詳細評価対象の 5 暗号の 1 つに選ばれている。今回の CRYPTREC の評価も受け、これらにおいて、提案版の RC6 の欠陥は報告されていない。期待する暗号は、攻撃に必要な平文数が平文総数未満かつ攻撃計算量が鍵の全数探索を下回る攻撃法が無いものである。

以下、各種の攻撃法に対する RC6 の耐性をまとめる。

差分攻撃や線形攻撃に対する耐性は、証明可能安全性の議論に基づくものでは無いが、特性確率に関し、自己評価書で適切な考慮に基づく評価がなされている。RC6 のようにデータ依存型

巡回シフトを用いるアルゴリズムでは、シフト数により差分経路や線形近似経路が変わり、これら経路ごとの特性確率の和に関する考察が必要となるが、この点も十分考慮されている。結果としては、差分攻撃で 12 段まで、線形攻撃で 16 段までは、解読に必要な平文数が全平文数を下回るという意味で、期待する暗号強度に達していないが、18 段でそれを上回る [2]。なお、複数の線形近似式を使用した攻撃において、2^{-90} の割合で存在する弱鍵の場合、$2^{126.9}$ の平文と $2^{192.9}$ の計算量で、18 段 RC6 の鍵推定が可能である [4]。

高階差分やその他の攻撃の中で、RC6 に対し、効果を上げているのは、カイ 2 乗攻撃である。この攻撃はカイ 2 乗統計量を使う攻撃であり、それによると 15 段が、2^{119} の選択平文と 2^{215} の計算量で、2^{138} のメモリを使い鍵の推定が可能である [3]。これらの範囲の段数では RC6 は期待する暗号強度に達していない。しかし、平文数等の数字は、通信速度や計算機能力が毎年 10 倍で上昇しても、今後 10 年間はあり得ない環境であり、実際的な攻撃とは考えられないが、RC6 における弱鍵を含めた統計的強度評価研究の進展に、今後とも注目する必要がある。

高階差分攻撃では、9 段で期待する暗号強度となり、アバランシュ評価では、6 段で期待する特性になることが報告されており、現在のところこれらの観点からは、十分な強度を持つと考えられる。

以上のように、RC6 は、現在知られている最強の攻撃に対し、16 段までは期待する暗号強度に達していないが、仕様段数は 20 段であり、それが少ないとの意見もあるが、現在における安全性には問題ないと考える。

3.3.9.5　ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.59、表 3.60 の通りである。

備考:　Pentium III で測定したコードは Microsoft Windows9X 用の製品プログラム、UltraSPARC IIi で測定したコードは SUN Solaris 用の製品プログラムを性能評価テスト仕様にあわせて修正したものであり、ベースとなった製品 (「BSAFE-Crypto-C5.1」) は現在販売中である。

Pentium III における暗号化および復号のデータ処理速度は、今回応募されたブロック暗号の中で最速である。しかし、拡大鍵生成まで含めた速度では、Pentium III の測定結果で、最も遅い暗号に近い。UltraSPARC IIi における暗号化および復号さらに拡大鍵生成まで含めたすべての速度は、今回のブロック暗号の中で、最も遅いデータに近い。提案者より提出されたものはいずれも製品版プログラムであり、今回の速度測定用に特化したものでは無いことである。前者はアセンブリ言語、後者は C 言語で記述してある。

なお、RC6 のソフトウェア実装評価としては、幾つかの評価環境 (CPU、言語、他) のもとでの実装結果が多数報告 [7] されているので、そちらも参考にされたい。

表 3.59: RC6 のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	1,200 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/O2 (マイクロソフト C コンパイラ)	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	258 / 260	262 / 266
2 回目	258 / 260	262 / 265
3 回目	258 / 259	262 / 265
UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	3,940 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	xo5 (WS Compiler C/SPARC オプティマイズ 5)	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	2,048 / 2,088	2,024 / 2,076
2 回目	2,047 / 2,088	2,023 / 2,074
3 回目	2,048 / 2,089	2,026 / 2,077

表 3.60: RC6 の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	1,200 bytes (暗号化/復号), 1,500 bytes (鍵スケジュール)	
コンパイラオプション	/O2 (マイクロソフト C コンパイラ)	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	1,631 / 1,644	1,633 / 1,639
2 回目	1,630 / 1,645	1,633 / 1,643
3 回目	1,630 / 1,642	1,633 / 1,640
UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	3,940 bytes (暗号化/復号), 2,196 bytes (鍵スケジュール)	
コンパイラオプション	xo5 (WS Compiler C/SPARC オプティマイズ 5)	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	4,078 / 4,111	4,026 / 4,054
2 回目	4,078 / 4,111	4,024 / 4,055
3 回目	4,075 / 4,112	4,019 / 4,054

■IC カード他のソフトウェア実装　自己評価書には、第 3 者実装を含め、Java、IC カード、DSP による実装に関し以下の特徴が記載されている。

Java　暗号処理の簡易性は Java における、コードの大きさ、パフォーマンス、およびダイナミック RAM の量といった点に反映される。AES の評価過程で行われた各調査では、RC6 は Java 環境の中で、顕著なパフォーマンスを示している。

IC カード　RC6 のパフォーマンスは、ARM チップや他の高機能プロセッサを採用した IC カードにおいて優れた暗号処理パフォーマンスを示す。

DSP　RC6 は余分なメモリを使うルックアップテーブルが不要なので、RC6 はこの種のプロセッサにおいても、十分なパフォーマンスが得られる。

3.3.9.6　ハードウェア実装評価結果

また、RC6 のハードウェア実装評価としては、ASIC, FPGA の実装例がいくつか報告されている [7]。

参考文献

[1] R.L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, The RC6 Block Cipher. Algorithm specification, August 20, 1998. Available at `http://www.rsasecurity.com/rsalabs/rc6/`

[2] S. Contini, R.L. Rivest, M.J.B. Robshaw, and Y.L. Yin, The security of the RC6 Block Cipher, August 20, 1998. Available at `http://www.rsasecurity.com/rsalabs/rc6/`

[3] L.R. Knuddsen and W. Meier, “Correlations in RC6 with a reduced number of rounds,” FSE2000, LNCS 1978, pp.94-108, 2001.

[4] T. Shimoyama, M. Takenaka, and T. Koshiba, “Multiple Linear Cryptanalysis of a reduced round RC6,” SCIS2002, Proceedings of the 2002 Symposium on Cryptography and Information Security, pp.931–936, 2002. (also presented at FSE2002).

[5] A. Elbirt, W. Yip, B. Chetwynd, and C. Parr, “An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists,” Proceedings of 3rd AES conference, pp.13-27, (2000).

[6] B. Weeks, M. Bean, T. Rozylowicz, and C. Ficke, “Hardware performance simulations of Round 2 AES algorithms,” Proceedings of 3rd AES conference, pp.286-304, (2000).

[7] J. Nechvatal, et al., Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000. Available at `http://csrc.nist.gov/encryption/aes/round2/r2report.pdf`

3.3.10 SC2000

3.3.10.1 技術概要

- 本暗号は富士通株式会社および東京理科大の研究者の考案によるもので 2000 年に学会発表され、富士通により提案された暗号である。AES と同じインタフェースである 128 ビットデータ入出力、128、192、256 ビット鍵長を持つ共通鍵ブロック暗号である。
- 暗号全体の構造は Festel 型構造と SPN 型構造の重ね合わせという新規構造であるが、S-box などの各暗号部品には十分に安全性の検証を行なった安全性に定評のある部品のみを用いることで、全体の安全性を検証しやすい構造としている。
- 高速実装のための技術としては、SPN 型構造として Bitslice と呼ばれる最新の高速実装法が適用可能な構造を採用していると共に、非線形演算処理に関して CPU の 1 次キャッシュの大きさに応じた高速実装が可能であるように設計されている。
- ハードウェア実装では、暗号化部において 6 ビット入出力以下の非線形演算装置と論理演算のみを用いることでコンパクトな実装を目指している。
- 想定するアプリケーションとしては、次世代ネットワーク間データの高速暗号通信、大容量データベースの高速暗号化処理、IC カードの認証処理および暗号データの送受信がある。

3.3.10.2 技術仕様

■データランダム化部 32 ビット ×4 の入力平文データを、鍵スケジュール部により作成された拡大鍵テーブルを用いて暗号化し、32 ビット ×4 のデータを暗号文として出力する。内部関数として 32 ビット ×4 の入出力である I 関数、B 関数、R 関数を持つ。このうち I 関数は鍵を排他的論理和する関数、B 関数と R 関数はデータを攪拌する関数である。128 ビット鍵の時の構成は、I 関数が 14 段、B 関数が 7 段、R 関数が 12 段で、データ攪拌関数 (B 関数と R 関数) は合計 19 段である。192、256 ビット鍵の場合には、I 関数が 16 段、B 関数が 8 段、R 関数が 14 段でデータ攪拌関数は合計 22 段である。各関数間の接続は、前段の関数の出力をそのまま次段の入力とするストレート接続 (-) と、前段の関数の出力を 64 ビットずつに分割してスワップし次段の入力とするクロス接続 (×) がある。各関数を I-B-I-R×R のように接続しこの処理を 6 回 (鍵長 128 ビット時) または 7 回 (鍵長 192、256 ビット時) 繰り返し、最後に I-B-I を経て暗号文が出力される。使用する拡大鍵は、128 ビット鍵の時は 32 ビット拡大鍵が 56 個、192、256 ビット鍵の場合は 32 ビット拡大鍵が 64 個である。

■復号関数 32 ビット ×4 の入力暗号文データを、入力の拡大鍵テーブルを用いて復号し、32 ビット ×4 のデータを復号文として出力する。内部関数として 32 ビット ×4 の入出力である I 関数、B^{-1} 関数、R 関数を持つ。このうち I 関数、R 関数はデータランダム化部のものと同じで、B^{-1} 関数は B 関数の逆関数である。各関数を I-B^{-1}-I-R×R のように接続しこの処理を 6 回 (鍵長 128 ビット時) または 7 回 (鍵長 192、256 ビット時) 繰り返し、最後に I-B^{-1}-I を経て復号文が出

力される。

■鍵スケジュール部　ユーザ鍵から 32 ビット拡大鍵 56 個 (鍵長 128 ビット時) または 64 個 (鍵長 192、256 ビット時) を生成する。中間鍵生成関数と拡大鍵生成関数からなる。まずユーザ鍵 32 ビット ×4 を 32 ビット ×8 に拡張して中間鍵生成関数により中間鍵を作成し、次いで拡大鍵生成関数により所定数の 32 ビット拡大鍵を生成する。

3.3.10.3　安全性評価結果

■総評　次の 3 種類の解析を行なったが、提案の構成においては明確な弱点は発見されなかった。現在のところ SC2000 の安全性上の欠陥は見つかっていないが、今後さらなる解析を重ねていくことが必要であると思われる。その他に 2001 年 1 月以降に [4, 5, 6] が発表されている。

■データランダム化部の従来型攻撃に対する安全性　差分攻撃あるいは線形攻撃への耐性を保証するために、差分特性確率や線形特性偏差の理論的上限を評価する設計手法が知られている [1]。SC2000 では、DES 等の安全性評価で用いられた有意な差分特性確率・線形特性偏差を持つ近似式を探索し、有意な確率あるいは偏差をもった近似式が存在しないことをもってこれら攻撃に対する耐性を示している [2]。近似式の導出を効率的に行なうために探索対象を truncated vector の差分波及パターンに置き換えているという方法をとっている [2]。

差分攻撃について、15 段の差分特性近似確率は、-B-R×R-による 3 段繰り返し構造を基本とする場合には 2^{-134} 以下であることが分かった。すなわち、差分攻撃に利用できる 15 段の差分特性近似式がないことを意味する。

また、線形攻撃に対しても truncated vector を利用した評価が可能である。15 段の線形特性近似確率は 3 段繰り返し構造を基本とする場合には 2^{-142} 以下になることが分かった。すなわち、線形攻撃に利用できる線形特性近似式がないことを意味する。

一方、2001 年 1 月の提案者グループによる発表 [3] では、3 段繰り返し構造による差分探索・線形検索を行なった結果、確率 2^{-33} で成立する差分特性と 2^{-34} で成立する線形特性が見つかり、この結果、128 ビット鍵の場合の 19 段のうち 13 段まで攻撃することが可能であることが報告されている。また、提案者グループによる [11] では、6 段繰り返し構造の差分特性・線形特性がそれぞれ確率 2^{-58} と 2^{-56} で見つかり、攻撃に必要な平文数とメモリ量を削減している。しかしこれらの差分特性・線形特性を用いても 19 段の SC2000 の特性確率はそれぞれ確率 2^{-159} と 2^{-156} であり差分攻撃あるいは線形攻撃は適用されない。一方、[7] によれば確率 2^{-106} で成立する 11 段の差分特性が見つかっている。この差分特性を用いて 13 段の SC2000 の鍵の一部が求められることが分かっている。

代数次数が小さい関数により構成される暗号には高階差分攻撃が有効に働く。SC2000 は少なくとも 2 次の係数を持つ B 関数および R 関数が 128 ビット鍵では 19 段利用されており、高階差分解読が適用できないと判断される。高階差分攻撃・補間攻撃に対しては、必要平文組数が 2^{64} 以上、計算量が 2^{256} 未満で攻撃可能な最高段数 8 段であるのに対して、仕様段数 22 段であるか

ら、高階差分攻撃・補間攻撃の立場では問題がないことが確認された。

truncated 差分攻撃に対しては、通常の差分攻撃に対するセキュリティマージンがそれほど大きくないことから、さらに詳細な評価を行なう必要がある。

カイ 2 乗攻撃・分割攻撃の適用可能性について、平文と暗号文の部分情報間に統計的な相関性を起こす構造を調べたが該当するものは見当たらなかった。今後計算機実験などでさらに調べることが望ましい。

不能差分攻撃、ブーメラン攻撃、mod n 攻撃、非全単射攻撃の各攻撃法に対する安全性を考察したが脅威となる欠点は認められなかった。

■鍵スケジュール部の従来型攻撃に対する安全性 鍵の全数探索は共通鍵暗号に適用されるもっとも非効率的であるが確実な解読方法である。既存の技術レベルでは 128 ビット以上の鍵の全数探索は現実的ではないと考えられる。弱鍵について、自己評価書には中間鍵の衝突の有無と、すべての中間鍵が一致する可能性について述べられており、評価は妥当であった。SC2000 では拡大鍵の計算にあたって、重複する場合が見られず鍵から拡大鍵の生成が有効に行なわれている。統計的性質についてカイ 2 乗特性を調べたが問題となる検定値は見られなかった。

以上のように鍵スケジュール部に関して問題となる欠点は認められなかった。

■実装に関する攻撃に対する安全性 SC2000 に対する実装攻撃として、ある種の特殊な条件のもとで、関連鍵を用いた、キャッシュメモリのヒット・ミスヒットの処理時間差を使ったタイミング攻撃を行うことにより、すべての秘密鍵が導出された例が報告されている [12]。本攻撃法は、関連鍵という特別な組み合わせの鍵を用いる上、実装法や使用環境に依存した攻撃法であるので、SC2000 のアルゴリズム自体の安全性に致命的な欠点をもたらすものではない。実装攻撃の一般的概要、および対策法の詳細については、第 6 章も参照すること。

3.3.10.4 ソフトウェア実装評価結果

以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.61、表 3.62 のとおりである。

備考 UltraSPARC IIi と Alpha 21264 の測定において、(カッコ) 内の値は応募者による測定プログラムの改変した場合の測定値。測定プログラムは汎用性を持たせるため巨大なバッファ領域を確保しているが、その領域を必要な分だけ取るように改変した。速度評価の主旨を違えるような改変は行っていないことは確認済み。

復号の処理時間を、暗号の処理時間と比べると、Pentium III と Alpha 21264 においては数 % 大きくなり、UltraSparc IIi においては逆に数 % 少なくなった。これらは特に問題になるほどは大きくない。

表 3.61: SC2000 のデータランダム化部処理速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	21,340 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/G6 /O2 /ML /W3 /GX	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	389 / 391	408 / 410
2 回目	388 / 392	408 / 411
3 回目	388 / 391	408 / 411

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	25,548 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-xtarget=ultra2 -x05	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	310 (275) / 313 (277)	309 (283) / 312 (286)
2 回目	310 (276) / 313 (278)	309 (283) / 312 (287)
3 回目	310 (276) / 314 (279)	309 (282) / 312 (285)

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	39,845 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-fast -arch ev6	
	処理クロック数 [clocks/block]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	289 (262) / 297 (276)	282 (275) / 296 (289)
2 回目	289 (262) / 297 (277)	282 (275) / 288 (289)
3 回目	289 (262) / 296 (276)	282 (275) / 288 (289)

表 3.62: SC2000 の鍵スケジュール部 + データランダム化部処理時間測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	23,700 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	/G6 /O2 /ML /W3 /GX	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	800 / 803	818 / 822
2 回目	800 / 803	818 / 821
3 回目	800 / 803	818 / 819

UltraSPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	22,524 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-xtarget=ultra2 -x05	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	623 / 627	618 / 622
2 回目	623 / 627	618 / 622
3 回目	623 / 627	618 / 622

Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	39,854 bytes (暗号化/復号/鍵スケジュール含む)	
コンパイラオプション	-fast -arch ev6	
	処理クロック数 [clocks]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	572 / 578	586 / 594
2 回目	572 / 578	586 / 595
3 回目	572 / 578	586 / 594

また、応募者からは以下のように自己評価が報告されている。なお、実装方法は、S-box の入力ビットをどのサイズとして実装したかを表す。そのほか、Pentium III あるいは Athlon におけるソフトウェアの高速実装が報告されている [8, 9, 10]。

プラットフォーム	:	Mobile Pentium III (1.2GHz), 128MB		
OS およびコンパイラ	:	Linux 2.4.18, Intel C Compiler 5.0		
使用言語	:	C		
実装方法	:	128-bit key	192-bit key	256-bit key
(16,16)	:	270 cycles/block	277 cycles/block	356 cycles/block
(11,10,11)	:	349 cycles/block	356 cycles/block	427 cycles/block
(6,10,10,6)	:	409 cycles/block	414 cycles/block	483 cycles/block
(6,5,5,5,5,6)	:	512 cycles/block	527 cycles/block	519 cycles/block

プラットフォーム	:	Athlon (1.4GHz), 1GB		
OS およびコンパイラ	:	Linux 2.4.17, Intel C Compiler 5.0		
使用言語	:	C		
実装方法	:	128-bit key	192-bit key	256-bit key
(16,16)	:	362 cycles/block	381 cycles/block	280 cycles/block
(11,10,11)	:	319 cycles/block	327 cycles/block	361 cycles/block
(6,10,10,6)	:	413 cycles/block	376 cycles/block	404 cycles/block
(6,5,5,5,5,6)	:	417 cycles/block	478 cycles/block	427 cycles/block

■IC カード実装 Z80 による IC カード実装評価を実施した。表 3.63 は、128 ビット鍵を利用したときの鍵スケジュール部 + データランダム化部処理時間測定結果である。

表 3.63: SC2000 の Z80 上での鍵スケジュール部 + データランダム化部処理時間測定結果

	ROM [bytes]	RAM [bytes]	Stack [bytes]	処理時間 [states]
暗号化	2,192	64	6	93,833
復号	2,192	64	6	94,263
暗復共用	2,350	—	—	—

また、設計者により IC カード用プロセッサでの実装が以下のように報告されている。

プロセッサ	暗号化 [msec/block]	復号 [msec/block]	鍵スケジュール [msec/key]	ROM [bytes]	RAM [bytes]
8051	8.113	8.609	21.666	1,597	294

3.3.10.5 ハードウェア実装評価結果

以下のブロック図 (図 3.22,3.23,3.24) に示すアーキテクチャで、FPGA 上で実装した結果 (表 3.64) を示す。なお、アルゴリズム実装上、評価基板の仕様に適合させるため、ラッチを挿入したため、見かけ上 clock 数が増えている。

Key Setup Clock 数	17
Data Randomize Clock 数	38
実装鍵ビット数	128

表 3.64: SC2000 の HW 実装評価結果

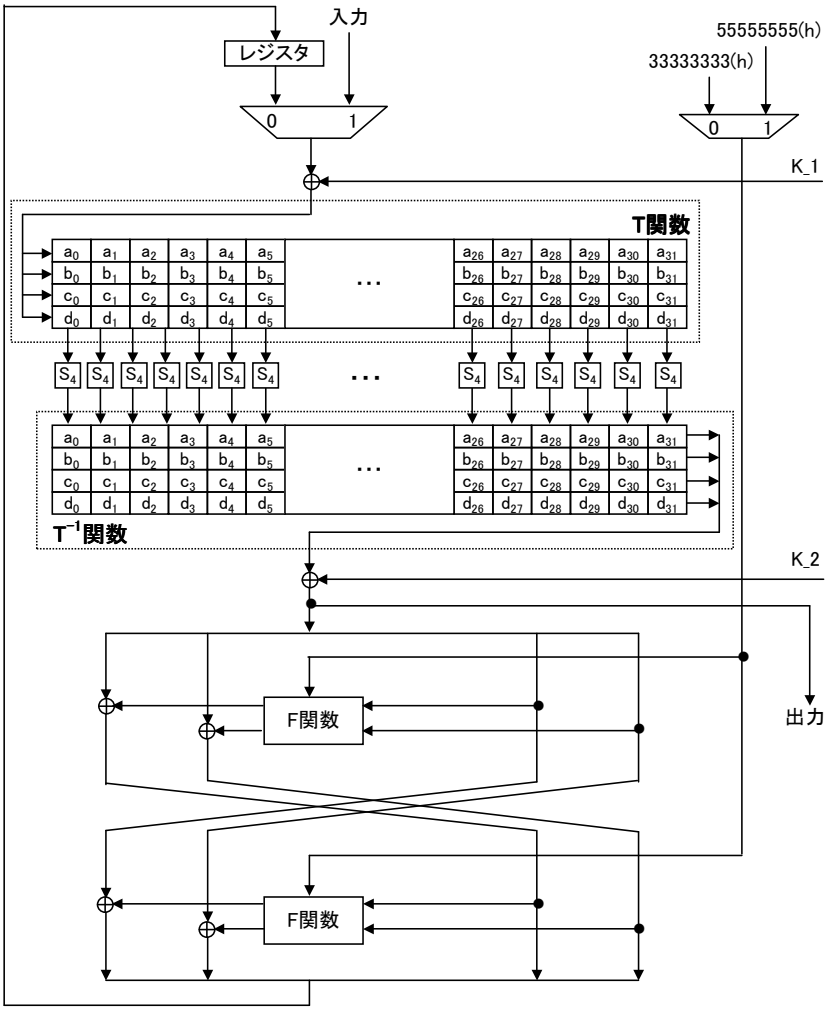

図 3.22: SC2000 暗号化回路ブロック図

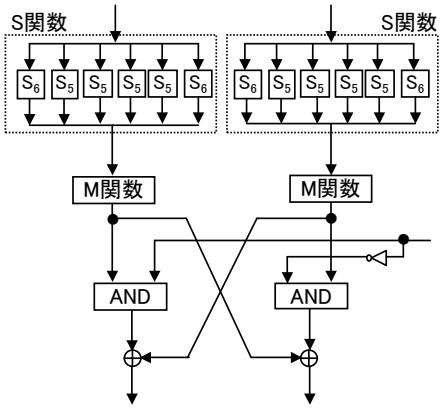

図 3.23: F 関数内部ブロック図

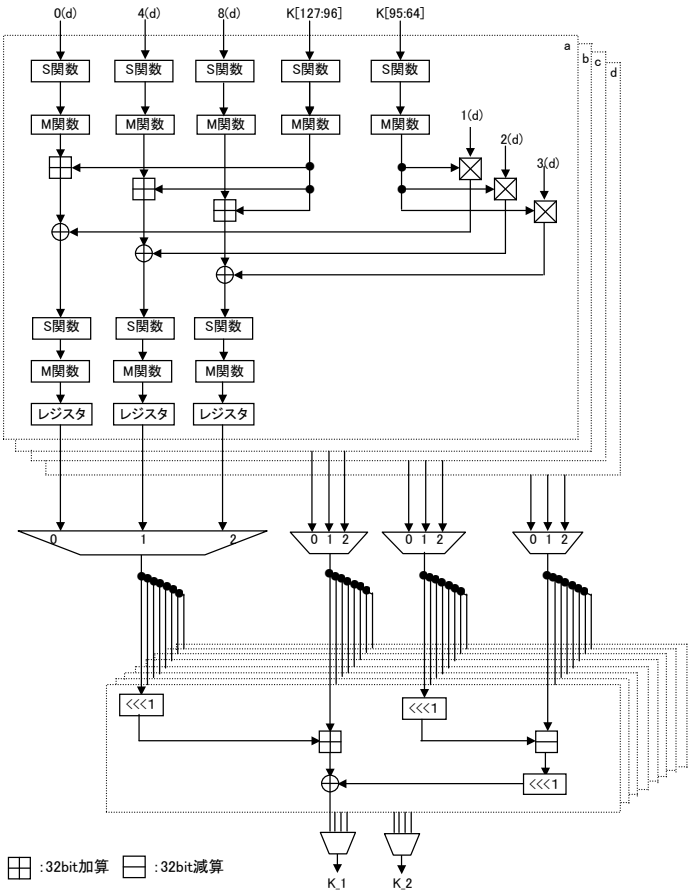

図 3.24: SC2000 鍵生成回路ブロック図

また、応募者からは、ASIC 実装に関して以下の自己評価が報告されている。なお、この実装は 128 ビット鍵専用である。

ASIC プロセス	:	0.18μm CMOS ASIC Design Library
速度優先実装	:	1,422.5 Mbps, 26.4 Kgates
規模優先実装	:	200.8 Mbps, 8.9 Kgates

参考文献

[1] 共通鍵ブロック暗号の選択／設計／評価に関するドキュメント, 通信・放送機構, 2000.

[2] 下山、屋並、横山、武仲、伊藤、矢嶋、鳥居、田中、「共通鍵ブロック暗号 SC2000」、信学技報 ISEC2000-72, 2000.

[3] 屋並、下山、「共通鍵ブロック暗号 SC2000 の差分/線形探索」、SCIS 2001, 12A-2, p.653, 2001.

[4] T. Shimoyama, H. Yanami, K. Yokoyama, M. Takenaka, K. Itoh, J. Yajima, N. Torii, and H. Tanaka, “Block Cipher SC2000,” FSE 2001, LNCS vol.2365, Springer p.312, 2002.

[5] 屋並、下山、「SC2000 の差分/線形探索 (II)」、信学技報 ISEC2001-10, 2001.

[6] 矢嶋、武仲、小柴、鳥居、「共通鍵ブロック暗号 SC2000 の乱数性」、信学技報 ISEC2001-11, 2001.

[7] H. Raddum and L. R. Knudsen, “A Differetial Attack on Reduced-Round SC2000,” SAC2001, LNCS Vol.2259, p.190, 2001.

[8] 武仲、岡田、矢嶋、鳥居、「共通鍵ブロック暗号 SC2000 の実装」、SCIS 2001, 13A-4, p.743, 2001.

[9] 武仲、岡田、矢嶋、鳥居、「共通鍵ブロック暗号 SC2000 の実装 (II)」、SCIS 2002, 9B-4, p.605, 2002.

[10] 武仲、鳥居、H. Lipmaa、「共通鍵ブロック暗号 SC2000 の実装 (III)」、信学技報 ISEC2002-39, 2002.

[11] H. Yamami, T. Shimoyama, and O. Dunkelman, “Differential and Linear Cryptanalysis of a Reduced-Round SC2000,” FSE2002, LNCS Vol.2365, p.34, 2002.

[12] 青木、山本、植田、盛合、「128 ビットブロック暗号に対するキャッシュ攻撃」、SCIS2003, 2D-4, 2003.

3.3.11　MUGI

3.3.11.1　技術概要

MUGI は株式会社日立製作所提案のストリーム暗号向け擬似乱数生成器であり、秘密鍵 128 ビット、初期ベクトル (公開値) 128 ビットをパラメータに持つ。

提案者の主張によると MUGI は、1998 年に Daemen と Clapp が提案した Panama を参考に設計されている。Panama は、擬似乱数生成器の設計法の 1 つである線形フィードバックシフトレジスタではなく、ブロック暗号と同じ原理に基づく設計を行っている。このため、ブロック暗号の設計、評価手法を適用しやすいと考えられる。また、基本的なアイデアが単純であり、同様の構造を持つバリエーションを設計しやすいことも特徴として挙げられる。しかし、一方で、Panama は従来にない設計を採用しており、安全性解析が困難であるという課題がある。このため、設計者は、MUGI の設計方針として、Panama と同様の構造を持ち、安全性が、既存のブロック暗号の解析手法がより適用しやすいような設計と評価を目指した、と主張している。

設計思想の 1 つに既存の評価された暗号技術を再利用することを掲げている。特に MUGI では、AES の構成要素 (例えば S-box) を採用している。

3.3.11.2　技術仕様

■おおまかな構造　MUGI は秘密鍵、初期ベクトルともに 128 ビット、出力ユニット長 n (自然数) を入力として持ち、n ユニットの乱数列を出力する。ただし、ここで言うユニットとは 64 ビットのデータブロックのことをいう。

■内部状態　MUGI の内部状態はステート、バッファと呼ばれる 2 つの部分で構成されている。時刻 t における各ユニットを以下のように表す。

- ステート a は 3 ユニットで構成され、上位からそれぞれ $a_0^{(t)}, a_1^{(t)}, a_2^{(t)}$ と表す。
- バッファ b は 16 ユニットで構成され、上位からそれぞれ $b_0^{(t)}, \ldots, b_{15}^{(t)}$ と表す。

時刻 t から時刻 $(t+1)$ へのこれらの変数の処理をラウンドと呼ぶ。

■全体構造　ρ 関数はステート a の状態遷移関数で、バッファ b の出力である $b_4^{(t)}, b_{10}^{(t)}$ を入力に持つ。F 関数は AES で使用されている S-box と MDS 行列による行列変換 M、さらにバイト置換を内部構造として持つ非線形関数である。λ 関数はバッファ b の状態遷移関数で、ステート a の一部である $a_0^{(t)}$ を入力に持つ線形関数である。

MUGI の状態遷移関数 $Update$ は ρ 関数と λ 関数の組み合わせで記述される:

$$(a^{(t+1)}, b^{(t+1)}) = Update(a^{(t)}, b^{(t)}) = (\rho(a^{(t)}, b^{(t)}), \lambda(b^{(t)}, a^{(t)})).$$

MUGI は初期化の完了後、MUGI 全体の状態遷移を繰り返しながら、ラウンド t において $a_2^{(t)}$ を乱数列 $Out[t]$ として出力する:

$$Out[t] = a_2^{(t)}.$$

■設計方針 MUGI は、ソフトウェア、ハードウェアいずれのプラットフォームにおいても高速な (もしくは軽量な) 実装が可能であることを目標として設計されたストリーム暗号向けの擬似乱数生成器である。MUGI は、Daemen と Clapp が提案した PANAMA[3] –これは擬似乱数生成器およびハッシュ関数用の暗号モジュールとして利用可能である– を参考にしている。PANAMA は、擬似乱数生成器の設計法として従来主流であった線形フィードバックシフトレジスタではなく、ブロック暗号と同じ原理に基づく設計を行っている。このため、ブロック暗号の設計、評価手法を適用しやすいと考えられる。また、基本的なアイデアが単純であり、同様の構造を持つバリエーションを設計しやすいことも特徴として挙げられる。一方で、PANAMA は従来にない設計を採用しており、PANAMA 自身の安全性評価も十分に行われているとは言い難い。このため MUGI は、PANAMA と同様の構造を持ち、かつ安全性をより評価できることを目指して設計されている。

3.3.11.3 安全性評価結果

■総評 2001 年度のスクリーニング評価結果では、安全性について問題は見つからなかったが、さらなる安全性および実装性の評価が必要と評価された。特に、2001 年度後半に、MUGI を含むある種のストリーム暗号系に対して、Coppersmith らにより、汎用的な Linear masking と呼ばれる解析手法が提案された [2]。しかし、この論文では、MUGI への攻撃可能性は示唆されているものの、詳細な検討・実際の解析は与えられていない。また、AES に対しては XL 攻撃 [1] が提案され、AES の S-box を利用している MUGI への XL 攻撃の影響も検討課題となる。さらに、内部で擬似乱数生成アルゴリズム PANAMA を使っていることから、MULTI-S01 との比較・優位性の検討が課題となった。このため 2002 年度は、この Coppersmith らの攻撃や XL 攻撃への耐性を含む、さらなる安全性評価と、MULTI-S01 との比較を 4 人の評価者 (No.0029, No.1012, No.1013, No.1014) に依頼した。

MUGI の基本設計の問題点を指摘する評価者 (No.1012, No.1013) がおり、また最新の攻撃に関しては、十分な評価が確定しているとは言い難いものの、どの評価者も、2^{128} よりも少ない計算量で秘密鍵を導出する攻撃法の存在は現時点でないとしており、今のところ、MUGI の安全性に致命的欠陥は見つかっていない。

■主な解読法ごとの安全性評価

差分攻撃・線形攻撃に対する耐性: SCIS2002 にて、提案者により、さらなる自己評価が発表された [4]。ここでは MUGI に対して、差分攻撃・線形攻撃を用いた再同期攻撃を適用することが検討され、主に ρ 関数の差分特性・線形特性の評価が行われている。この結果でも、線形攻撃に対する耐性を厳密に評価するためにさらなる解析の必要性を提案者自身示唆しているが、安全性

について問題があるという結果には至っていない。

また、他の評価者で差分攻撃・線形攻撃の適用可能性に関して論じている者 (No.1013, No.1014) も、現時点では攻撃に成功してはいない。

Linear masking 解析法に対する耐性: 評価者の一人 (No.0029) は、Coppersmith らの提案する Linear masking 解析法の MUGI への適用可能性を詳細に検討した。この際、非線形部である ρ 関数については Coppersmith と同様に distinguisher として線形近似式を取り上げ、線形部である λ 関数についてはこれを線形動的システムとして捉え解析している。その上で、64 ビットを一まとめとする truncated linear 解析により最大線形特性確率の上界を導出している。結論としては、本評価手法で示し得る active S-box 数の下限は 23 個となり、S-box の最大線形確率が 2^{-6} であることを考慮すると、示し得る最大線形特性確率の上界は 2^{-138} であり、2^{-128} を下回る。したがって、Coppersmith らの提案する攻撃に対して MUGI は十分な耐性を持つと結論付けている。

他の評価者 (No.1012, No.1013, No.1023) も Linear masking 解析法の MUGI への適用と耐性に関して、検討、議論しているが、いずれも現時点では攻撃に成功してはいない。

XL 攻撃に対する耐性: 評価者の一人 (No.1013) は、Courtois らの提案した AES への XL 攻撃を、MUGI に適用しその効率を論じている。しかし、その攻撃効率は 2^{128} を超え、解読成功にはいたっていない。

■統計的安全性評価 評価者の一人 (No.1014) は、MUGI の (鍵ストリームとしての) 統計的性質を解析している。ここでとりあげられている統計的性質としては周期・線形複雑度・頻度 (the frequency)・2 進変分 (binary derivative)・連分布 (runs distribution) 等、代表的な統計的性質であるが、これらの評価結果からは問題点は見つかっていない。

3.3.11.4 他のストリーム暗号 (MULTI-S01, Panama) との違いと比較

提案者の安全性解析をみると、MUGI は、Panama 以上に既存のブロック暗号の解析手法がより適用しやすい設計であり、この点は暗号設計評価上、重要な優位性と評価できる。

3.3.11.5 ソフトウェア実装評価

2002 年度に、以下の環境でソフトウェア実装評価を実施した。評価結果は表 3.65、表 3.66 の通りである。

備考 (カッコ) 内の値は応募者による測定プログラムの改変した場合の測定値。この改変は、計測途中における割り込み制御を軽減するためのものであり、速度評価の主旨を違えるような改変は行っていないことは確認済み。

応募者からは以下の自己評価結果が報告されている。

表 3.65: MUGI の暗号化・復号 (擬似乱数生成込) 処理速度測定結果

Pentium III (650MHz)		
言語	ANSI C	
コンパイラオプション	/G6 /ML /W3 /GX /O2	
	処理クロック数 [clocks/128bits]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	159 (161) / 193 (162)	161 (160) / 200 (161)
2 回目	162 (159) / 207 (160)	165 (159) / 217 (161)
3 回目	163 (161) / 193 (161)	160 (161) / 191 (163)

表 3.66: MUGI の鍵セットアップ処理時間測定結果

Pentium III (650MHz)		
言語	ANSI C	
コンパイラオプション	/G6 /ML /W3 /GX /O2	
	処理クロック数 [clocks/key]	
	暗号化鍵 (最速値 / 平均値)	復号鍵 (最速値 / 平均値)
1 回目	23,377 (20,755) / 52,739 (39,367)	20,363 (30,621) / 27,660 (38,154)
2 回目	28,388 (31,614) / 50,569 (49,390)	22,700 (24,582) / 25,189 (29,556)
3 回目	28,477 (19,052) / 48,942 (41,192)	25,771 (26,797) / 31,388 (30,180)

プラットフォーム	:	Pentium III (800MHz), 512MB
OS およびコンパイラ	:	Windows 2000, Visual C++ Ver 6.0
使用言語	:	ANSI C
鍵セットアップ	:	15,029 clocks/key
暗号化 (擬似乱数生成を含む)	:	21.8 clocks/byte

また、MUGI に関する高速実装に関する検討がなされていて、処理性能での改善が見られている [5]。

プラットフォーム	:	Pentium III (667MHz), 128MB
OS およびコンパイラ	:	Windows 2000, Visual C++ Ver 6.0
暗号化鍵セットアップ	:	9,967 clocks/key
復号鍵セットアップ	:	9,187 clocks/key
暗号化 (擬似乱数生成を含む)	:	6.5 clocks/byte
復号 (擬似乱数生成を含む)	:	6.5 clocks/byte

3.3.11.6 ハードウェア実装評価

また、応募者からは、ASIC 実装に関して以下の自己評価が報告されている。

ASIC プロセス	:	Hitachi 0.35μm CMOS ASIC Design Library
速度優先実装	:	2,922 Mbps(暗号化), 1,095 nsec(初期化), 26.1 Kgates
規模優先実装	:	676 Mbps(暗号化), 4,590 nsec(初期化), 18.0 Kgates

参考文献

[1] N. Courtois and J. Pieprzyk, “Cryptanalysis of block ciphers with overdefined systems of equations,” Cryptology ePrint Archive, IACR, 2002/044.

[2] D. Coppersmith, S. Halevi, and C. Jutla, “Cryptanalysis of stream ciphers with linear masking,” Cryptology ePrint Archive, IACR, 2002/20.

[3] J. Daemen, C. Clapp, “Fast Hashing and Stream Encryption with Panama,” *Fast Software Encryption, 5th International Workshop, FSE’98, Proceedings*, Springer-Verlag, LNCS 1372, pp. 60-74, 1998.

[4] 渡辺大、古屋聡一、吉田博隆、宝木和夫、「鍵ストリーム生成器 MUGI の安全性評価 (1)」、2002 年暗号と情報セキュリティシンポジウム、SCIS2002 講演予稿集、5B-4、2002.

[5] 吉田博隆、古屋聡一、「擬似乱数生成器のソフトウェア高速実装に関する考察」、2003 年暗号と情報セキュリティシンポジウム、SCIS2003 講演予稿集、9C-4、2003.

3.3.12 MULTI-S01

3.3.12.1 技術概要

MULTI-S01 は、2000 年に、ISEC 研究会において、株式会社日立製作所により提案された暗号技術である。MULTI-S01 は、擬似乱数生成器と暗号化・復号処理部の 2 つの部分から構成される。擬似乱数生成器は秘密鍵 K (256 ビット) から鍵ストリーム A, B, S を (処理するデータの長さに応じた長さだけ) 生成する。暗号化は、メッセージ M ($n \times 64$ ビット)、冗長符号 R (64 ビット)、秘密鍵 A ($\neq 0$、64 ビット)、秘密鍵 B ($(n+2) \times 64$ ビット)、秘密鍵 S (64 ビット) を入力として、暗号文 C ($(n+2) \times 64$ ビット) を出力する。復号は、暗号文 C ($64 \times n'$ ビット)、冗長符号 R (64 ビット)、秘密鍵 A ($\neq 0$、64 ビット)、秘密鍵 B ($64 \times n'$ ビット)、秘密鍵 S (64 ビット) を入力して、改ざん検出信号、またはメッセージ M ($64 \times (n'-2)$ ビット) を出力する。安全性については、メッセージ秘匿とメッセージ認証を同時に達成することと、現実的に攻撃の適用が困難となるような構成 (暗号解読の標的となる擬似乱数生成器の出力が一意に決められない) を目指した、としている。安全性は擬似乱数を発生する機構「擬似乱数生成器」の安全性に基づく。MULTI-S01 は擬似乱数生成器として Panama を用いている。

3.3.12.2 技術仕様

暗号化処理では、メッセージ M、冗長符号 R (64 ビット)、秘密鍵 K (256 ビット) をそれぞれバイト列によるデータ ($M(8)_i$ ($i = 1, \ldots, \lceil m/8 \rceil$), $R(8)_i$ ($i = 1, \ldots, 8$), $K(8)_i$ ($i = 1, \ldots, 32$)) として入力する。暗号化処理の出力は暗号文 C であり、C の長さは $64 \times (\lceil m/64 \rceil + 2)$ ビットで、バイト列として出力する。これに対応する復号処理では、暗号文 C (c ビット)、冗長符号 R (64 ビット)、秘密鍵 K (256 ビット) をそれぞれバイト列によるデータ ($C(8)_i$ ($i = 1, \ldots, \lceil c/8 \rceil$), $R(8)_i$ ($i = 1, \ldots, 8$), $K(8)_i$ ($i = 1, \ldots, 32$)) として入力する。復号処理の出力は復号結果 M' または改ざん検出信号であり、メッセージが出力される場合には、これをバイト列として出力する。暗号化処理・復号処理の内部は 64 ビットのブロックごとの処理で構成され、処理全体のブロックの数を $n = \lceil m/64 \rceil + 2$ とする。擬似乱数生成器は、K を入力として、A (64 ビット) と B ($64 \times (n+2)$ ビット)、S (64 ビット) を出力する。よって、暗号化処理部は M, R, A, B, S を入力として C を出力し、復号処理部は C, R, A, B, S を入力とし、復号結果 M' または改ざん検出信号を出力する。鍵、平文、暗号文、冗長符号、初期値はバイト単位の列として扱う。これらは 64 ビットのデータ型との変換の際、Big-Endian により変換される。

3.3.12.3 その他

MULTI-S01 が、ベースとして用いる技術に擬似乱数生成器 Panama [5] がある。Panama は、1998 年に J. Daemen と C. Clapp が提案した暗号モジュールであり、ストリーム暗号、およびハッシュ関数の構成方法として用いることができる。Panama は、計算量的に安全な擬似乱数生成器として提案されており、これまでの暗号学、共通鍵暗号技術、計算量理論、計算機科学、代数学、統計学などに基づいた設計が行われた、としている。なお MULTI-S01 で用いているのは

Panama の擬似乱数生成器の機能のみである。

標準化関連として、ISO/IEC 18033-4 (Committee Draft) に掲載されている。

3.3.12.4 安全性評価結果

■総評 ストリーム暗号としての安全性に関しては、現時点では学会等での厳密な評価が得られていないがおおむね安全である。システム設計時に、改ざん検出機能と鍵管理機能に関して注意を払えば、運用上の問題は少ないと思われる。

MULTI-S01 は、擬似乱数生成器 Panama と暗号化・復号処理部で構成されている。MULTI-S01 の安全性は、擬似乱数生成器 Panama の性質に帰着されることが示され、一方で Panama に関する安全性解析、統計検定の結果からは、致命的な欠点は見付かっていない。また、設計者らによって文献 [1] および [2] が発表されている。どちらの文献も、MULTI-S01 の評価であることを明示的には示しておらず、ストリーム暗号のスキームもしくは暗号利用モードの提案と評価である。しかし、両論文とも Panama を使用した実装例を記載しているので、MULTI-S01 の評価を含むと考えられる。文献 [1] は計算量的に安全な擬似乱数生成器、文献 [2] は、真の乱数生成器を用いた場合の、秘匿とメッセージ認証の安全性を考察しており妥当な評価と思われる。従って、ストリーム暗号 Panama が計算量的に安全な擬似乱数生成器または真の乱数生成器として扱える場合には、両文献のモデルに置き換えた上で MULTI-S01 の安全性の評価になると考えられる。

擬似乱数生成器からの出力系列の乱数性に関しては、長周期性、線形複雑度、相関値、0/1 等頻度性、連、一様性に関して詳細評価が行なわれ、特筆すべき問題点は報告されなかった。乱数系列の周期に関しては K, Q から決定されるため、十分な評価が行なわれていない。ただし、これは乱数系列に問題があるという積極的な指摘ではない。攪拌関数の入出力における相関性に関しては、擬似乱数生成器からの入力と暗号文出力との相関、およびメッセージ系列と暗号文出力の相関に関して詳細評価が行なわれ、特筆すべき問題点は報告されなかった。MULTI-S01 の攻撃評価では、Divide and Conquer Attack、相関攻撃、差分攻撃、線形攻撃が行なわれたが、大きな危険性は報告されていない。ただし、差分攻撃では、同一の鍵でメッセージを暗号化した場合の攻撃が報告されているが、鍵の管理を厳密に行えば回避できる。

■その他評価 仕様書では「乱数列番号 Q の使用法と注意」で、「平文の暗号化ではいつも必ず新しい (今まで装置が発生したことがない) 擬似乱数を使わなければならない。これは安全性に関する技術的理由によるものである。」とあるが、その「技術的理由」については述べられていない。特に、今まで装置が発生したことがない擬似乱数かどうかをどのように判定するのか不明である。前者の「技術的理由」が、同じ乱数列の重複使用をしてはならないことを意味するならば、ストリーム暗号一般に言えることであり、MULTI-S01 の問題とはならない。後者に関しては、「乱数列番号 Q が異なれば、同一の乱数列が発生することが無い」ことを示した報告が無いので不明となる。現在の統計評価が十分とは言いきれないが、安全性の問題が指摘されるような結果は出ていない。

以下、CRYPTREC で行なったこれらの安全性評価について各々の概要とその結果を述べる。

なお次節以降において、記号 “•” に続く記述は外部評価者によるレポート内での評価をまとめたものであり、“√” に続く記述はそれらを受けた委員会コメントである。

3.3.12.5 MULTI-S01 の暗号利用モードとしての安全性評価結果

MULTI-S01 装置内部の暗号部品である Panama の利用方法と安全性との関連について評価を行う。なお MULTI-S01 で用いているのは Panama の擬似乱数生成器の機能のみであるため、本評価では Panama の内部構造までは踏み込まない。

■MULTI-S01 仕様書について

- MULTI-S01 の仕様書は、一つのメッセージを暗号化する方法について述べているが、複数のメッセージを暗号化する方法について明確に述べていない (評価者 2)。

√ 評価者 2 は複数のメッセージの暗号化方法を提案している。提案暗号化方法は、MULTI-S01 の仕様書にある暗号化方法の自然な拡張になっている。

■MULTI-S01 の安全性の定義について

- 自己評価書の安全性の定義は、標準的な安全性の定義と比べると非常に弱い定義である。自己評価書では、privacy に関しては「一つの暗号文による暗号文単独攻撃を行う敵」に対する安全性を、authenticity に関しては「一つの平文/暗号文ペアからなる既知平文攻撃を行う敵」に対する安全性を証明している。これに対し、提案者は privacy および authenticity 両方に関して「適応的選択平文攻撃を行う敵」に対する安全性を証明しなければならない (評価者 1)。
- ストリーム暗号の authenticated encryption の安全性の定義は、一般的には知られていない。ストリーム暗号に適した安全性の定義を与える (評価者 2)。

√ 評価者 1 および評価者 2 ともに提案者が考えるべき安全性の定義を与えている。privacy に関する安全性の定義 (indistinguishability from random bits) は評価者 1 および評価者 2 で全く同一の定義となっている。authenticity に関しては評価者 1 は verification oracle に 1 回しか質問を許さないのに対し、評価者 2 は複数回の質問を許しており、評価者 2 の方が強い定義になっている。

■MULTI-S01 の安全性について

- 評価者 1 は MULTI-S01 が自ら提案した安全性の定義を満たすことの証明の指針、概要を与えている。評価者 1 の証明は未完成であり、提案者自身が証明を完成させるべきである、と述べている。
- 評価者 2 は MULTI-S01 が自ら提案した安全性の定義を満たすことを示している。

√ MULTI-S01 の計算量的安全性は評価者 2 によって示された。また SCIS2002 において、提案者による同様の結果が報告されている [7]。

■代替方式について

- Vernam 暗号と Carter-Wegman MAC を組み合わせれば MULTI-S01 と同様のストリーム暗号の authenticated encryption を作ることができ、この構成の方が、構造的に簡単で暗号文が短く、任意の長さのメッセージを扱うことができる (評価者 2)。

√ Vernam 暗号と Carter-Wegman MAC の組み合わせの場合と、MULTI-S01 とはそれぞれに優位点があると考えられるため、現状では比較が困難と考えられる。

■結論

√ 評価者 1 および評価者 2 は MULTI-S01 の安全性を Panama の安全性に帰着させる、という観点から、MULTI-S01 の計算量的安全性を評価している。MULTI-S01 の安全性を適切に定義しその観点のもとで評価した結果、本暗号の安全性は Panama の安全性に帰着可能であることが示された。

3.3.12.6 Panama の理論的暗号解析結果

MULTI-S01 の安全性は、Panama の影響が大きいことからモジュールとして利用されている Panama 自身の安全性を理論的に評価する必要がある。なお前節の結果から、MULTI-S01 の安全性は Panama に帰着されることが示されたことから、本節の結果は直接 MULTI-S01 の安全性に影響する。Panama は 1998 年に Daemen と Clapp によって提案された暗号アルゴリズム [5] であり、ハッシュ関数と擬似乱数生成器の 2 種類を含むが、MULTI-S01 で用いているのは擬似乱数生成器のみである。よって本評価では Panama の擬似乱数生成部の安全性のみ評価する。

本評価で実施された暗号解析は次の 5 項目である。

1. Chosen-IV Collision Attacks (評価者 1)
2. Chosen-IV Differential Attacks (評価者 1)
3. Chosen-IV Related-Key Attacks (評価者 1)
4. An Equivalent Representation (評価者 2)
5. Analysis of Simpler Variants of Panama (評価者 1, 評価者 2)

1 から 3 までの解析はいずれも Panama の Deviation Parameter と呼ばれる 256 ビットの値を、攻撃者が自由に選択できる場合を想定している。この値は Panama へ入力される公開情報であると考えられる。秘密鍵 (256 ビット) を K、Deviation Parameter を Q とし、Panama の出力する鍵ストリームを Panama(K, Q) とする。安全性の評価基準としては、あらゆる K について、Q を攻撃者がどのように定めたとしても Panama(K, Q) と 2 値の一様確率変数との区別が計算量的に困難となることが挙げられる。解析 4 は Panama のある等価表現を示し、その中で互いに独立な要素を示している。解析 5 は Panama に対する数種類の簡略化バージョンについて暗号解析を行なった結果である。

■Chosen-IV Collision Attacks

- ある K について、PANAMA(K, Q) = PANAMA(K, Q') となる (Q, Q') $(Q \neq Q')$ が存在するのならば、このような Q と Q' を選択することにより秘密鍵が K であるか否かを、ほぼ正確に判定できるが、PANAMA の構造を調査したところ、このような K は存在しないと分かった。しかし、Q がより長いビット長を持つ場合ははっきりしておらず、例えば、Q が 512 ビットの場合でも存在するかもしれない (評価者 1)。

√ 少なくとも任意に Q を長くとれる場合は、このような K が存在するはずである。また Rijmen et. al. の結果 [6] を見れば、K が与えられた元では、そのような (Q, Q') の組は現実的に見つけられるといえる。

■Chosen-IV Differential Attacks

- K と Q を Push (バッファへ入力) した直後の PANAMA の内部状態を M、Blank Pull を済ませて鍵ストリームを出力する直前の内部状態を M' とすると、Q, M, M' の差分に関して、高い確率で成立する式 (Differential Equation) が存在すれば、この式の計算を攻撃の手段として利用できるが、解析の結果高い確率を持つ Differential Equation は見つけられなかった。ただし Rijmen et. al. の結果からは、入力全体 (K, Q) の差分と M, M' の差分については、高い確率の Differential Equation が存在することを示している (評価者 1)。

■Chosen-IV Related-Key Attacks

- K と Q に任意の差分値を加えた時の鍵ストリーム PANAMA$(K \oplus \Delta K, Q \oplus \Delta Q)$ を得られるという仮定のもとでの攻撃が考えられる (評価者 1)。

√ この攻撃は、Q が 512 ビットであるときは Collision Attacks と関連するが、そうでない限りは非現実的な状況のもとでの攻撃である。

■An Equivalent Representation of PANAMA

- 時点 t におけるバッファの i 番目 $(i = 0, \ldots, 31)$ の段の j 番目 $(j = 1, \ldots, 8)$ の語 (ワード) の、k 番目 $(k = 1, \ldots, 32)$ のビットを、$b^i_{j,k}(t)$ で表す。j, k を省略した場合は、それぞれについてのベクトル表現であるとする。同様に、時点 t における State の j 番目 $(j = 0, \ldots, 17)$ の語 (ワード) の k 番目のビットを $a_{j,k}(t)$ で表す。
  バッファの 25 番目の更新が $b^{25}_j(t+1) = b^{24}_j(t) \oplus b^{31}_{j+2 \bmod 8}(t)$ となることから、バッファの更新だけを見ればビットの変化が影響する範囲は $\{b_0, b_2, b_4, b_6\}$ と $\{b_1, b_3, b_5, b_7\}$ とに分けられる。また、更新が語 (ワード) 単位であるので、ある段での k 番目のビットの変更は、他の段の k 番目以外のビットには影響しない。従って、PANAMA の構成を 64 個の部分構造に分解して表現することができる (評価者 2)。

√ バッファが上述の部分構造に分割できるため、Push モードではバッファ内の 1 ビットを変えても、その影響は各部分構造内に限定される。しかし Pull モードでは、バッファへの入力が State に依存するため、バッファ内の 1 ビットの影響が全体に及ぶことになる。

■Analysis of Simpler Variants of Panama

Panama を簡略化したものとして評価者 1 により Blank Pulls を省略した場合、評価者 2 により Panama-S1, Panama-S2, Panama-SM があげられているが、ここでは主に、最も重要と思われる Blank Pulls を省略した場合と Panama-S2, Panama-SM について説明する。

(1) Panama における Blank Pulls を省略した場合

- Blank Pulls を省略した場合は、K と Q を Push した後の内部状態のうち、State の一部である $a_8, \ldots, a_{16}$ は鍵ストリームの最初の 256 ビットに相当するが、これらは Q に依存せず決まる。従って、Q を変えた時、鍵ストリームの最初の 256 ビットが変化するか否かで区別すればよい。同様のことは、Blank Pull の回数が、正規の回数である 33 回よりも少ない場合には起きうる。Blank Pulls が 14 回以内の場合、適当な差分ベクトルを持つ入力ペアを用いて一様確率変数との区別が可能となる (評価者 1)。

√ この方法は Related-Key Attack に分類されるため、あまり現実的とは言えないものの K の分布が低いエントロピーを持つような場合には有効と思われる。ここで示したような解析が、正しく 33 回の Blank Pulls を行う場合にも有効であるかは不明である。

√ MULTI-S01 は、Blank Pulls を行うかどうかについて、何も言及していない。Reference Implementation では Blank Pulls を行っているため問題がないが、前述の Related-Key Attack を考えると、Blank Pulls を行うということを明記する必要がある。

(2) Panama-S2

- Panama-S2 とは、State の更新関数 $\rho = \sigma \circ \theta \circ \pi \circ \gamma$ の代わりに $\rho = \sigma \circ \pi$ という更新関数を用いるバージョンである。攻撃は、長さ n (語単位) の鍵ストリーム $[a_j(t)]_{j=9}^{16}, t = 1, \ldots, n$ を得たもとで、Pull を行っているある時点での、バッファと State の内容を推定することにより、以後の鍵ストリームを計算することが可能となる (評価者 2)。

√ 攻撃アルゴリズムは論理的には間違いがないと思われるが、攻撃アルゴリズムの計算量を、“Proportional to 2^{65}” であるとしているが、アルゴリズム中で探索する変数の数は $2^{4.7} \cdot 2^{25} = 2^{53}$ なので、トータルでは “Proportional to 2^{58}” が正しいと思われる。

(3) Panama-SM

- 最も Panama と似ているバージョンである Panama-SM では、State の更新関数を $\rho = \sigma \circ \theta^* \circ \pi \circ \gamma^*$ と置く。報告書では θ^* と γ^* が満たすべき条件を具体的に挙げているが、要は $a_j^*(t) \equiv \theta^* \circ \pi \circ \gamma^*(a_j(t))$ としたときに、$a_2^*(t), a_4^*(t), a_7^*(t), a_9^*(t), a_{11}^*(t), a_{12}^*(t), a_{14}^*(t), a_{16}^*(t)$ については鍵ストリームのみで計算可能であるという条件を満たしていればよい、という

ものである。この条件により、攻撃アルゴリズム自体は Panama-S2 となんら変わりなく実行することが可能となっている (評価者 2)。

■結論

- 結果として、一番シンプルな攻撃については安全であり、その他 2 つについても特に安全性が低いとは結論できていないが、これは評価にかける時間の少なさに依るところが大きい (評価者 1)。

√ Panama-S2 と Panama-SM では、State の更新における中間の計算結果の一部が、鍵ストリームより確定的に求まるという、Panama にはない性質を持つ。この性質自体が攻撃アルゴリズムに大きく寄与しているため、報告書で示された攻撃が直接的な脅威となることはないと思われる。しかし An Equivalent Representation で述べられているように、“Panama の構造を 64 個の部分構造へ分割して表現することができる” という性質があり、この性質を利用した攻撃が存在する可能性はある。

√ 実際の Panama では、$\theta \circ \pi \circ \gamma(a_j(t))$ が上記のような都合のよい条件を持つ訳ではないので、この攻撃アルゴリズム自体をわずかに修正 (例えば、探索すべき変数の数を増やすなど) することで実際の Panama へ攻撃できるとは考えられない。

√ 評価者 2 は Panama の改良案として、バッファの $b_j^{25}(t)$ の更新をより複雑にすることで、上記の性質を持たないようにすべきだとしている。これは妥当な案だが、バッファの更新を並列処理することが困難となるという、実装上の問題を同時に引き起こす可能性がある。

3.3.12.7 Panama の乱数性に関する統計検定結果

ここでは MULTI-S01 で使用されている Panama の統計的性質について解析を行う。すなわち Panama の内部構造には一切立ち入らず、Panama を擬似乱数生成器としてブラックボックス的に捉え、その出力系列の各種統計的性質について評価を行ない、ストリーム暗号 MULTI-S01 の構成要素として必要十分な性能を備えているかを検定した。擬似乱数性の検定法には NIST の SP 800-22 に付属する擬似乱数検定プログラムを用いて検証した (第 5.4.2 節参照)。

その結果、初期攪拌 32 回後では、入力の偏りによる出力乱数の系列間の偏りは全く見られないと考えられる。また、Panama 生成の出力系列と真性乱数系列とは多くの統計的性質において区別がつかないと判断される。

■擬似乱数生成器 Panama Panama は 256 ビットの秘密鍵 K と 256 ビットの乱数列番号 Q を入力とし、任意長の擬似乱数列を出力する。Panama は次の 3 つの動作モードを有する。

- reset モード: 内部状態のリセット
- push モード: 秘密鍵 K や乱数列番号 Q の入力
- pull モード: 初期攪拌と擬似乱数列の生成

■乱数検定 (実験結果)　評価方法としては、暗号アルゴリズムの特性を検討するために、暗号アルゴリズムの一部を使用して攪拌過程が検討可能な [Partial Round Test] と暗号アルゴリズム全体での擬似乱数性を検査する [Full Round Test] の 2 種類のテストを行った。

[Partial Round Test]

Panama を 256 ビットブロック暗号とみなし、秘密鍵を鍵、乱数列番号を平文、32 回の初期攪拌を Round と見なすことで Partial Round Test を行った。Panama では鍵と平文 (乱数列番号) の扱いが同じであることに注意して、Plaintext/Ciphertext Correlation と同様に鍵との相関を見る Key/Ciphertext Correlation を追加した。こうして秘密鍵や乱数列番号に偏りがある場合に出力系列間に偏りがあるかどうかを評価できる。その結果を Round 順に並べることで、入力された鍵や乱数列番号の攪拌されてゆく様子が観測できる。

[Full Round Test]

以下の手順で Full Round Test を実行した。

1. Panama に対して乱数で生成した秘密鍵と乱数列番号を入力し初期攪拌後の 314,572,800 ビットの擬似乱数列を生成
2. 1. で生成した擬似乱数列を $1,048,576 \times 300$ とみなし SP 800-22 で統計試験を実行
3. 1. および 2. を 128 回繰返し SP800-22 の「合格率」と「分布」を出力

ここで手順 3. を 128 回繰り返したのは信頼性向上のためである。

以上の統計検査の結果、Panama の擬似乱数性には特段の欠陥は見当たらないと判断される。

■結論　Panama の統計的性質に関して SP 800-22 を用いて評価を行った。Panama の初期攪拌における秘密鍵と乱数列番号の攪拌性を評価した結果、入力する秘密鍵や乱数列番号のデータの偏り、差分の偏り、および入力データとの相関のすべてについて、初期攪拌回数は 7 回程度で十分な攪拌が行われていることが判明した。そのために Panama 仕様の擬似乱数列が出力される初期攪拌回数 32 回では、入力の偏りによる出力乱数の系列間の偏りはまったく見られないと考えられる。

また Panama を用いて長い擬似乱数系列を出力した場合の統計的性質を評価した結果、その擬似乱数系列は数多くの統計的性質において真性乱数系列と区別がつかないと判断できる。以上 Panama に対する乱数検定からは、Panama の擬似乱数性に関して特段の欠陥は見当たらないと言える。

3.3.12.8　ソフトウェア実装評価結果

2002 年度に、以下の環境でソフトウェア実装評価を再実施した。評価結果は表 3.67、表 3.68 のとおりである。

備考　(カッコ) 内の値は応募者による測定プログラムの改変した場合の測定値。この改変は、計測途中における割り込み制御を軽減するためのものであり、速度評価の主旨を違えるような改変は行っていないことは確認済み。

表 3.67: MULTI-S01 の暗号化・復号 (擬似乱数生成込) 処理速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
コンパイラオプション	/G6 /ML /W3 /GX /O2	
	処理クロック数 [clocks/128bits]	
	暗号化 (最速値 / 平均値)	復号 (最速値 / 平均値)
1 回目	240 (239) / 283 (240)	229 (227) / 269 (229)
2 回目	240 (238) / 307 (240)	228 (226) / 285 (227)
3 回目	239 (239) / 290 (240)	227 (227) / 293 (228)

表 3.68: MULTI-S01 の鍵セットアップ処理時間測定結果

Pentium III (650MHz)		
言語	アセンブラ	
コンパイラオプション	/G6 /ML /W3 /GX /O2	
	処理クロック数 [clocks/key]	
	暗号化鍵 (最速値 / 平均値)	復号鍵 (最速値 / 平均値)
1 回目	7,088 (5,632) / 20,511 (20,675)	7,428 (5,433) / 10,258 (8,719)
2 回目	6,789 (5,532) / 20,921 (22,073)	8,213 (5,838) / 9,856 (8,378)
3 回目	5,649 (5,543) / 21,159 (20,655)	5,696 (7,339) / 9,516 (9,265)

また、応募者からは以下の自己評価結果が報告されている。

プラットフォーム	:	Alpha 21164A (600MHz), 512MB
OS およびコンパイラ	:	UNIX 4.0E, DEC cc
使用言語	:	C
鍵セットアップ (Panama の初期化を含む)	:	31,737 clocks/key
暗号化 (擬似乱数生成を含む)	:	17.7 clocks/byte
復号 (擬似乱数生成を含む)	:	18.0 clocks/byte

3.3.12.9 ハードウェア実装評価結果

また、応募者からは、ASIC 実装に関して以下の自己評価が報告されている。なお、処理回路には Panama (61.5 Kgates) が含まれている。

ASIC プロセス	:	Hitachi 0.35μm CMOS ASIC Design Library
速度優先実装	:	9,100 Mbps, 139.5 Kgates
規模優先実装	:	620 Mbps, 67.8 Kgates

参考文献

[1] S. Furuya, D. Watanabe, Y. Seto, and K. Takaragi, “Integrity-Aware Mode of Stream Cipher,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E85-A NO.1, pp. 55-65, January 2002.

[2] S. Furuya and K. Sakurai, “Single-path Authenticated-encryption Scheme Based on Universal Hashing,” in preproceedings of SAC 2002, Ninth Annual Workshop on Selected areas in cryptography, 2002, to appear in LNCS.

[3] 古屋聡一、高橋昌史、渡部大、宝木和夫、「疑似乱数生成器を使ったメッセージ認証可能な共通鍵暗号の提案」、電子情報通信学会技術研究報告 ISEC2000-8, 2000.

[4] 古屋聡一、渡部大、宝木和夫、「MULTI-S01 のパディングと安全性についての考察」、電子情報通信学会技術研究報告 ISEC2000-68, 2000.

[5] J. Daemen and C. Clapp, “Fast Hashing and Stream Encryption with Panama,” *Fast Software Encryption, 5th International Workshop, FSE’98, Proceedings*, Springer-Verlag, LNCS 1372, pp. 60-74, 1998.

[6] V. Rijmen, B. Rompay, B. Preneel, and J. Vandewalle, “Producing Collisions for PANAMA,” *Fast Software Encryption, FSE2001, Revised Papers*, Springer-Verlag, LNCS 2355, pp. 37-51, 1998.

[7] 古屋聡一、「MULTI-S01 の計算量的安全性」、2002 年暗号と情報セキュリティシンポジウム、SCIS2002 講演予稿集、5B-3、2002.

[8] NIST Special Publication 800-22, “A Statistical test suite for random and pseudorandom number generators for cryptographic applications,” (`http://csrc.nist.gov/rng/SP800-22.pdf`, `http://csrc.nist.gov/rng/errata2.pdf`)

[9] NIST Special Publication 800-22, “NIST Statistical Test Suite,” (`http://csrc.nist.gov/rng/sts-1.4.tar`, `http://csrc.nist.gov/rng/sts.data.tar`)

3.3.13 RC4 および Arcfour

3.3.13.1 技術概要

RC4 は 1987 年に RSA Data Security 社 (現 RSA Security 社) の Ron Rivest により開発された可変長な秘密鍵を利用できる、アルゴリズム非公開のストリーム暗号である。また、RC4 との相互通信が可能なストリーム暗号として Arcfour などが広く知られている。

RC4 の安全性評価を実施するにあたり、CRYPTREC は、RC4 がアルゴリズム非公開であることを考慮し、RSA セキュリティ株式会社との交渉を行った結果、RSA 社が “alleged RC4” と認めているアルゴリズム (文献 [1] など) が RC4 と等価なアルゴリズムであるとの確約を得ることができた。

そこで、文献 [1] に記載のアルゴリズムを CRYPTREC としては RC4 とみなして安全性評価を実施した。以下において、RC4 とは文献 [1] に記載のアルゴリズムを指すものとする。

RC4 の中核技術は、擬似乱数生成器であり、ワード長 n ビットと n から決定される状態数 2^n の状態表で規定され、時々刻々と状態表の内容を置換しながら、状態表の内容から擬似乱数を生成している。秘密鍵の役割は状態表における初期状態の決定である。RC4 は、SSL (Secure Socket Layer) プロトコルにも暗号化アルゴリズムの一つとして組み込まれ、無線 LAN で利用される WEP (Wired Equivalent Privacy) プロトコルにも組み込まれている。

3.3.13.2 技術仕様

RC4 は状態数 2^n である状態表を時々刻々と状態表の内容を置換していくことによって、状態表を変化させ、その内容を出力とする擬似乱数を利用するストリーム暗号である。一般に $n = 8$ とする、状態数 256 の RC4 が用いられる。このときの各内部状態を $S_0, S_1, \ldots, S_{255}$ とする。$i = j = 0$ として、

$$
\begin{aligned}
i &= (i + 1) \bmod 256 \\
j &= (j + S_i) \bmod 256 \\
&\text{swap } S_i \text{ and } S_j \\
t &= (S_i + S_j) \bmod 256
\end{aligned}
$$

を繰り返し、各時点での S_t を 1 バイトの擬似乱数として出力する。

また、内部状態の初期状態は ℓ ビットの鍵 ($40 \le \ell \le 256$) によって決められる。この鍵を 1 バイトブロックに分割してそれぞれを $K_0, K_1, \ldots, K_{\lceil \ell/8 \rceil - 1}$ とし、$S_x = x$ ($x = 0, 1, \ldots, 255$), $t = 0$ として、

$$
\begin{aligned}
i &= (i + 1) \bmod 256 \\
j &= (j + S_i + K_t) \bmod 256 \\
&\text{swap } S_i \text{ and } S_j \\
t &= (t + 1) \bmod \lceil \ell/8 \rceil
\end{aligned}
$$

によって初期値が定まる。なお、SSL3.0/TLS1.0 においては、40 ビットあるいは 128 ビットの秘密鍵 ($\ell = 40$ or 128) が用いられる。

3.3.13.3 その他

標準化関連として、ISO/IEC 9979 アルゴリズム非公開登録、SSL3.0 および RFC 2246: TLS1.0 (Proposed Standard) に掲載されている。

3.3.13.4 安全性評価結果

■総評 標準仕様、すなわちワード長 $n = 8$、状態数 256 の RC4 (および Arcfour) については、現在のところ、現実的な解読法は提案されていない。しかし、秘密鍵から生成される初期状態によっては、かならずしも安全ではないという報告がなされている。したがって、RC4 の利用に関してその初期状態の決定には注意が必要である。

また SSL3.0/TLS1.0 での利用に関しては、現在のところその安全性に関して欠陥は報告されていない。ただし、40 ビットの秘密鍵によって初期状態を生成する 40-bit RC4 は、鍵の推定が可能であることから安全ではない。

■評価概要 128 ビット鍵を用いるワード長 8 ビット、状態数 256 の RC4 について、現実的な解読法は発表されていない。しかし、RC4 の秘密鍵の設定や運用方法にいくつかの注意が必要である。

特に WEP での利用に関しては文献 [1, 2] において安全性に関する欠点が指摘されている。RC4 での擬似乱数出力は、短い経過時間、すなわち十分な初期状態の攪拌が行われない場合、その初期状態との間で、大きな相関が見られ、特にその初期状態が鍵によって十分な攪拌が行われていない場合、鍵そのものとの相関が現れ、鍵を解読される可能性がある。鍵による初期状態の決定には十分な攪拌が必要である。WEP ではユーザの秘密鍵と初期化ベクトル (IV) を結合させた鍵をセッション鍵として利用している。したがって互いのセッション鍵に強い相関があれば、RC4 の出力系列においても強い相関を有することを利用して 128 ビットの秘密鍵を推定可能である。しかしながら、この欠点は IV を MD5 などのハッシュ関数を通したハッシュ値を利用することにより、解決すると RSA 社からコメントが公表されている [3, 4]。なお、SSL での RC4 利用においては、その安全性の評価が行われており [5]、問題点は指摘されていない。

RC4 で出力される擬似乱数系列の統計的性質に関する研究もなされている [6]。その結果、ワード長が小さい場合の RC4 に関しては統計的性質が悪いという結果を与え、ワード長が 8 以上である必要性を強調している。また文献 [7, 8] では、真の乱数系列と RC4 で生成される擬似乱数系列との判別について議論され、$n = 8$ において 2^{30} ワード (バイト) の出力で判別可能としている。また文献 [6] および [9] では、出力の 1 バイト目と 2 バイト目の出力に統計的な偏りが存在することを示している。

秘密鍵によって初期状態を決定し、その初期状態から擬似乱数出力が決定されることから、初期状態を推定する研究も存在する。文献 [10] では $n = 8$ として、1 ワードの擬似乱数出力を生成し、内部状態を更新する計算量を一単位としたとき、2^{30} の計算量で、256 個の内部状態の変数のうち、100 個が既知であれば、残りの内部状態の変数を推定可能であると述べている。文献

[11] では、出力の相関の利用および内部状態の変数間の関係を用いることによって、高々 73 個の内部状態の変数が既知であれば、2^{20} の計算量で残りの内部状態の変数を推定可能である内部状態が無視できない確率で存在することを示している。さらに文献 [12] では、57 個の内部状態の変数が既知であれば、残りの内部状態の変数を完全に推定し得る初期状態が存在することを示している。

文献 [13] では補間攻撃を行なうために、RC4 の内部状態変数 S_x を Z/256Z の加法群から GF(257) の乗法群への同型変換を行ない、擬似乱数生成部の書き換え規則を GF(257) 上の多項式で表現する方法を示している。しかしながら現在のところ、補間攻撃に要する連立方程式を解くことは効率的ではないと考えられている。

参考文献

[1] S. Fluhrer, I. Mantin, and A. Shamir, “Attacks On RC4 and WEP,” CryptoBytes Volume 5, No. 2, Summer/Fall 2002, available at `http://www.rsasecurity.com/rsalabs/cryptobytes/index.html`.

[2] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the key scheduling algorithm of RC4,” Eighth Annual Workshop on Selected Areas in Cryptography, Aug. 2001.

[3] RSA Laboratories Tech Notes, “RSA Security Response to Weaknesses in Key Scheduling Algorithm of RC4,” available at `http://www.rsasecurity.com/rsalabs/technotes/wep.html`, Sep. 2001.

[4] RSA Laboratories Tech Notes, “WEP Fix using RC4 Fast Packet Keying,” available at `http://www.rsasecurity.com/rsalabs/technotes/wep-fix.html`, Dec. 2001.

[5] H. Krawczyk, “The order of encryption and authentication for protecting communications (or: How secure is SSL),” CRYPTO2001, LNCS2139, pp.310–331, 2001.

[6] I. Mantin and A. Shamir, “A Practical Attack on Broadcast RC4,” Proceedings FSE 2000, LNCS 2355, pp.152–154, 2002.

[7] Jovan Dj. Golic, “Linear Statistical Weakness of Alleged RC4 Keystream Generator,” Proceedings of EUROCRYPT’97, Lecture Notes in Computer Science, vol.1233, W.Fumy ed., pp.226–238, 1997.

[8] S. R. Fluhrer and D. A. MeGrew, “Statistical Analysis of Alleged RC4 Key Stream Generator,” Proceedings FSE 2000, LNCS 1978, pp.19–30, 2001.

[9] I. Mironov, “(Not So) Random Shuffles of RC4,” Proceedings of CRYPTO2002, LNCS 2442, pp.304–319, 2002

[10] L. Kundsen, W. Meier, B. Preneel, V. Rijmen, and S. Verdoolaege, “Analysis methods for (alleged) RC4,” Advances in Cryptology - ASIACRYPT’98, LNCS 1514, Springer-Verlag, pp.327–341, 1998.

[11] 大東俊博、白石善明、森井昌克、「RC4 の効率的な内部状態推定法」、第 25 回情報理論とその応用シンポジウム (SITA2002) 予稿集, Vol.2, pp.607–610, 2002.

[12] 大東俊博、白石善明、森井昌克、「RC4 の内部状態推定法に関する考察」、SCIS2003, 6-D1,

pp.447–452, 2003.

[13] 下山武司、「RC4 の多項式表現と補間攻撃への応用」、SCIS2003, 5-D4, pp.369–374, 2003.

第 4 章

ハッシュ関数の評価

4.1　評価方法総評

ハッシュ関数は、ほぼ任意のビット長 m の入力メッセージを一定ビット長 n のメッセージダイジェスト (ハッシュ値) に圧縮する関数のことである。特に、「一方向性」と「無衝突性」を満たすハッシュ関数のことを暗号学的ハッシュ関数とも言う。

一方向性とは、出力ハッシュ値から入力メッセージを簡単に計算できない性質を意味する。衝突とは、異なる 2 つの入力メッセージに対し同じハッシュ値を出力することである。ハッシュ関数は出力ビット長 n よりも入力ビット長 m の方が大きいことを許しているため、完全に無衝突であることはあり得ない。そこで、現実的な計算量で衝突が発見されなかった場合、無衝突性を持つと判断する。

今回は、ハッシュ関数の応募はなかったため、広く使用されていると判断された技術について、文献等の調査によりその安全性評価を実施した。

■評価内容　暗号学的ハッシュ関数が満たすべき性質である、一方向性および無衝突性の検討を行う。また、各ハッシュ関数に特化した新たな攻撃による評価、SHA-1 や MD 型ハッシュ関数との安全性の観点からの比較、発表された攻撃結果などの文献調査も行う。

4.2　評価結果

評価対象は、RIPEMD-160、SHA-1、SHA-256、SHA-384、SHA-512 の 5 種類である。

■安全性評価の総評　すべての評価対象ハッシュ関数に対して、実用的な安全性を脅かす攻撃方法は報告されていないため、これらのハッシュ関数は暗号の応用分野で使うのに十分安全であると考えられる。しかしながら、ハッシュ値の全数探索や誕生日攻撃による衝突からの安全性を確保するためには、ハッシュ値が 160 ビット以上必要である。

このため、より長いハッシュ値のものを採用することができるのであれば、ハッシュ値が 256

表 4.1: 各ハッシュ関数の特徴

アルゴリズム	RIPEMD -160	SHA-1	SHA-256	SHA-384	SHA-512
入力可能メッセージ長 [bits]	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
出力ハッシュ長 [bits]	160	160	256	384	512
基本処理単位ブロック長 [bits]	512	512	512	1,024	1,024
基本演算処理ワード長 [bits]	32	32	32	64	64
処理ステップ数	2×80	80	64	80	80

ビット以上となるハッシュ関数を選択することが望ましい。ただし、例えば、公開鍵暗号の仕様上利用すべきハッシュ関数が指定されている場合やインターオペラビリティの必要性が生じる場合はこの限りではない。

4.3 個別暗号技術の評価

4.3.1 RIPEMD-160

4.3.1.1 技術概要

RIPEMD-160 は Dobbertin、Bosselaers、Preneel により提案されたハッシュ関数であり、ヨーロッパの RIPE (Race Integrity Primitive Evaluation) プロジェクトの成果の一つである。その後、SHA-1 や RIPEMD-128 などとともに ISO/IEC 10118-3 の国際規格にも採用されている [1]。RIPEMD-160 はビット長が 512 ビットの倍数になるようにパディングされた任意のメッセージを入力として 160 ビットのハッシュ値を出力する。

4.3.1.2 技術仕様

RIPEMD-160 は MD4 や MD5 を改良する形で設計されたが、MD4 同様に 32 ビット計算機において高速処理が可能となるように、32 ビットの算術加算、論理演算、巡回シフト命令などを主要演算として用いて構成されている。RIPEMD-160 は入力・圧縮・出力の 3 つの部分で構成される。RIPEMD-160 は 2 つのほぼ同じ形をした関数を並列で走らせて任意長のメッセージから 160 ビットのハッシュ値を出力する。2 つの関数は右ラインおよび左ラインと呼ばれ、各々 5 ラウンド 80 ステップで構成される。RIPEMD-160 の詳細仕様については [1] を参照。

(1) 入力

入力メッセージは Little-Endian 方式により 32 ビット整数に変換され、512 ビットのブロックに分けられる。16 個の 32 ビット入力 $X[0] \sim X[15]$ は定められた順番によって右ラインと左ラインに入力される。

(2) 圧縮関数

圧縮関数の計算には 5 つの連鎖変数 (A, B, C, D, E) を用いる。A, B, C, D の初期値は MD5 と同じ値であり、新しく E の初期値が定められている。(A, B, C, D, E) の初期値 $IV = (h_1, h_2, h_3, h_4, h_5)$ を以下に示す。

$$h_1 = \texttt{0x67452301}$$
$$h_2 = \texttt{0xefcdab89}$$
$$h_3 = \texttt{0x98badcfe}$$
$$h_4 = \texttt{0x10325476}$$
$$h_5 = \texttt{0xc3d2e1f0}$$

この初期値は左右両ラインで共通に用いられる。また、圧縮関数では次に示す 5 つのブール

関数を用いる。

$$
\begin{aligned}
f(x,y,z) &= x \oplus y \oplus z \\
g(x,y,z) &= (x \wedge y) \vee (\overline{x} \wedge z) \\
h(x,y,z) &= (x \wedge \overline{y}) \oplus z \\
k(x,y,z) &= (x \wedge y) \vee (y \wedge \overline{z}) \\
l(x,y,z) &= x \oplus (y \vee \overline{z})
\end{aligned}
$$

ここで、記号 $\wedge, \vee, \oplus$ はそれぞれビットごとの論理積、論理和、排他的論理和を表し、$\overline{x}$ は x のビット反転を表す。

RIPEMD-160 の圧縮関数を構成するステップ関数は次のとおりである。ここで、変数への添え字 R は右ラインの、L は左ラインに関する変数であることを示す。RIPEMD-160 は右ラインと左ラインを並列に実行することでハッシュを行う。ステップ関数で用いられる定数 $K_L[j], K_R[j]$ は次のように与えられる。

$$
\begin{aligned}
&K_L[j] = \texttt{0x00000000}, \quad K_R[j] = \texttt{0x50a28be6}, \quad (1 \le j \le 16) \\
&K_L[j] = \texttt{0x5a827999}, \quad K_R[j] = \texttt{0x5c4dd124}, \quad (17 \le j \le 32) \\
&K_L[j] = \texttt{0x6ed9eba1}, \quad K_R[j] = \texttt{0x6d703ef3}, \quad (33 \le j \le 48) \\
&K_L[j] = \texttt{0x8f1bbcdc}, \quad K_R[j] = \texttt{0x7a6d76e9}, \quad (49 \le j \le 64) \\
&K_L[j] = \texttt{0xa953fd4e}, \quad K_R[j] = \texttt{0x00000000}, \quad (65 \le j \le 80)
\end{aligned}
$$

また、ステップ関数で用いられる左巡回シフト量 $s_L[j], s_R[j]$ はあらかじめ定められている。RIPEMD-160 のステップ関数は次のとおりである。ただし、記号 $X^{<<s}$ により、変数 X を s ビット左巡回シフトする演算を表すものとする。

1 ラウンド $(1 \le j \le 16)$

$$
\begin{aligned}
&FF_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) : \\
&\qquad A_L = (A_L + f(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10} \\
&LL_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) : \\
&\qquad A_R = (A_R + l(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}
\end{aligned}
$$

2 ラウンド $(17 \le j \le 32)$

$$
\begin{aligned}
&GG_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) : \\
&\qquad A_L = (A_L + g(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10} \\
&KK_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) : \\
&\qquad A_R = (A_R + k(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}
\end{aligned}
$$

3 ラウンド $(33 \le j \le 48)$

$$
\begin{aligned}
&HH_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) : \\
&\qquad A_L = (A_L + h(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10} \\
&HH_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) : \\
&\qquad A_R = (A_R + h(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}
\end{aligned}
$$

4 ラウンド ($49 \leq j \leq 64$)

$$KK_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]):$$
$$A_L = (A_L + k(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10}$$
$$GG_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]):$$
$$A_R = (A_R + g(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}$$

5 ラウンド

$$LL_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]):$$
$$A_L = (A_L + l(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10}$$
$$FF_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]):$$
$$A_R = (A_R + f(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}$$

(3) 出力

出力は基本的に MD5 同様、最後の段階で求められた連鎖変数の値と初期値 IV を加えた後、(A, B, C, D, E) の 5 つの変数を結合することでハッシュ値を出力するが、2 つのラインを使うため以下のように計算する。

$$A = h_2 + C_L + D_R$$
$$B = h_3 + D_L + E_R$$
$$C = h_4 + E_L + A_R$$
$$D = h_5 + A_L + B_R$$
$$E = h_1 + B_L + C_R$$

4.3.1.3 その他

1995 年に RIPE プロジェクトに提案された RIPEMD について、処理ラウンドを減らした場合には衝突 (Collision) を見つけることができることがわかったため [3]、その改良方式として RIPEMD-160 が提案された [2]。その後、RIPEMD の構成要素について、より強力な攻撃法も発見されている [4]。RIPEMD-160 は MD4 をベースとするハッシュ関数のひとつである。

標準化関連として、ISO/IEC 10118-3 に掲載されている。

4.3.1.4 安全性評価結果

ハッシュ関数の安全性については大きく二つの観点から安全性が評価される。一つ目の指標は特定の出力に対応する入力値を発見する手間、すなわち (1) 入力値 (Preimage) 探索の手間、である。二つ目の指標は出力値が一致するような異なる入力値を発見する手間、すなわち (2) 衝突 (Collision) 発見の手間、である。RIPEMD-160 について (1)、(2) への耐性共に現状では充分であると考えられる。以下、補足説明する。

■RIPEMD-160 固有の攻撃　RIPEMD-160 の前身である RIPEMD に対しては、最初のラウンドまたは最後のラウンドを省略した場合、2^{31} 以内の計算量で衝突を発見できることが報告さ

れている [3]。この結果に基づいて RIPEMD-160 においては、ラウンド数を 5 段に拡張し、左右 2 つの並列ライン間の独立性を向上させることで安全性が高められている。RIPEMD に対する攻撃は RIPEMD-160 には適用できず、RIPEMD-160 に対する固有の攻撃はまだ報告されていない。

■入力値探索の手間　ハッシュ値の長さが n ビットのハッシュ関数が出力する値のパターンは 2^n 通りしか存在しない。従って、特定のハッシュ値を出力する入力値を探索しようとした場合、異なるハッシュ値を出力する 2^n 通りの入力値をあらかじめ用意しておけば指定されたハッシュ値に一致する入力値を得ることができる。RIPEMD-160 では $n = 160$ であり、この方法を適用した場合 2^{160} 通りの入力値が必要となるが、これは現在の技術では用意不可能なほど大きい数であると考えられている。

■衝突発見の手間　ハッシュ値の長さが n ビットの場合、誕生日攻撃という一般的な解析手法により $2^{n/2}$ 個の入力値を用意すると、その中に比較的高い確率でハッシュ値の一致するペアを見出すことができ、これを防止するためには、ハッシュ値を充分に長くする必要がある。RIPEMD-160 は $n = 160$ であるため、2^{80} 個程度のメッセージを用意することができれば誕生日攻撃が適用できるが、これだけの入力値を用意することは現時点では現実的でないと考えられている。

4.3.1.5　ソフトウェア実装評価

CRYPTREC ではソフトウェア実装評価はしていないが、2002 年度報告書作成時において [5] に次のような実装結果が示されている。

プラットフォーム	:	Celeron (850MHz)
OS および使用言語	:	Windows 2000 SP1, C++
処理性能	:	30.725 Mbps

また、ハッシュ関数の高速実装に関する最近の研究論文 [6] では、以下のような結果が報告されている。

プラットフォーム	:	Pentium III (800MHz)
OS および使用言語	:	Windows98, Visual C++ 6.0 および MASM6.15
処理性能例	:	564.4 Mbps (11.34 cycles/byte)

4.3.1.6　ハードウェア実装評価

ハードウェア実装時の速度および回路規模については評価していない。

参考文献

[1] ISO/IEC 10118-3, Information technology – Security techniqes – Hash-functions – Part3: Dedicated hash-functions

[2] H. Dobbertin, A. Bosselaers, B. Preneel, RIPEMD-160: A strengthened version of RIPEMD, Fast Software Encryption – Cambridge Workshop, LNCS vol.1039, Springer-Verlag, pp.71-82, 1996.

[3] H. Dobbertin, RIPEMD with two-round compress function is not collision-free, Journal of Cryptology 10 (1): pp51-70, 1997.

[4] C. Debaert, H. Gibert, The RIPEMD-L and RIPEMD-R improved variants of MD4 are not collision free, Fast Software Encryption, LNCS vol.2355, Springer-Verlag, 2001.

[5] `http://www.eskimo.com/~weidai/benchmarks.html`

[6] J. Nakajima, and M. Matsui, Performance Analysis and Parallel Implementation of Dedicated Hash Functions on Pentium III, IEICE Trans. Fundamentals, Vol.E86-A, No.1, pp.54-63, 2003.

4.3.2 SHA-1 / SHA-256 / SHA-384 / SHA-512

4.3.2.1 技術概要

NIST は 1992 年に 160 ビットハッシュ関数 SHA と 1994 年にその改訂版の SHA-1 を FIPS PUB 180-1 で提案している [1]。さらに、NIST は SHA-1 とほぼ同様な設計原理に基づいているが、SHA-1 よりもハッシュ値が長い 256 ビットハッシュ関数 SHA-256、384 ビットハッシュ関数 SHA-384、512 ビットハッシュ関数 SHA-512 の三種類を提案し、SHA-1 とともに 4 種類をすべて NIST の標準ハッシュ関数 Secure Hash Standard (SHS) として規定している (2002 年 8 月 1 日付) [2]。 新しい標準ハッシュ関数として三種類のハッシュ関数 SHA-256、SHA-384、SHA-512 を導入した最大の理由は、これらのハッシュ関数に対する誕生日攻撃のセキュリテイレベルは各々 2^{128}、2^{192}、2^{256} であるので、最近採用されたブロック暗号 AES の鍵長 128、192、256 ビットと対応させるためである。

上記 4 種類のハッシュ関数の諸特性は以下の表でまとめられる。なお、セキュリテイの数値は誕生日攻撃に対する強度を表す。

より長いハッシュ値のものを採用することができるのであれば、ハッシュ値が 256 ビット以上となるハッシュ関数を選択することが望ましい。ただし、例えば、公開鍵暗号の仕様上利用すべきハッシュ関数が指定されている場合やインターオペラビリティの必要性が生じる場合はこの限りではない。

表 4.2: SHA の特徴

アルゴリズム	SHA-1	SHA-256	SHA-384	SHA-512
入力可能メッセージ長 [bits]	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
出力ハッシュ長 [bits]	160	256	384	512
基本処理単位ブロック長 [bits]	512	512	1,024	1,024
基本演算処理ワード長 [bits]	32	32	64	64
処理ステップ数	80	64	80	80
セキュリティ	2^{80}	2^{128}	2^{192}	2^{256}

4.3.2.2 SHA-1 の技術仕様

MD4、MD5 と同様な設計である。SHA-1 のアルゴリズム[*1]は、以下の (I) の関数を用いた、(II) の前処理と (III) のハッシュ値計算の 2 段階からなる。

(I) SHA-1 で使用される関数

[*1] FIPS PUB 180-2 の SHA-1 は、FIPS PUB 180-1 で規定された SHA-1 とまったく同じである。ただし、FIPS PUB 180-1 で用いた n ビット左巡回シフトする演算記号 $S^n(x)$ は FIPS PUB 180-2 では $\mathrm{ROTL}^n(x)$ に変更されている。また、SHA-256、SHA-384、SHA-512 の記述との整合性のため幾つかの記号を変更している。

SHA-1 では、32 ビットワード (word) を入力変数および出力変数とする、以下の論理関数 f_0, $f_1, \cdots, f_{79}$ が用いられる。

$$f_t(x,y,z) = \begin{cases} \mathrm{Ch}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) & 0 \le t \le 19 \\ \mathrm{Parity}(x,y,z) = x \oplus y \oplus z & 20 \le t \le 39 \\ \mathrm{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) & 40 \le t \le 59 \\ \mathrm{Parity}(x,y,z) = x \oplus y \oplus z & 60 \le t \le 79 \end{cases}$$

ここで、記号 $\wedge, \oplus$ はそれぞれビットごとの論理積、排他的論理和を表し、$\neg x$ は x のビット反転を表す。

(II) SHA-1 の前処理

(i) 入力メッセージ M について、メッセージ長が 512 ビットの倍数になるように初期パディングされたメッセージ

$$M||1||0^k||\ell$$

を計算する。ただし、ℓ は M のメッセージ長をバイナリ表現した時のビット数、k は $\ell + 1 + k \equiv 448 \pmod{512}$ を満たす最小値である。

(ii) 初期パディングされたメッセージは N 個の 512 ビット単位のブロック (block) $\{M^{(i)}\}_{i=1}^{N}$ に分割される。ただし、各々の $M^{(i)}$ は、16 個の 32 ビット長のワード

$$M^{(i)} = M_0^{(i)}||M_1^{(i)}||\cdots||M_{15}^{(i)}$$

からなる。

(iii) 初期ハッシュ値の j 番目のワード $H_j^{(0)}$ $(0 \le j \le 4)$

$$\begin{aligned} H_0^{(0)} &= \texttt{67452301} \\ H_1^{(0)} &= \texttt{efcdab89} \\ H_2^{(0)} &= \texttt{98badcfe} \\ H_3^{(0)} &= \texttt{10325476} \\ H_4^{(0)} &= \texttt{c3d2e1f0} \end{aligned}$$

を設定する。

(III) SHA-1 のハッシュ値計算

N 個のメッセージブロック $M^{(1)}, \cdots, M^{(N)}$ の $M^{(i)}$ に対して、$1 \le i \le N$ の順に以下のことを実行する。

(i) 次式で定義する SHA-1 メッセージスケジュール関数を用いて拡張メッセージ W_t を計算する。

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \mathrm{ROTL}^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & 16 \le t \le 79 \end{cases}$$

ただし、$\mathrm{ROTL}^n(x)$ は w ビット x に対する n ビット左巡回シフトである。

$$\mathrm{ROTL}^n(x) = (x \ll n) \vee (x \gg (w-n))$$

(ii) 5 個のバッファ変数を $(i-1)$ 番目のハッシュ値 $\{H_j^{(i-1)}\}_{j=0}^4$ で初期値化する。

$$\begin{aligned} a_0 &= H_0^{(i-1)} \\ b_0 &= H_1^{(i-1)} \\ c_0 &= H_2^{(i-1)} \\ d_0 &= H_3^{(i-1)} \\ e_0 &= H_4^{(i-1)} \end{aligned}$$

(iii) $0 \le t \le 79$ に対して以下の計算を繰り返す。

$$\left\{ \begin{array}{rcl} T & = & \mathrm{ROTL}^5(a_t) + f_t(b_t, c_t, d_t) + e_t + K_t + W_t \\ e_{t+1} & = & d_t \\ d_{t+1} & = & c_t \\ c_{t+1} & = & \mathrm{ROTL}^{30}(b_t) \\ b_{t+1} & = & a_t \\ a_{t+1} & = & T \end{array} \right.$$

ただし、K_t $(0 \le t \le 79)$ は 32 ビットワードの定数 (FIPS PUB 180-2 参照)、+ は 32 ビットのワード単位ごとの 2^{32} を法とした加算を意味する。

(iv) i 番目の中間ハッシュ値を

$$\begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + a_{80} \\ H_1^{(i)} &= H_1^{(i-1)} + b_{80} \\ H_2^{(i)} &= H_2^{(i-1)} + c_{80} \\ H_3^{(i)} &= H_3^{(i-1)} + d_{80} \\ H_4^{(i)} &= H_4^{(i-1)} + e_{80} \end{aligned}$$

で計算する。

上記手続き (i)–(iv) を N 回繰り返した最終的な 160 ビット中間ハッシュ値

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)}$$

がメッセージ M のハッシュ値である。なお、160 ビット中間ハッシュ値を $H^{(i)} = H_0^{(i)} \| H_1^{(i)} \| H_2^{(i)} \| H_3^{(i)} \| H_4^{(i)}$ と表すと、SHA-1 圧縮関数 $C_\cdot^1(\cdot)$ は以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}^1(H^{(i-1)}), \quad 1 \le i \le N$$

でまとめて表現できる。

(IV) SHA-1 のハッシュ値計算の別法

(III) の SHA-1 のハッシュ値計算の手続き中の 80 個のワード群 $W_0, W_1, \cdots, W_{79}$ の使用を節約しつつ、以下のように SHA-1 のハッシュ値を計算できる[*2]。

MASK = `0000000f` とおく。(III) の SHA-1 の前処理を実行したものとする。

$M^{(i)}$ を処理するために、$1 \leq i \leq N$ に対して以下の手続き (i)–(iv) を実行する。

(i) $0 \leq t \leq 15$ に対して

$$W_t = M_t^{(i)}$$

(ii) 5 個のバッファ変数を $(i-1)$ 番目のハッシュ値 $\{H_j^{(i-1)}\}_{j=0}^4$ で初期値化する。

$$\begin{aligned} a_0 &= H_0^{(i-1)} \\ b_0 &= H_1^{(i-1)} \\ c_0 &= H_2^{(i-1)} \\ d_0 &= H_3^{(i-1)} \\ e_0 &= H_4^{(i-1)} \end{aligned}$$

(iii) $0 \leq t \leq 79$ に対して以下の計算を繰り返す。

$$s = t \wedge \mathrm{MASK}$$

If $t \geq 16$ then

$$W_s = \mathrm{ROTL}^1(W_{(s+13)\wedge\mathrm{MASK}} \oplus W_{(s+8)\wedge\mathrm{MASK}} \oplus W_{(s+2)\wedge\mathrm{MASK}} \oplus W_s)$$

$$\left\{ \begin{array}{rcl} T & = & \mathrm{ROTL}^5(a_t) + f_t(b_t, c_t, d_t) + e_t + K_t + W_s \\ e_{t+1} & = & d_t \\ d_{t+1} & = & c_t \\ c_{t+1} & = & \mathrm{ROTL}^{30}(b_t) \\ b_{t+1} & = & a_t \\ a_{t+1} & = & T \end{array} \right.$$

(iv) i 番目の中間ハッシュ値を

$$\begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + a_{80} \\ H_1^{(i)} &= H_1^{(i-1)} + b_{80} \\ H_2^{(i)} &= H_2^{(i-1)} + c_{80} \\ H_3^{(i)} &= H_3^{(i-1)} + d_{80} \\ H_4^{(i)} &= H_4^{(i-1)} + e_{80} \end{aligned}$$

で計算する。

上記手続き (i)–(iv) を N 回繰り返した最終的な 160 ビット中間ハッシュ値

$$H^{(N)} = H_0^{(N)} || H_1^{(N)} || H_2^{(N)} || H_3^{(N)} || H_4^{(N)}$$

がメッセージ M のハッシュ値である。

[*2] (III) で得られるメッセージダイジェストと (IV) のそれとは同一であるが、(IV) の方法では使用するメモリを節約できるが、その代わり計算時間が長くなる。

4.3.2.3　SHA-256 の技術仕様

MD4、MD5、SHA-1 と同様な設計である。SHA-256 のアルゴリズムは、以下の (I) の関数を用いた、(II) の前処理と (III) のハッシュ値計算の 2 段階からなる。

(I) SHA-256 で使用される関数

SHA-256 では、32 ビットワードを入力変数および出力変数とする、以下の 6 個の論理関数が用いられる。

$$\begin{cases} \mathrm{Ch}(x,y,z) &= (x \wedge y) \oplus (\neg x \wedge z) \\ \mathrm{Maj}(x,y,z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \Sigma_0^{\{256\}}(x) &= \mathrm{ROTR}^2(x) \oplus \mathrm{ROTR}^{13}(x) \oplus \mathrm{ROTR}^{22}(x) \\ \Sigma_1^{\{256\}}(x) &= \mathrm{ROTR}^6(x) \oplus \mathrm{ROTR}^{11}(x) \oplus \mathrm{ROTR}^{25}(x) \\ \sigma_0^{\{256\}}(x) &= \mathrm{ROTR}^7(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{SHR}^3(x) \\ \sigma_1^{\{256\}}(x) &= \mathrm{ROTR}^{17}(x) \oplus \mathrm{ROTR}^{19}(x) \oplus \mathrm{SHR}^{10}(x) \end{cases}$$

ここで、記号 $\wedge, \oplus$ はそれぞれビットごとの論理積、排他的論理和を表し、$\neg x$ は x のビット反転を表す。また、$\mathrm{ROTR}^n(x)$ は 32 ビットワード x の n ビット右巡回シフト、$\mathrm{SHR}^n(x)$ は 32 ビットワード x の n ビット右シフトを意味する。

(II) SHA-256 の前処理

(i) 入力メッセージ M について、メッセージ長が 512 ビットの倍数になるように初期パディングされたメッセージ

$$M||1||0^k||\ell$$

を計算する。ただし、ℓ は M のメッセージ長をバイナリ表現した時のビット数、k は $\ell + 1 + k \equiv 448 \pmod{512}$ を満たす最小値である。

(ii) 初期パディングされたメッセージは N 個の 512 ビット単位のブロック $\{M^{(i)}\}_{i=1}^{N}$ に分割される。ただし、各々の $M^{(i)}$ は、16 個の 32 ビット長のワード

$$M^{(i)} = M_0^{(i)}||M_1^{(i)}||\cdots||M_{15}^{(i)}$$

からなる。

(iii) 初期ハッシュ値の j 番目のワード $H_j^{(0)}$ $(0 \le j \le 7)$

$$
\begin{aligned}
H_0^{(0)} &= \texttt{6a09e667} \\
H_1^{(0)} &= \texttt{bb67ae85} \\
H_2^{(0)} &= \texttt{3c6ef372} \\
H_3^{(0)} &= \texttt{a54ff53a} \\
H_4^{(0)} &= \texttt{510e527f} \\
H_5^{(0)} &= \texttt{9b05688c} \\
H_6^{(0)} &= \texttt{1f83d9ab} \\
H_7^{(0)} &= \texttt{5be0cd19}
\end{aligned}
$$

を設定する。

(III) SHA-256 のハッシュ値計算

N 個のメッセージブロック $M^{(1)}, \cdots, M^{(N)}$ の $M^{(i)}$ に対して、$1 \le i \le N$ の順に以下のことを実行する。

(i) 次式で定義する SHA-256 メッセージスケジュール関数を用いて拡張メッセージ W_t を計算する。

$$
W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases}
$$

ただし、+ は 32 ビットのワード単位ごとの 2^{32} を法とした加算を意味する。

(ii) 8 個のバッファ変数を $(i-1)$ 番目のハッシュ値 $\{H_j^{(i-1)}\}_{j=0}^{7}$ で初期値化する。

$$
\begin{aligned}
a_0 &= H_0^{(i-1)} \\
b_0 &= H_1^{(i-1)} \\
c_0 &= H_2^{(i-1)} \\
d_0 &= H_3^{(i-1)} \\
e_0 &= H_4^{(i-1)} \\
f_0 &= H_5^{(i-1)} \\
g_0 &= H_6^{(i-1)} \\
h_0 &= H_7^{(i-1)}
\end{aligned}
$$

(iii) $0 \le t \le 63$ に対して以下の計算を繰り返す。

$$\begin{cases} T_1 &= h_t + \Sigma_1^{\{256\}}(e_t) + \mathrm{Ch}(e_t, f_t, g_t) + K_t^{256} + W_t \\ T_2 &= \Sigma_0^{\{256\}}(a_t) + \mathrm{Maj}(a_t, b_t, c_t) \\ h_{t+1} &= g_t \\ g_{t+1} &= f_t \\ f_{t+1} &= e_t \\ e_{t+1} &= d_t + T_1 \\ d_{t+1} &= c_t \\ c_{t+1} &= b_t \\ b_{t+1} &= a_t \\ a_{t+1} &= T_1 + T_2 \end{cases}$$

ただし、K_t^{256} $(0 \le t \le 63)$ は 32 ビットワードの定数 (FIPS PUB 180-2 参照) である。

(iv) i 番目の中間ハッシュ値を

$$\begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + a_{64} \\ H_1^{(i)} &= H_1^{(i-1)} + b_{64} \\ H_2^{(i)} &= H_2^{(i-1)} + c_{64} \\ H_3^{(i)} &= H_3^{(i-1)} + d_{64} \\ H_4^{(i)} &= H_4^{(i-1)} + e_{64} \\ H_5^{(i)} &= H_5^{(i-1)} + f_{64} \\ H_6^{(i)} &= H_6^{(i-1)} + g_{64} \\ H_7^{(i)} &= H_7^{(i-1)} + h_{64} \end{aligned}$$

で計算する。

上記手続き (i)−(iv) を N 回繰り返した最終的な 256 ビット中間ハッシュ値

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$$

がメッセージ M のハッシュ値である。なお、256 ビット中間ハッシュ値を $H^{(i)} = H_0^{(i)} \| H_1^{(i)} \| \cdots \| H_7^{(i)}$ と表すと、SHA-256 圧縮関数 $C_{\cdot}^{256}(\cdot)$ は以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}^{256}(H^{(i-1)}), \quad 1 \le i \le N$$

でまとめて表現できる。

4.3.2.4　SHA-512 の技術仕様

SHA-256 のワード長 32 ビットを 64 ビットに変更したものである。SHA-512 のアルゴリズムは、以下の (I) の関数を用いた、(II) の前処理と (III) のハッシュ値計算の 2 段階からなる。

(I) SHA-512/SHA-384 で使用される関数

SHA-512、SHA-384 では、64 ビットワードを入力変数および出力変数とする以下の 6 個の論理関数が用いられる。

$$\begin{cases} \mathrm{Ch}(x,y,z) &= (x \wedge y) \oplus (\neg x \wedge z) \\ \mathrm{Maj}(x,y,z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \Sigma_0^{\{512\}}(x) &= \mathrm{ROTR}^{28}(x) \oplus \mathrm{ROTR}^{34}(x) \oplus \mathrm{ROTR}^{39}(x) \\ \Sigma_1^{\{512\}}(x) &= \mathrm{ROTR}^{14}(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{ROTR}^{41}(x) \\ \sigma_0^{\{512\}}(x) &= \mathrm{ROTR}^{1}(x) \oplus \mathrm{ROTR}^{8}(x) \oplus \mathrm{SHR}^{7}(x) \\ \sigma_1^{\{512\}}(x) &= \mathrm{ROTR}^{19}(x) \oplus \mathrm{ROTR}^{61}(x) \oplus \mathrm{SHR}^{6}(x) \end{cases}$$

ここで、記号 $\wedge, \oplus$ はそれぞれビットごとの論理積、排他的論理和を表し、$\neg x$ は x のビット反転を表す。また、$\mathrm{ROTR}^n(x)$ は 64 ビットワード x の n ビット右巡回シフト、$\mathrm{SHR}^n(x)$ は 64 ビットワード x の n ビット右シフトを意味する。

(II) SHA-512 の前処理

(i) 入力メッセージ M について、メッセージ長が 1,024 ビットの倍数になるように初期パディングされたメッセージ

$$M||1||0^k||\ell$$

を計算する。ただし、ℓ は M のメッセージ長をバイナリ表現した時のビット数、k は $\ell + 1 + k \equiv 896 \pmod{1024}$ を満たす最小値である。

(ii) 初期パディングされたメッセージは N 個の 1,024 ビット単位のブロック $\{M^{(i)}\}_{i=1}^N$ に分割される。ただし、各々の $M^{(i)}$ は、16 個の 64 ビット長のワード

$$M^{(i)} = M_0^{(i)}||M_1^{(i)}||\cdots||M_{15}^{(i)}$$

からなる。

(iii) 初期ハッシュ値の j 番目のワード $H_j^{(0)}$ $(0 \le j \le 7)$

$$\begin{aligned} H_0^{(0)} &= \texttt{6a09e667f3bcc908} \\ H_1^{(0)} &= \texttt{bb67ae8584caa73b} \\ H_2^{(0)} &= \texttt{3c6ef372fe94f82b} \\ H_3^{(0)} &= \texttt{a54ef53a5f1d36f1} \\ H_4^{(0)} &= \texttt{510e527fade682d1} \\ H_5^{(0)} &= \texttt{9b05688c2b3e6c1f} \\ H_6^{(0)} &= \texttt{1f83d9abfb41bd6b} \\ H_7^{(0)} &= \texttt{5be0cd19137e2179} \end{aligned}$$

を設定する。

(III) SHA-512 のハッシュ値計算

N 個のメッセージブロック $M^{(1)}, \cdots, M^{(N)}$ の $M^{(i)}$ に対して、$1 \le i \le N$ の順に以下のことを実行する。

(i) 次式で定義する SHA-512 メッセージスケジュール関数を用いて拡張メッセージ W_t を計算する。

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{512\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{512\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 79 \end{cases}$$

ただし、+ は 64 ビットのワード単位ごとの 2^{64} を法とした加算を意味する。

(ii) 8 個のバッファ変数を $(i-1)$ 番目のハッシュ値 $\{H_j^{(i-1)}\}_{j=0}^{7}$ で初期値化する。

$$\begin{aligned} a_0 &= H_0^{(i-1)} \\ b_0 &= H_1^{(i-1)} \\ c_0 &= H_2^{(i-1)} \\ d_0 &= H_3^{(i-1)} \\ e_0 &= H_4^{(i-1)} \\ f_0 &= H_5^{(i-1)} \\ g_0 &= H_6^{(i-1)} \\ h_0 &= H_7^{(i-1)} \end{aligned}$$

(iii) $0 \le t \le 79$ に対して以下の計算を繰り返す。

$$\begin{cases} T_1 &= h_t + \Sigma_1^{\{512\}}(e_t) + \mathrm{Ch}(e_t, f_t, g_t) + K_t^{512} + W_t \\ T_2 &= \Sigma_0^{\{512\}}(a_t) + \mathrm{Maj}(a_t, b_t, c_t) \\ h_{t+1} &= g_t \\ g_{t+1} &= f_t \\ f_{t+1} &= e_t \\ e_{t+1} &= d_t + T_1 \\ d_{t+1} &= c_t \\ c_{t+1} &= b_t \\ b_{t+1} &= a_t \\ a_{t+1} &= T_1 + T_2 \end{cases}$$

ただし、K_t^{512} $(0 \le t \le 79)$ は 64 ビットワードの定数 (FIPS PUB 180-2 参照) である。

(iv) i 番目の中間ハッシュ値を

$$\begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + a_{80} \\ H_1^{(i)} &= H_1^{(i-1)} + b_{80} \\ H_2^{(i)} &= H_2^{(i-1)} + c_{80} \\ H_3^{(i)} &= H_3^{(i-1)} + d_{80} \\ H_4^{(i)} &= H_4^{(i-1)} + e_{80} \\ H_5^{(i)} &= H_5^{(i-1)} + f_{80} \\ H_6^{(i)} &= H_6^{(i-1)} + g_{80} \\ H_7^{(i)} &= H_7^{(i-1)} + h_{80} \end{aligned}$$

で計算する。

上記手続き (i)–(iv) を N 回繰り返した最終的な 512 ビット中間ハッシュ値

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$$

がメッセージ M のハッシュ値である。なお、512 ビット中間ハッシュ値を $H^{(i)} = H_0^{(i)} \| H_1^{(i)} \| \cdots \| H_7^{(i)}$ と表すと、SHA-512 圧縮関数 $C_{\cdot}^{512}(\cdot)$ は以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}^{512}(H^{(i-1)}), \quad 1 \le i \le N$$

でまとめて表現できる。

4.3.2.5 SHA-384 の技術仕様

SHA-512 とほとんど同じで以下の 2 点だけが異なる。

1. 初期ハッシュ値 $H^{(0)}$ は SHA-256、SHA-512 のそれと異なり、以下のように変更する。

$$H_0^{(0)} = \texttt{cbbb9d5dc1059ed8}$$
$$H_1^{(0)} = \texttt{629a292a367cd507}$$
$$H_2^{(0)} = \texttt{9159015a3070dd17}$$
$$H_3^{(0)} = \texttt{152fecd8f70e5939}$$
$$H_4^{(0)} = \texttt{67332667ffc00b31}$$
$$H_5^{(0)} = \texttt{8eb44a8768581511}$$
$$H_6^{(0)} = \texttt{db0c2e0d64f98fa7}$$
$$H_7^{(0)} = \texttt{47b5481dbefa4fa4}$$

2. SHA-512 の 512 ビットハッシュ値 $H^{(N)}$ の左 384 ビットで打ち切った 384 ビットハッシュ値

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)}$$

をメッセージ M の SHA-384 でのハッシュ値 $H^{(N)}$ として採用する。

4.3.2.6 その他

標準化関連として、SHA-1 は、FIPS PUB 180-2 のほか、ANSI X9.30 (Part 2)、ISO/IEC 10118-3、RFC 3174 (Informational)、SSL3.0 および RFC 2246: TLS1.0 (Proposed Standard) などに掲載されている。

SHA-256、SHA-384、SHA-512 のいずれも FIPS PUB 180-2 および NESSIE に掲載されている。

4.3.2.7 SHA-1 の安全性評価結果

■総評 SHA-1 のアルゴリズムで示したように SHA-1 の解析のためには圧縮関数だけでなく、入力メッセージを拡張する部分も分析しなければならない。SHA-1 の前のバージョンである

SHA は入力の拡張部分が排他的論理和だけで構成されており、その分析に基づいて圧縮関数に対する衝突が発見できた。しかし、この SHA に対する攻撃はメッセージ拡張の部分で 1 ビットの左巡回シフトを用いる SHA-1 には適用できないことが報告されている。現在、SHA-1 に対する実用的な攻撃は報告されていないため、暗号の応用分野で使うには安全であると考えられる。しかし、全数探索攻撃に対する安全性は確保しなければならないため、ハッシュ値の長さが n ビットである場合、誕生日攻撃により $2^{n/2}$ 個のメッセージに対するハッシュ値の中で衝突が見つかる可能性から、ハッシュ値を十分長くする必要があり、ハッシュ値が 160 ビットである SHA-1 は 2^{80} 個のハッシュ値に対して衝突が発見される可能性があることから、将来にわたって安全であるとは保証できない。

4.3.2.8 SHA-256 の安全性評価結果

■総評 現在広く使用されている 160 ビットのハッシュ値を出力する SHA-1 の安全性が損なわれるとの報告は今のところないが、長期間の使用および今後の計算機向上を見越して SHA-1 を設計変更した SHA-256 に関しては、その設計基準は明確ではないが、メッセージ拡張処理が複雑になり、ハッシュ関数に対する既存の攻撃法がいずれも適用できない。

また、提案されて日が浅いので今後も引き続き安全性の検討が必要であるものの、SHA-256 は現在のところ安全であると結論付けられる。

1. 評価者 1

   評価結果概要: SHA-256 の安全性に関して以下に掲げる観点に関して評価を行った。

   (a) MD 型ハッシュ関数に対する Dobbertin[3, 4, 5] や Chaubaud & Joux[6] の攻撃法はいずれも SHA-256 には適用することは難しい。

   (b) SHA-1 と比較して繰り返し段数が少ないように見えるし、またその上、公式文書が何も無く僅かな仕様から設計選択に関する選択基準や暗号用変数を再構成することは困難ではあるが、SHA-256 の基本構成部分の最大の特長は、既存ハッシュ関数に対するよりもかなり高い暗号強度を提供している。

   (c) 内蔵している圧縮関数の差分特性に関して調査した結果、考えられる繰り返し特性も各段での圧縮関数に共通な特性のいずれも見出すことはできなかった。

   (d) 段数定数が 2 分割に関して互いに対称となるような簡略版の SHA-256 (modified SHA-256、詳細 (vi) 参照) は安全ではない。

   以上の調査結果から、現在既知のあらゆる攻撃法のいずれも SHA-256 には適用できないし、原像 (preimage)、弱い意味の原像 (second preimage) の計算複雑度を 2^{256} 以下に、また通常の誕生日攻撃である $2^{256/2} = 2^{128}$ の複雑度以下に帰着可能とする攻撃法は見当たらない。

   詳細:

   (a) 主に三つの安全性評価基準:

      (i) 衝突困難性 (collision resistance)

(ii) 原像計算困難性[*3]、一方向性 (preimage resistance、one-wayness)

(iii) 弱原像計算困難性[*4]、弱衝突困難性 (second preimage resistance、weak collision-resistance)

に関して検討を行った結果、いずれも問題が見付からなかった。

(b) SHA-1 のアルゴリズムと SHA-256 のそれとの主な相違点は以下のとおりである。

i. メッセージスケジュール計算において排他的論理和 $\oplus$ の代わりに加法演算 + を採用して計算を複雑にした結果、

(i) 差分パターンが線形符号にはなりえないので、解析を困難にしている。

(ii) SHA-1 の特性が強化された。

(iii) SHA-0、SHA-1 で観測されていた入力語の回転不変性が見られなくなった。

(iv) メッセージスケジュール長対作業変数用レジスタ長比 (各圧縮関数計算ごとの作業変数の全回転の数) は劣化している。

この劣化による安全性度の具体的評価法は定かではないが、これは、一見すると SHA-256 の安全性劣化につながると思われる。一方、この比の低下は SHA-256 の作業変数の更新の複雑さの向上と SHA-1 と異なり、二レジスタ変数が各段で修正されていることとを考え併せれば補償されているともいえる。

ii. 状態レジスタ更新関数において、32 ビットの 8 レジスタ $(a_t, b_t, c_t, d_t, e_t, f_t, g_t, h_t)$ を使用する SHA-256 は、32 ビットの 5 レジスタ $(a_t, b_t, c_t, d_t, e_t)$ を使用する SHA-0 や SHA-1 と類似しているものの以下の点で異なる。

(i) ラウンド関数が複雑になり、強力でかつ高速な拡散性を有する。すなわち、多数決関数 Maj(·) および選択関数 Ch(·) 等の非線形関数の両方が各段で適用され、また、二つのレジスタ変数が各段で修正されている。

(ii) SHA-0、SHA-1 で見られた状態レジスタ更新関数の低い一様性が更に低くなっている。これは少し安全性向上につながるかも知れない。

iii. 多数決関数 Maj(·) および選択関数 Ch(·) 等の非線形関数、シグマ関数 $\sigma_0(\cdot), \sigma_1(\cdot), \Sigma_0(\cdot), \Sigma_1(\cdot)$、定数 K_t 等は適切に設計されている。

(c) ハッシュ関数に対する既存の攻撃法: (i) Dobbertin の衝突探索法、(ii) Chaubaud & Joux の差分攻撃による衝突探索法、に関して検討した結果、メッセージ拡張処理が複雑になっているのでいずれの攻撃法も適用できないので、設計変更は安全性の向上に役立っていることが判った。

(d) 差分攻撃: 圧縮関数の差分特性を検討した結果、4 段での差分特性は、確率 2^{-8} 以下であり、64 段全体では $2^{-8\times16} = 2^{-128}$ 以下となる。これは 256 ビットハッシュ関数の衝突確率と同程度に低くなるので、圧縮関数に関する差分攻撃は適用できないと結論付けられる。

(e) 繰り返し差分攻撃に関して検討した結果、SHA-256 に対して適用できないことが判った。

(f) 極度に対称性のある初期ハッシュ値、定数 $(H_0^{(0)} = H_1^{(0)} = \cdots = H_7^{(0)} \in \Omega_{32})$ を選択し、

[*3] ハッシュ値 $h(M)$ が与えられたとき、$h(M) = h(M')$ を満たす M' を計算する困難性

[*4] メッセージ M とそのハッシュ値 $h(M)$ が与えられたとき、$h(M) = h(M')$ を満たす M' を計算する困難性

加法演算 + を排他的論理和 ⊕ に変更した、簡略化 SHA-256 (modified SHA-256) の場合、衝突困難性が無くなるので安全性が損なわれる。ただし、Ω_{32} は

$$\Omega_{32} = \{C \in \{0,1\}^{32} | \exists c \in \{0,1\}^{16}, C = c||c\} \tag{4.1}$$

で定義される対称 32 ビットワードの集合を意味する。

2. 評価者 2

評価結果概要:

（a）SHA-256、SHA-512 のアルゴリズムは SHA-1 のそれと以下の主な点でかなり異なる。

(i) メッセージ拡張処理が複雑になっているので、安全性の向上に役立っている。

(ii) 1 ステップで更新される変数が二つになっているので既存の攻撃法の適用を困難にしている。

（b）Draft FIPS180-2 への公開コメント (Jonsson, Kelsey's comments) に言及している。

（c）現在のところ、SHA-256 の致命的な欠点は勿論、安全性を疑わせるような点は見当たらない。提案されて日が浅いので、今後も引き続き安全性を検討する必要がある。

（d）SHA-1 より設計変更した点は幾つかの攻撃に対して安全性の向上に役立っていることが確認できた。

（e）Jonsson のコメントにあるように、NIST が SHA-256、SHA-384、SHA-512 の安全性評価を公開し、SHA-1 と異なる設計にした理由を明らかにすることが望まれる。

以上の結果、現在のところ SHA-256 の致命的な欠点や安全性を疑わせる点も見当たらないが、提案されて日が浅いので、安全性が十分に研究されているとはいえないので、引き続き安全性を検討する必要がある。

詳細:

（a）主に四つの安全性評価基準: (i) ランダム性、(ii) 誕生日攻撃法、(iii) 衝突困難性、(iv) 一方向性に関して検討を行った結果、いずれも問題が見付からなかった。

（b）並列処理による高速化の観点から安全性の検討を行った結果、問題が見付からなかった。すなわち、SHA-256、SHA-384、SHA-512 では、並列化アルゴリズムに基づく誕生日攻撃法による安全性の低下が難しくなるように SHA-1 から設計変更されているといえる。

（c）ハッシュ関数に対する既存の攻撃法: (i) Dobbertin の衝突探索法、(ii) Chaubaud & Joux の差分攻撃による衝突探索法、(iii) 簡略版 (1-round) ハッシュ関数の衝突探索法に関して検討した結果、メッセージ拡張処理が複雑になっているのでいずれの攻撃法も適用できないので、設計変更は安全性の向上に役立っていることが判った。

（d）ハッシュ関数を鍵付ハッシュのメッセージ認証に利用する場合、extension property 問題を解決する Kelsey 法の安全性の証明がなされるまで、Bellare, Canetti, Krawczyk による、効率的かつ安全性が証明可能な鍵付ハッシュ関数の構成法を利用することに言及している。

4.3.2.9 SHA-384 / SHA-512 の安全性評価結果

■総評 現在広く使用されている 160 ビットのハッシュ値を出力する SHA-1 の安全性が損なわれるとの報告は今のところないが、長期間の使用および今後の計算機向上を見越して SHA-1 を設計変更した SHA-384/SHA-512 に関しては、その設計基準は明確ではないが、メッセージ拡張処理が複雑になり、またハッシュ関数に対する既存の攻撃法がいずれも適用できない。

また、提案されて日が浅いので今後も引き続き安全性の検討が必要であるものの、SHA-384/SHA-512 は現在のところ安全であると結論付けられる。

1. 評価者 1

   評価結果概要: SHA-384/SHA-512 の安全性に関して以下に掲げる観点に関して評価を行った。

   (a) MD 型ハッシュ関数に対する Dobbertin[3, 4, 5] や Chaubaud &Joux[6] の攻撃法はいずれも SHA-384 や SHA-512 には適用することは難しい。

   (b) SHA-1 と比較して繰り返し段数が少ないように見えるし、またその上、公式文書が何も無く僅かな仕様から設計選択に関する選択基準や暗号用変数を再構成することは困難ではあるが、SHA-384 や SHA-512 の基本構成部分の最大の特長は、既存ハッシュ関数に対するよりもかなり高い暗号強度を提供している。

   (c) 内蔵している圧縮関数の差分特性に関して調査した結果、考えられる繰り返し特性も各段での圧縮関数に共通な特性のいずれも見出すことはできなかった。

   (d) 段数定数が 2 分割に関して互いに対称となるような簡略版の SHA-384/SHA-512 (modified SHA-384/modified SHA-512、詳細 (vi) 参照) は安全ではない。

   以上の調査結果から、現在既知のあらゆる攻撃法のいずれも SHA-384 や SHA-512 には適用できないし、原像 (preimage)、弱い意味の原像 (second preimage) の計算複雑度を各々 2^{384} や 2^{512} 以下に、また通常の誕生日攻撃の複雑度 (各々 $2^{384/2} = 2^{192}$ や $2^{512/2} = 2^{256}$) 以下の複雑度に帰着可能とする攻撃法は見当たらない。

   詳細:

   (a) 主に三つの安全性評価基準:

   (i) 衝突困難性 (collision resistance)

   (ii) 原像計算困難性、一方向性 (preimage resistance, one-wayness)

   (iii) 弱原像計算困難性、弱衝突困難性 (second preimage resistance, weak collision-resistance)

   に関して検討を行った結果、いずれも問題が見付からなかった。

   (b) SHA-1 のアルゴリズムと SHA-384 や SHA-512 のそれとの主な相違点は以下のとおりである。

   i. メッセージスケジュール計算において排他的論理和 $\oplus$ の代わりに加法演算 + を採用して計算を複雑にした結果、

   (i) 差分パターンが線形符号にはなりえないので、解析を困難にしている。

   (ii) SHA-1 の特性が強化された。

(iii) SHA-0、SHA-1 で観測されていた入力語の回転不変性が見られなくなった。
(iv) メッセージスケジュール長対作業変数用レジスタ長比 (各圧縮関数計算ごとの作業変数の全回転の数) は劣化している。
この劣化による安全性度の具体的評価法は定かではないが、これは、一見すると SHA-384 や SHA-512 の安全性劣化につながると思われる。一方、この比の低下は SHA-384 や SHA-512 の作業変数の更新の複雑さの向上と SHA-1 と異なり、二レジスタ変数が各段で修正されていることとを考え併せれば補償されているともいえる。

ii. 状態レジスタ更新関数において、64 ビットの 8 レジスタ $(a_t, b_t, c_t, d_t, e_t, f_t, g_t, h_t)$ を使用する SHA-384 や SHA-512 は 32 ビットの 5 レジスタ $(a_t, b_t, c_t, d_t, e_t)$ を使用する SHA-0 や SHA-1 と類似しているものの以下の点で異なる。
(i) ラウンド関数が複雑になり、強力でかつ高速な拡散性を有する。すなわち、多数決関数 Maj(·) および選択関数 Ch(·) 等の非線形関数の両方が各段で適用され、また、二つのレジスタ変数 T_1, T_2 が各段で更新されている。
(ii) SHA-0、SHA-1 で見られた状態レジスタ更新関数の低い一様性が更に低くなっている。これは少し安全性向上につながるかも知れない。

iii. 多数決関数 Maj(·) および選択関数 Ch(·) 等の非線形関数、シグマ関数 $\sigma_0(\cdot), \sigma_1(\cdot), \Sigma_0(\cdot), \Sigma_1(\cdot)$、定数 K_t 等は適切に設計されている。

(c) ハッシュ関数に対する既存の攻撃法: (i) Dobbertin の衝突探索法、(ii)Chaubaud & Joux の差分攻撃による衝突探索法、に関して検討した結果、メッセージ拡張処理が複雑になっているのでいずれの攻撃法も適用できないので、設計変更は安全性の向上に役立っていることが判った。

(d) 差分攻撃: 圧縮関数の差分特性を検討した結果、4 段での差分特性は、確率 2^{-8} 以下であり、80 段全体では $2^{-8\times 20} = 2^{-160}$ 以下となる。これは 384 ビットハッシュ関数あるいは 512 ビットハッシュ関数の衝突確率よりも低くはないが、同一の低い重み特性をつなぎ併せることによりこの限界値に漸近する全体の差分確率を構成することはできそうにない。また、差分特性は最初の段階での擬衝突 (pseudocollision) の検出だけを意味し、また実際には多重の衝突を検出しなければならないこと等を考え併せれば、SHA-384 や SHA-512 の圧縮関数に関する差分攻撃は適用できないと結論付けられる。

(e) 繰り返し差分攻撃に関して検討した結果、SHA-384 や SHA-512 に対して適用できないことが判った。

(f) 極度に対称性のある初期ハッシュ値、定数 $(H_0^{(0)} = H_1^{(0)} = \cdots = H_7^{(0)} \in \Omega_{64})$ を選択し、加法演算 + を排他的論理和 $\oplus$ に変更した、簡略化 SHA-512 (modified SHA-512) の場合、衝突困難性が無くなるので安全性が損なわれる。ただし、Ω_{64} は

$$\Omega_{64} = \{C \in \{0,1\}^{64} | \exists c \in \{0,1\}^{32}, C = c||c\} \tag{4.2}$$

で定義される対称 64 ビットワードの集合を意味する。

2. 評価者 2

評価結果概要: SHA-384/SHA-512 の安全性に関して以下に掲げる観点に関して評価を

行った。

（a）SHA-384 は SHA-512 と本質的に同じであるから SHA-512 だけに言及。

（b）SHA-256 と同じコメントで同一の評価。

4.3.2.10 ソフトウェア実装評価

CRYPTREC では実装評価はしていないが、2002 年度報告書作成時において [8, 9] に Pentium III 上などでの実装結果が示されている。

4.3.2.11 ハードウェア実装評価

CRYPTREC では実装評価はしていないが、2002 年度報告書作成時において [7] に SHA-1 と SHA-512 の FPGA 実装結果が示されている。

参考文献

[1] SHA-1: National Institute of Standrads and Technology (NIST) FIPS 180-1: Secure Hash Standard, April 1994.

[2] SHS: National Institute of Standrads and Technology (NIST) FIPS 180-2: Secure Hash Standard, August 2 2002.

[3] H. Dobbertin, *Cryptanalysis of MD4*, in *Journal of Cryptology*, **11**-4, Autumn, 1998.

[4] H. Dobbertin, *Cryptanalysis of MD5 Compress*, Presented at the rump session of Eurocrypt'96, May 14, 1996.

[5] H. Dobbertin, *The status of MD5 after a recent attack*, *CryptBytes*, **2**-2, 1996, pp3-6.

[6] F. Chaubaud and A. Joux, *Differential Collisions in SHA-0*, extended abstract, in CRYPTO'98, LNCS 1462, pp.56–71, 1998.

[7] T. Grembowski, et. al, *Comparative Analysis of the Hardware Implementations of Hash Functions SHA-1 and SHA-512*, Information Security, LNCS2433, pp.75-89, 2002.

[8] J. Nakajima and M. Matsui, *Performance Analysis and Parallel Implementation of Dedicated Hash Functions*, Eurocrypt02, LNCS2332, pp.167-pp.180.

[9] J. Nakajima and M. Matsui, *Performance Analysis and Parallel Implementation of Decicated Hash Functions on Pentium III*, IEICE TRANSACTIONS, Vol.E86-A, No.1, January 2003, pp.54-pp.63

第 5 章

擬似乱数生成系の評価

5.1 評価方法

ここで記述する擬似乱数生成は、ストリーム暗号用途とは違い、暗号の鍵または鍵の種生成などで利用する乱数の生成を目的とし、性質は真性乱数に近く、暗号学的強度を要求される。

乱数生成法は一般に、乱数生成器 (非決定論的生成器と通称される真性乱数器、Random Number Generator(RNG)) と擬似乱数生成器 (決定論的生成法と通称される Pseudorandom Number Generator (PRNG)) とに大別される。前者は電気回路の雑音や半導体の量子効果等のある種の物理量から生成したものであり、RNG の出力そのものを乱数として用いたり、PRNG への入力として用いられる。後者は RNG の出力からの種 (seed) を用いた、単一あるいは複数個の入力に対して複数の “擬似乱数” を生成 (出力値は seed の関数) するものである。

PRNG の数列は物理源から生成された RNG より、“乱雑さ (randomness)” の統計的検定法に対して良好な値をしばしば与えることが知られている。今回は、公開鍵暗号に付随して利用される擬似乱数生成アルゴリズム (PRNG) を中心として、文献等の調査によりその安全性評価を実施した。

5.2 評価結果総評

評価対象は、以下の規格の Appendix (Annex) に記載されている擬似乱数アルゴリズムである。これらの評価概要は表 5.1、表 5.2 のようにまとめられる。

- PRNG in ANSI X9.42-2001 Annex C.1/C.2
- PRNG in ANSI X9.62-1998 Annex A.4
- PRNG in ANSI X9.63-2001 Annex A.4
- PRNG for DSA in FIPS PUB 186-2 Appendix 3[*1]
- PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1
- PRNG in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1/3.2

[*1] CRYPTREC Report 2001 で PRNG based on SHA-1 としていたものと同一である。

今回の電子政府推奨暗号リスト素案においては、評価対象アルゴリズムのうち、特に実用上の問題点が指摘されていないものを 3 つほど例示として掲載した。

ただし、該当する規格中には複数の方式が存在しているが、そのなかから最善の方式を例示対象として採用することとしたため、電子政府推奨暗号リスト素案では、各擬似乱数アルゴリズムの規格名だけでなく、例示対象を具体的に示すアルゴリズム名が使用されている。具体的には、一つの規格中に内部関数として SHA-1 ベースと Triple DES ベースを利用した方式がある場合、それぞれを「PRNG based on SHA-1」「PRNG based on Triple DES」と表記する。同様に、擬似乱数の生成目的として、DSA 用とした場合と汎用目的用とした場合があるとき、それぞれを「PRNG for DSA」「PRNG for general purpose」と表記する。例えば、PRNG based on SHA-1 for general purpose in FIPS 186-2 (+ change notice 1) revised Appendix 3.1 は、規格名 FIPS 186-2 (+ change notice 1) revised Appendix 3.1 中にある擬似乱数生成アルゴリズムで、内部関数として SHA-1 ベースを利用した汎用目的のものであることを指し示す。

5.3 個別暗号技術の評価

5.3.1 PRNG in ANSI X9.42-2001 Annex C.1

FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法 PRNG in ANSI X9.42-2001 Annex C.1 については、5.3.5 節を参照のこと。

5.3.2 PRNG in ANSI X9.42-2001 Annex C.2

本節では、CRYPTREC 評価対象擬似乱数生成系のうち、ANSI X9.17 Annex C から派生した擬似乱数生成法 PRNG in ANSI X9.42-2001 Annex C.2 について結果を記す。

5.3.2.1 技術概要

Triple DES をベースに設計されている擬似乱数生成法であり、時刻情報を入力し任意ビット長の乱数を生成する。ANSI X9.17 Annex C をベースに、いくつかの乱数を必要とする暗号を取り扱った ANSI で標準化されているもののうちの一つである。

5.3.2.2 技術仕様

仕様は次のとおりである。

入力: 64 ビットの乱数の種 V_0、Triple DES の鍵 (K_1, K_2, K_3)、出力ビット数 L、および日付・時刻ベクトル DT_j $(1 \le j \le \lceil \frac{L}{64} \rceil)$

出力: L ビットの乱数 p

Step 1 p を null string とする

表 5.1: 擬似乱数生成系の評価結果総評 (1/2)

ANSI X9.42-2001 Annex C.1/C.2	特徴
	Annex C.1: FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法の一つ。任意長の擬似乱数 (ビット列) を出力。内部関数として SHA-1 ベースのものが示されている。 Annex C.2: ANSI X9.17 Annex C から派生した、標準化された擬似乱数生成法の一つ。任意長の擬似乱数 (ビット列) を出力。擬似乱数生成に Triple DES および時刻情報を利用。
	総合評価
	パラメータが適切に設定された Annex C.1 については、今のところ実用上の重大な問題点は見つかっていない。適切なパラメータの設定法については 5.3.1 節を参照のこと。 Annex C.2 については強い攻撃法を仮定した場合の弱点が指摘されているので推奨できない。
ANSI X9.62-1998 Annex A.4	特徴
	FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法の一つ。複数個の 1 以上 q 未満の擬似乱数を出力。内部関数として SHA-1 および DES ベースのものが示されている。
	総合評価
	q 等のパラメータによっては擬似乱数出力分布について、PRNG for DSA in FIPS PUB 186-2 Appendix 3 を利用した DSA に対する攻撃に使われたのと同様の大きな偏りが生じるので推奨できない。
ANSI X9.63-2001 Annex A.4	特徴
	FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法の一つ。複数個の 1 以上 q 未満の擬似乱数を出力。内部関数として SHA-1 ベースのものが示されている。
	総合評価
	q 等のパラメータによっては擬似乱数出力分布について、PRNG for DSA in FIPS PUB 186-2 Appendix 3 を利用した DSA に対する攻撃に使われたのと同様の大きな偏りが生じるので推奨できない。
FIPS PUB 186-2 Appendix 3	特徴
	DSA 用の擬似乱数生成法として、FIPS PUB 186 Appendix 3 から記載されていた擬似乱数生成法。一個あるいは複数個の L ビットの乱数を出力するなど、さまざまなバリエーションがある。内部関数として SHA-1 および DES ベースのものが示されている。
	総合評価
	{0, 1} の分布の偏りを利用した 2^{64} の計算量と 2^{22} の既知署名を必要とする攻撃法が発表されている。この攻撃法は DSA での擬似乱数の使用時に特定の一つの鍵の使用回数を 200 万回以下に抑えることで防御できるものの、擬似乱数としては乱数出力に大きな偏りが生じているので推奨できない。change notice 1 による更新版を使用すべきである。

表 5.2: 擬似乱数生成系の評価結果総評 (2/2)

FIPS PUB 186-2 (+ change notice 1) Appendix 3.1	特徴
	FIPS PUB 186-2 Appendix 3.1 で標準化された擬似乱数生成法に対して、DSA 以外の目的でも使用できるように、擬似乱数出力がビット列となるよう仕様の一部を修正したものである。change notice 1 に伴う更新版である。
	総合評価
	いまのところ、パラメータを適切に設定すれば、実用上の重大な問題点は見つかっていない。但し、仕様書中で定義されている使い方の中には安全とは言い切れない方法が含まれているので、利用の際には 5.3.5 節を参照の上、適切な使い方を選択する必要がある。
FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1/3.2	特徴
	FIPS PUB 186-2 Appendix 3.1/3.2 をベースに標準化された擬似乱数生成法の一つで、change notice 1 に伴う更新版である。DSA 用の乱数生成については、PRNG for DSA in FIPS PUB 186-2 Appendix 3 で発見された DSA に対する攻撃に利用された擬似乱数出力の偏りがなくなるよう修正されている。DSA 以外の目的でも使用できるように、擬似乱数出力がビット列となるような仕様の修正もされている。
	総合評価
	いまのところ、パラメータを適切に設定すれば、実用上の重大な問題点は見つかっていない。但し、仕様書中で定義されている使い方の中には安全とは言い切れない方法が含まれているので、利用の際には 5.3.5 節を参照の上、適切な使い方を選択する必要がある。

Step 2 **for** $j \leftarrow 1$ **to** $\left\lceil \frac{L}{64} \right\rceil$ **do**

$$
\begin{aligned}
I_j &\leftarrow \mathrm{DES}_{K_3}(\mathrm{DES}^{-1}_{K_2}(\mathrm{DES}_{K_1}(\mathrm{DT}_j))) \\
x_j &\leftarrow \mathrm{DES}_{K_3}(\mathrm{DES}^{-1}_{K_2}(\mathrm{DES}_{K_1}(I_j \oplus V_{j-1}))) \\
V_j &\leftarrow \mathrm{DES}_{K_3}(\mathrm{DES}^{-1}_{K_2}(\mathrm{DES}_{K_1}(I_j \oplus x_j)))
\end{aligned}
$$

$p \leftarrow p \parallel x_j$

end for

Step 3 $p \leftarrow \left\lfloor \frac{p}{2^{-L \bmod 64}} \right\rfloor$ (左 L ビット)

注 1 V は秘密に保持する。

注 2 DT_j は j ごとに更新される。

5.3.2.3 調査結果

関連文献 [2, 1] を調査した結果を記す。

■調査のまとめ 出力系列が 64×2^{32} ビットに比べて十分に短い場合は直接攻撃 (乱数系列のみからそれを真正乱数と識別する攻撃方法) に対し Triple DES が安全なブロック暗号であるならば安全であることが知られている。しかしながら、より強い攻撃を仮定した場合、以下のような点が指摘されている。

- 一旦、鍵が知られてしまうと、以前の乱数出力もすべて導出可能となる。
- 攻撃者が時刻入力を制御できる場合には 2^{32} ブロック程度の乱数出力で真正乱数と識別可能となる。

■Kelsey らの解析

再入力攻撃に対するぜい弱性: 再入力攻撃が行われた場合 (つまり、なんらかの方法により補助入力の値を固定させることができた場合)、64×2^{32} ビット程度の擬似乱数系列を真正乱数と識別することができる。理由は以下のとおりである。

ブロック暗号は全単射であるため、補助入力値が固定された場合、その出力は約 2^{63} ブロック分 (1 ブロック = 64 ビット) の出力で一つの周期を形成する。また、各ブロックの出力は一つ前のブロックに対して決定的であるため、一つの周期の中に同じシンボルは二つ以上出現しないことになる。これに対して真正乱数の場合、2^{32} ブロック程度でおよそ一組同じシンボルが現れるため、真正乱数との間には識別可能な差が生じることとなる。

漏洩内部状態拡張攻撃に対するぜい弱性: 一般に、内部状態が一旦漏洩したとしても、その地点から十分離れた出力に対応する内部状態を推測することは難しくなっているほうが好ましい。

しかしながら、漏洩内部状態拡張攻撃が行われた場合 (ある時刻の内部状態である鍵 $K = (K_1, K_2, K_3)$ が漏洩した場合)、攻撃者は連続する 2 ブロック分の出力 x_j, x_{j+1} から中間状態 V_{j+1} を求めることができる。攻撃手法の詳細を以下に述べる。

攻撃者は入手した出力 x_j, x_{j+1} に対応する補助入力 $\mathrm{DT}_j, \mathrm{DT}_{j+1}$ を推定することで中間状態 V_{j+1} の候補 V'_{j+1} と V''_{j+1} を以下の式により求める。

$$V'_{j+1} := D_K(x_{j+1}) \oplus \mathrm{DT}_{j+1}$$
$$V''_{j+1} := E_K(x_j \oplus \mathrm{DT}_j)$$

ここで $E_K()$ および $D_K()$ は鍵 K による Triple DES の暗号化と復号処理を示す。$V'_{j+1} = V''_{j+1}$ となっている値が最終的な V_{j+1} の候補となる。補助入力である日付・時間ベクトルのエントロピー (特に DT_j が既知の場合の DT_{j+1} のエントロピー) はそれほど大きくないためこの推定により得られる候補の数は V の全状態数 2^{64} よりはるかに小さくなる。

■Desai らの解析 Desai らは Triple DES を安全なブロック暗号と仮定し、さまざまな攻撃および未来安全について検討している。

攻撃モデル 考察対象の攻撃モデルは次のとおり。

	鍵	状態	補助入力
選択入力攻撃 (CIA)	未知	既知	選択可能
選択状態攻撃 (CSA)	未知	選択可能	既知
既知鍵攻撃 (KKA)	既知	未知	既知

安全性は通常の識別不能性 (indistinguishability) で定義。

未来安全 (forward security) 現在の鍵と状態が攻撃者に知られたときの、それ以前の乱数出力について、

弱未来安全 (WFS): 攻撃者は解読前には擬似乱数出力は隠されている。攻撃者は隠されているものを秘密情報を知った後に導出する。

強未来安全 (SFS): 攻撃者は解読前から擬似乱数出力を知っている。攻撃者は知っていた出力と真正乱数との区別を秘密情報を知った後に行なう。

解析結果

CIA	CSA	KKA	WFS	SFS
安全	安全	危険	危険	危険

注: 上記「安全」は乱数出力が短い場合であり、出力ブロック長を m としたときの PRNG の advantage との gap は $O(\frac{m^2}{2^{64}})$ 程度存在することに注意されたい。(正確な数値については原文 [1] を当たって欲しい)

参考文献

[1] Anand Desai, Alejandro Hevia, and Yiqun Lisa Yin. A practice-oriented treatment of pseudorandom number generators. In Lars Ramkilde Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 368–383. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[2] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Cryptanalytic attacks on pseudorandom number generators. In Serge Vaudenay, editor, *Fast Software Encryption — 5th International Workshop, FSE'98*, volume 1372 of *Lecture Notes in Computer Science*, pages 168–188, Berlin, Heidelberg, New York, 1998. Springer-Verlag.

5.3.3 PRNG in ANSI X9.62-1998 Annex A.4

FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法 PRNG in ANSI X9.62-1998 Annex A.4 については、5.3.5 節を参照のこと。

5.3.4 PRNG in ANSI X9.63-2001 Annex A.4

FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法 PRNG in ANSI X9.63-2001 Annex A.4 については、5.3.5 節を参照のこと。

5.3.5 PRNG in FIPS PUB 186-2 (+ change notice 1) Appendix & revised Appendix

本節では、CRYPTREC 評価対象擬似乱数生成系のうち、FIPS PUB 186 Appendix 3 から派生した擬似乱数生成法

- PRNG for DSA in FIPS PUB 186-2 Appendix 3 (5.3.6 節参照)
- PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1
- PRNG in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 & 3.2
- PRNG in ANSI X9.42-2001 Annex C.1
- PRNG in ANSI X9.62-1998 Annex A.4
- PRNG in ANSI X9.63-2001 Annex A.4

について結果を記す。

5.3.5.1 技術概要

署名アルゴリズム DSA を定義している FIPS PUB 186[*2]で、DSA で必要となる乱数を生成する方法の一例として擬似乱数生成法がいくつか示されている。FIPS PUB 186 Appendix 3 を元にしたと考えられる擬似乱数生成法は多数標準化されている。これらの擬似乱数生成法は、ビット列を生成するもの、ある範囲の整数を生成するもの、補助入力を必要とするもの・しないものなど多数の派生アルゴリズムがある。これらのアルゴリズムは内部で一方向性関数 G を必要とし、G の構成法として SHA-1 ベースのものと DES ベースのものが参考に上げられている。

5.3.5.2 技術仕様

評価対象アルゴリズムの仕様を順に示す。内部で呼び出されている補助関数の仕様はこの節の最後にまとめて示す。なお、パラメータの細かい条件については下記の擬似乱数生成法ごとに微妙に異なるので FIPS PUB 186-2 (+ change notice 1) および ANSI そのもので確認されたい。

PRNG for DSA in FIPS PUB 186-2 Appendix 3 については 5.3.6 節で解説されているので、この節では、change notice 1 に関係する PRNG for DSA in FIPS PUB 186-2 Appendix 3.1 & 3.2 の仕様を本節の記号で書き直したものを紹介するのみとした。

なお、以下の仕様において、$\mathrm{IV}_{\text{SHA-1}}$ を SHA-1 の $H_0 \| H_1 \| H_2 \| H_3 \| H_4$ の初期値、$\mathrm{IV}_{\text{SHA-1}}^{\lll 32}$ を $\mathrm{IV}_{\text{SHA-1}}$ を左 32 ビット巡回シフトしたものを表すものとする。

■PRNG for DSA in FIPS PUB 186-2 Appendix 3.1

入力: 160 ビットの素数 q, XKEY, XSEED_j $(1 \le j \le m)$

[*2] この規格の最新版が FIPS PUB 186-2 (+ change notice 1) である。

出力: m 個の秘密鍵 $x_1, x_2, \ldots, x_m$

for $j \leftarrow 1$ **to** m **do**
$(x_j, \mathrm{XKEY}) \leftarrow B(\mathrm{IV}_{\mathrm{SHA\text{-}1}}, \mathrm{XKEY}, \mathrm{XSEED}_j, q)$
end for

注 1 入力の XKEY は新しくかつ秘密の値を選ぶ。

■PRNG for DSA in FIPS PUB 186-2 Appendix 3.2

入力: 160 ビットの素数 q, XKEY, m 個一組のメッセージ $M_1, M_2, \ldots, M_m$
出力: 署名生成のための乱数 k_j $(1 \le j \le m)$

Step 1 $t \leftarrow \mathrm{IV}_{\mathrm{SHA\text{-}1}}^{\lll 32}$
Step 2 **for** $j \leftarrow 1$ **to** m **do**
$(k_j, \mathrm{XKEY}) \leftarrow B(t, \mathrm{XKEY}, 0, q)$
end for
Step 3 $t = \mathrm{SHA\text{-}1}(M_m)$ とし、Step 2 に戻る。

注 1 入力の XKEY は秘密の値を選ぶ。

■PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1
PRNG for DSA in FIPS PUB 186-2 Appendix 3.1 において、$\bmod q$ を外した x_j を出力する。DSA 以外でも利用可能な、一般的な擬似乱数生成アルゴリズムである。

■PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1

入力: 160 ビットの素数 q, XKEY, XSEED_j $(1 \le j \le m)$
出力: m 個の秘密鍵 $x_1, x_2, \ldots, x_m$

for $j \leftarrow 1$ **to** m **do**
$(w_1, \mathrm{XKEY}) \leftarrow B(\mathrm{IV}_{\mathrm{SHA\text{-}1}}, \mathrm{XKEY}, \mathrm{XSEED}_j, 2^{160})$
$(w_2, \mathrm{XKEY}) \leftarrow B(\mathrm{IV}_{\mathrm{SHA\text{-}1}}, \mathrm{XKEY}, \mathrm{XSEED}_j, 2^{160})$
$x_j \leftarrow (w_1 \parallel w_2) \bmod q$
end for

注 1 入力の XKEY は新しくかつ秘密の値を選ぶ。

■PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.2

入力: 160 ビットの素数 q, XKEY, m 個一組のメッセージ $M_1, M_2, \ldots, M_m$

出力: 署名生成のための乱数 k_j $(1 \le j \le m)$

Step 1 $t \leftarrow \mathrm{IV}_{\text{SHA-1}}^{\lll 32}$
Step 2 **for** $j \leftarrow 1$ **to** m **do**
$(w_1, \mathrm{XKEY}) \leftarrow B(t, \mathrm{XKEY}, 0, 2^{160})$
$(w_2, \mathrm{XKEY}) \leftarrow B(t, \mathrm{XKEY}, 0, 2^{160})$
$k_j \leftarrow (w_1 \parallel w_2) \bmod q$
end for
Step 3 t = SHA-1(M_m) とし、Step 2 に戻る。

注 1 入力の XKEY は秘密の値を選ぶ。

■PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1

PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 において、mod$q$ を外した x_j を出力する。DSA 以外でも利用可能な、一般的な擬似乱数生成アルゴリズムである。

■PRNG in ANSI X9.42-2001 Annex C.1

入力: 160 ビットの素数 q (使われない), XKEY, 出力ビット数 L, XSEED_j $(1 \le j \le \lceil \frac{L}{160} \rceil)$ (選択しない場合はすべて 0 とする)
出力: L ビットの乱数 p

Step 1 p を null string とする
Step 2 **for** $j \leftarrow 1$ **to** $\lceil \frac{L}{160} \rceil$ **do**
$(x, \mathrm{XKEY}) \leftarrow B(\mathrm{IV}_{\text{SHA-1}}, \mathrm{XKEY}, \mathrm{XSEED}_j, 2^{160})$
$p \leftarrow p \parallel x$
end for
Step 3 $p \leftarrow \lfloor \frac{p}{2^{-L \bmod 160}} \rfloor$ (左 L ビット)

注 1 入力の XKEY は新しくかつ秘密の値を選ぶ。
注 2 XSEED と XKEY は分けて入力し、XSEED と XKEY は別の情報源から作る。
注 3 G は SHA-1 ベースのもののみ紹介されているが、一方向性関数なら何でもよいというようにも読めなくもない。

■PRNG in ANSI X9.62-1998 Annex A.4

入力: XKEY, 素数 q ($f = \left\lceil \frac{\lfloor \log_2 q \rfloor + 1}{160} \right\rceil$ とおく), 発生すべき乱数の個数 l, $\mathrm{XSEED}_{i,j}$ $(1 \le i \le l,$ $1 \le j \le f)$
出力: l 個の乱数 $k_1, k_2, \ldots, k_l \in [1, q-1]$

for $i \leftarrow 1$ **to** l **do**
  for $j \leftarrow 1$ **to** f **do**
    $(x_j, \mathrm{XKEY}) \leftarrow B(\mathrm{IV}_{\mathrm{SHA\text{-}1}}, \mathrm{XKEY}, \mathrm{XSEED}_{i,j}, 2^{160})$
  end for
  $k_i \leftarrow ((x_1 \parallel x_2 \parallel \cdots \parallel x_f) \bmod (q-1)) + 1$
end for

注 1 入力の XKEY は新しくかつ秘密の値を選ぶ。

注 2 XSEED と XKEY は分けて入力し、XSEED と XKEY は別の乱数源から作られ XKEY と同じ安全性の要求条件を見たさなければならない。つまり許可されない開示から守られ予測不可能でなければならない。

注 3 G は SHA-1 ベースと DES ベースのものが紹介されているが、一方向性関数なら何でもよいというようにも読めなくもない。

■PRNG in ANSI X9.63-2001 Annex A.4

ANSI X9.62-1998 Annex A.4 と同じ。但し G は SHA-1 ベースのもののみ紹介されている。

■補助関数 基本アルゴリズム B

$$B : \{0,1\}^{160} \times \{0,1\}^b \times \{0,1\}^b \times \{0,1\}^{160} \to \{0,1\}^{160} \times \{0,1\}^b$$
$$(t, \mathrm{XKEY}, \mathrm{XSEED}, q) \mapsto (x, \mathrm{XKEY}')$$

を

$$x \leftarrow G(t, (\mathrm{XKEY} + \mathrm{XSEED}) \bmod 2^b) \bmod q$$
$$\mathrm{XKEY}' \leftarrow (1 + \mathrm{XKEY} + x) \bmod 2^b$$

と定義する。

一方向性関数

$$G : \{0,1\}^{160} \times \{0,1\}^b \to \{0,1\}^{160}; \quad (t, c) \mapsto x \qquad (160 \le b \le 512)$$

の参考実現法として、次の 2 方法が定義されている。

SHA-1 **ベースの構成**: $G(t, c)$ は SHA-1 の、

- 初期ベクトルを t
- メッセージのパディング方式を単に右に 0 パディング

に変更したもの。

DES **ベースの構成**: 添字は mod5 で考える。DES の鍵は下位 56 ビットを使う。

Step 1 $t_0 \parallel t_1 \parallel t_2 \parallel t_3 \parallel t_4 \leftarrow t$, $c_0 \parallel c_1 \parallel c_2 \parallel c_3 \parallel c_4 \leftarrow c$
(t_i, c_i は 32 ビット)

Step 2 $x_i \leftarrow t_i \oplus c_i$ for $0 \le i \le 4$

Step 3 $y_{i,0} \| y_{i,1} \leftarrow \mathrm{DES}_{c_{i+4} \| c_{i+3}}(x_i \| (x_{i+1} \oplus x_{i+4}))$ for $0 \le i \le 4$
($y_{i,j}$ は 32 ビット)

Step 4 $z_i \leftarrow y_{i,0} \oplus y_{i+2,1} \oplus y_{i+3,0}$ for $0 \le i \le 4$

Step 5 $G(t, c) \leftarrow z_0 \| z_1 \| z_2 \| z_3 \| z_4$

5.3.5.3 調査結果

関連文献 [3, 2, 1] を調査した結果を記す。また、以下の調査では一方向性関数 G として、SHA-1 ベースの構成と一般的な構成とで差が出なかったので、擬似乱数生成法の構成法として G を一般的な一方向性関数とした場合について明記しない。

■調査のまとめ 今回の調査結果を表 5.3 にまとめる。各アルゴリズムにおいて、√のマークがある結論が安全性評価として該当する。

表 5.3: FIPS PUB 186 Appendix 3 派生擬似乱数生成法の調査

	結論 1	結論 2	結論 3	結論 4	結論 5	結論 6	結論 7	結論 8
A			√					
B		√				√		
C					√			
D				√		√		
E			√				√	
F		√				√	√	
G			√				√	
H		√				√	√	
I			√				√	
J	√		√					√
K	√	√				√		√
L	√		√					√

表中の記号は次のとおりである。

A PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 (G が SHA-1 ベース)

B PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 (G が DES ベース)

C PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.2 (G が SHA-1 ベース)

D PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.2 (G が DES ベース)

E PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1 (G が SHA-1 ベース)

F PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1 (G が DES ベース)

G PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 (G が SHA-1 ベース)

H PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 (G が DES ベース)

I PRNG in ANSI X9.42-2001 Annex C.1

J PRNG in ANSI X9.62-1998 Annex A.4 (G が SHA-1 ベース)

K PRNG in ANSI X9.62-1998 Annex A.4 (G が DES ベース)

L PRNG in ANSI X9.63-2001 Annex A.4

結論 1 $160\left\lceil\frac{\lfloor\log_2 q\rfloor+1}{160}\right\rceil - \log_2 q$ が小さい場合は、乱数出力に大きな偏りが生じるので推奨できない。

結論 2 オプション入力が無い場合、周期 2^{80} 程度の乱数出力となるので、より長い周期となる擬似乱数生成器を使うのが望ましい。

結論 3 オプション入力が無い場合、周期 $2^{b/2}$ 程度の乱数出力となるので、$b < 256$ の場合は推奨できない。

結論 4 周期 2^{80} 程度の乱数出力となるので、より長い周期となる擬似乱数生成器を使う方が望ましい。

結論 5 周期 $2^{b/2}$ 程度の乱数出力となるので、$b < 256$ の場合は推奨できない。

結論 6 DES の鍵の全数探索を組み合わせた攻撃に対する考察が不十分と考えられるため、より考察が進んでいて安全と考えられる擬似乱数生成器が使える場合はそちらを使う方が望ましい。

結論 7 攻撃者がブロックごとの出力 (5.3.5.2 節の基本アルゴリズム B の x_i) を観測した後に次のブロックの補助入力 XSEED_{i+1} を操作できる場合、後述する式 (5.2) を用いることで $x_{i+1} = x_i$ とすることができる。補助入力が攻撃者の操作できない値であることを確認するか、補助入力 XSEED_{i+1} の入力が終わってから、x_i を出力するようにする必要がある。

結論 8 $160\left\lceil\frac{\lfloor\log_2 q\rfloor+1}{160}\right\rceil - \log_2 q$ が小さく、かつ攻撃者がブロックごとの出力 (5.3.5.2 節の基本アルゴリズム B の x_i) を観測した後に次のブロックの補助入力 XSEED_{i+1} を操作できる場合、後述する式 (5.2) を用いることで $x_{i+1} = x_i$ とすることができる。補助入力が攻撃者の操作できない値であることを確認するか、補助入力 XSEED_{i+1} の入力が終わってから、x_i を出力するようにする必要がある。

■Bleichenbacher の攻撃について 報道発表記事 [3] などにより、FIPS PUB 186-2 Appendix 3 で紹介されている DSA の乱数生成器 (PRNG for DSA in FIPS PUB 186-2 Appendix 3) の偏りを利用した攻撃が知られている。攻撃の詳細は明らかではないが、文献 [4] などのいくつかの文献

によると r を 160 ビットの乱数、q を 160 ビットの素数としたときに

$$r \bmod q \tag{5.1}$$

の分布が偏ることを利用したとある。

今回の調査では、式 (5.1) の原理により乱数出力に偏るものがあるかどうかを調べた。

発生する乱数が bit string であり、関係ないもの

- PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1
- PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1
- PRNG in ANSI X9.42-2001 Annex C.1

注: もちろん、160 ビットの素数 q に対し、乱数の値域を $[1, q-1]$ に制限するために 160 ビット以上程度の乱数出力に対し mod$q$ などをすると PRNG for DSA in FIPS PUB 186-2 Appendix 3 と同様の偏りを生じる。

偏りは存在するが無視できる程度に小さいと考えられるもの

- PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1
- PRNG for DSA in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.2

注: 文献 [4] によると、一様な乱数である場合のそれぞれの値に対する発生確率 q^{-1} に対し、偏りは最大 2^{-319} である。

パラメータによっては偏りが存在するもの

- PRNG in ANSI X9.62-1998 Annex A.4
- PRNG in ANSI X9.63-2001 Annex A.4

注: 出力する乱数の値域 $[1, q-1]$ を定義する q が 160 ビットの倍数である場合には PRNG for DSA in FIPS PUB 186-2 Appendix 3 と同程度の偏りが生じる。

■DES based G について　文献 [4] によると、G の一方向性を妨げるような性質として

$$G(t^{\lll 32}, c^{\lll 32}) = G(t, c)^{\lll 32}$$
$$(z_1, z_2, z_3, z_4, z_5) = G((t,t,t,t,t),(c,c,c,c,c)) \Rightarrow z_1 = z_2 = z_3 = z_4 = z_5$$

が知られている (ここで $\lll 32$ は 32 ビット左巡回シフト)。また、現時点では G の完全な一方向性を崩す解読法は知られていないが、非線形処理として用いられている DES のブロック長および実効鍵長が現在の計算機能力を持ってすれば十分に全数探索可能であるので、DES 全数探索の少数の組合せにより内部状態が復元できる可能性の考察が必要である。

■Kelsey らの解析 文献 [2] では、攻撃者がさまざまな能力を持っていると仮定した場合に擬似乱数生成器が脆弱性を持つか否かの検討が行われている。

選択入力攻撃に対するぜい弱性: 基本アルゴリズム B に対して以下のような補助入力 XSEED_i を与えることにより出力シンボルを固定させることができる。

$$\mathrm{XSEED}_i := \mathrm{XSEED}_{i-1} - x_{i-1} - 1 (\mathrm{mod} 2^b) \tag{5.2}$$

■Desai らの解析 Desai らは G を安全な一方向性関数と仮定し、さまざまな攻撃および未来安全について検討している。

攻撃モデル 考察対象の攻撃モデルは次のとおり。

	鍵	状態	補助入力
既知鍵攻撃 (KKA)	既知	未知	既知

安全性は通常の識別不能性 (indistinguishability) で定義。

未来安全 (forward security) 現在の鍵と状態が攻撃者に知られたときの、それ以前の乱数出力について、

弱未来安全 (WFS): 攻撃者は解読前には擬似乱数出力は隠されている。攻撃者は隠されているものを秘密情報を知った後に導出する。

強未来安全 (SFS): 攻撃者は解読前から擬似乱数出力を知っている。攻撃者は知っていた出力と真正乱数との区別を秘密情報を知った後に行なう。

解析結果

KKA	WFS	SFS
安全	安全	危険

注: 上記「鍵」は XKEY の初期値とし、アルゴリズムを等価変形したものが考察されている。

参考文献

[1] Anand Desai, Alejandro Hevia, and Yiqun Lisa Yin. A practice-oriented treatment of pseudorandom number generators. In Lars Ramkilde Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 368–383. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[2] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Cryptanalytic attacks on pseudorandom number generators. In Serge Vaudenay, editor, *Fast Software Encryption — 5th International Workshop, FSE'98*, volume 1372 of *Lecture Notes in Computer Science*, pages 168–188, Berlin, Heidelberg, New York, 1998. Springer-Verlag.

[3] Scientist discovers significant flaw that would have threatened the integrity of on-line transactions. Lucent Technologies Press Release, 2001. (`http://www.lucent.com/press/0201/010205.bla.html`).

[4] 外部評価者 1. Evaluation report on DSA. IPA Work Delivery, CRYPTREC 外部評価報告書 1002, 2001.

5.3.6　PRNG for DSA in FIPS PUB 186 Appendix 3

Digital Signature Standard (DSS) はデジタル署名の標準として三種のアルゴリズム: (1) Digital Signature Algorithm (DSA)、(2) RSA digital signature algorithm、(3) ECDSA が FIPS PUB 186-2 (+ change notice 1[*3]) で規定されている。これらのアルゴリズムの諸パラメータは、乱数または擬似乱数でなければならないので、それらの生成法が FIPS PUB 186-2 Appendix 3 に Random Number Generation for the DSA として規定されている。また、重要な change notice 1 が末尾に添付されている。

5.3.6.1　技術概要

DSA では、以下のパラメータを利用する。

1. p:$2^{L-1} < p < 2^L$ を満たす素数である。ただし、$512 \le L \le 1024$ を満たす 64 の倍数。
2. q:$2^{159} < q < 2^{160}$ を満たす $p-1$ の素因数。
3. g: $g = h^{(p-1)/q} \bmod p$ である。ただし、h は $h^{(p-1)/q} \bmod p > 1$ および $1 < h < p-1$ を満たす任意の整数。
4. x: 乱数または擬似乱数で生成された $0 < x < q$ を満たす整数。
5. y: $y = g^x \bmod p$
6. k: 乱数または擬似乱数で生成された $0 < k < q$ を満たす整数。

上記の p, q, g は公開パラメータでグループ内で共通に使用される。x, y は個人用の秘密鍵、公開鍵である。これらは通常ある一定期間固定される。x と k は署名生成時のみ用い、k は署名ごとに生成しなければならない。p と q の生成法は FIPS PUB 186-2 Appendix 2 で規定され、あるいは FIPS 推奨の安全な方法で生成することとする。x と k の生成法は FIPS PUB 186-2 Appendix 3 で規定され、あるいは FIPS 推奨の安全な方法で生成することとする。

メッセージ M に対する署名は次式で計算される r, s の組で与えられる。

$$\begin{cases} r &= (g^k \bmod p) \bmod q \\ s &= (k^{-1}(\text{SHA-1}(M) + xr)) \bmod q \end{cases}$$

ただし、k^{-1} は k の mod$q$ に関する逆元、すなわち、$(kk^{-1}) \bmod q = 1,\ 0 < k^{-1} < q$ であり、SHA-1(M) は、FIPS PUB 180-2 で規定した Secure Hash Algorithm (SHA) の 160 ビット出力値である。

Digital Signature Standard (DSS) で必要となる、乱数または擬似乱数によって、ユーザの秘密鍵 x やメッセージごとの k を生成するために、3 種類の擬似乱数生成法

(1) ANSI X9.17 Annex C の “Financial Institution Key Management (Wholesale)” による 160 ビットの一方向性関数 $G(t, c)$[*4]　(t は 160 ビット、c は b ビットである。$G(\cdot)$ が SHA-1 によ

[*3] change notice 1 に関しては 5.3.5 節でも紹介する。
[*4] FIPS PUB 186-2 Appendix 3.3 & 3.4 に該当する。

る場合 $160 \le b \le 512$、$G(\cdot)$ が Data Encryption Algorithm (DEA) による場合 (ANSI X9.17 Appendix C では DES を使用) $b = 160$ 固定)

(2) FIPS PUB 186-2 Appendix 3.1 の m 種類の x の生成法 (160 ビットの一方向性関数 $G(t, c)$ は SHA-1 または DES に基づく)

(3) FIPS PUB 186-2 Appendix 3.2 の m 種類の署名すべきメッセージの知識を前提としない k および r の生成法 (160 ビットの一方向性関数 $G(t, c)$ は SHA-1 または DES に基づく)

を FIPS PUB 186-2 推奨版として規定している。

5.3.6.2 技術仕様

■FIPS PUB 186-2 Appendix 3.1 の m 種類の x の生成法の技術仕様

(1) 新しい秘密数 ω_{xkey} を選択する。

(2) 512 ビットの初期ハッシュ値を $t = H_0||H_1||H_2||H_3||H_4$ とする。ただし、

$$\begin{cases} H_0 &= \texttt{67452301} \\ H_1 &= \texttt{EFCDAB89} \\ H_2 &= \texttt{98BADCFE} \\ H_3 &= \texttt{10325476} \\ H_4 &= \texttt{C3D2E1F0} \end{cases}$$

これは Secure Hash Standard (SHS) の初期ハッシュ値と同一である。

(3) $0 \le j \le m - 1$ として以下の (3.a)–(3.d) を繰り返す。

(3.a) ω_j を選択 (ユーザオプション) する。

(3.b) $c_j = (\omega_{\mathrm{xkey}} + \omega_j) \bmod 2^b$ $(160 \le b \le 512)$

(3.c) $x_j = G(t, c_j) \bmod q$

(3.d) $\omega_{\mathrm{xkey}} = (1 + \omega_{\mathrm{xkey}} + x_j) \bmod 2^b$

■FIPS PUB 186-2 Appendix 3.2 の m 種類の r, k の事前生成法の技術仕様

このアルゴリズムは m 個のメッセージに対して、一時に k, k^{-1}, r を事前に計算する方法を与えるものである。

(1) 新しい秘密数 ω_{kkey} を選択する。

(2) SHS での 512 ビットの初期ハッシュ値

$$H_0||H_1||H_2||H_3||H_4 = \texttt{67452301}||\texttt{EFCDAB89}||\texttt{98BADCFE}||\texttt{10325476}||\texttt{C3D2E1F0}$$

を左 32 ビット巡回シフトした

$$t = \texttt{EFCDAB89}||\texttt{98BADCFE}||\texttt{10325476}||\texttt{C3D2E1F0}||\texttt{67452301}$$

を選択する。

(3) $0 \le j \le m - 1$ として以下の (3.a)–(3.d) を繰り返す。

(3.a) $k = G(t, \omega_{\mathrm{kkey}}) \bmod q$

(3.b) $k_j^{-1} = k^{-1} \bmod q$

(3.c) $r_j = (g^k \bmod p) \bmod q$

(3.d) $\omega_{\mathrm{kkey}} = (1 + \omega_{\mathrm{kkey}} + k) \bmod 2^b$

(4) m 個のメッセージを $M_0, M_1, \ldots, M_{m-1}$ として、$0 \le j \le m-1$ に対して以下の (4.a)–(4.c) を繰り返す。

(4.a) h = SHA-1(M_j)、ここで SHA-1(·) は SHA-1 による一方向性関数を意味する。

(4.b) $s_j = (k_j^{-1}(h + xr_j)) \bmod q$

(4.c) (r_j, s_j) を M_j の署名とする。

(5) $t = h$

(6) (3) に戻る[*5]。

■FIPS PUB 186-2 Appendix 3.3 の SHA-1 による一方向性関数 $G(t, c)$ の技術仕様

$G(t, c)$ は、SHS の技術仕様 (FIPS PUB 180-2) の Sec. 6 の手順 (a)–(e)[*6]で計算できるが、これらの実行前に $\{H_j\}$ と M_1 の初期化を以下の手順で行う[*7]。

(i) 160 ビットの t の 32 ビット分割 $t = t_0||t_1||t_2||t_3||t_4$ に対し

$$H_j = t_j, \ (0 \le j \le 4)$$

とおく。

(ii) $M_1 = c||0^{512-b}$

SHS の技術仕様の Sec. 6 の手順の (a)–(e) を実行すると、ステップ (e) の最終段で 5 個のワードからなる 160 ビット長の $H = H_0||H_1||H_2||H_3||H_4$ が得られ、これを $G(t, c)$ とする。

■FIPS PUB 186-2 Appendix 3.4 の DES による一方向性関数 $G(t, c)$ の技術仕様

a_1, a_2, b_1, b_2 を 32 ビット文字列とし、b_1' を b_1 の下位 24 ビット文字列とする。

$$\begin{cases} K &= b_1'||b_2 \\ A &= a_1||a_2 \end{cases}$$

に対して記号 $\mathrm{DES}_K(A)$ を

$$\mathrm{DES}_K(A) = \mathrm{DES}_{(b_1', b_2)}(a_1, a_2)$$

で定義すると、$\mathrm{DES}_K(A)$ は、通常の 64 ビットブロック A に対する 56 ビット鍵 K を有する DES の暗号文を意味する。160 ビットの t, c に対する一方向性関数 $G(t, c)$ を以下の手順で計算する。

[*5] ステップ (3) では次の m 個のメッセージの署名に必要な量の事前計算を可能にしている。次の m 個のメッセージが用意されない限りステップ (4) は実行されない。ステップ (4) とステップ (5) が完了すればステップ (3) が実行され、次の m 個のメッセージの最初が用意されるまで、結果は保存されている。また、ステップ (3) の ω_{kkey} の計算時に $r_0, \ldots, r_{m-1}$ と $k_0^{-1}, \ldots, k_{m-1}^{-1}$ を記憶するために、2 個の m 次元のアレイが必要となる。

[*6] SHA-1 の技術仕様の SHA-1 のハッシュ値計算のステップ (i)–(iv) に相当する。

[*7] FIPS PUB 186-2 (+ change notice 1) では Sec. 6 の手順による別法であった FIPS PUB 180-2 の Sec. 8 の手順が削除されている。

(1) t, c を各々 32 ビットに分割する。

$$\begin{cases} t &= t_1||t_2||t_3||t_4||t_5 \\ c &= c_1||c_2||c_3||c_4||c_5 \end{cases}$$

(2) $1 \le i \le 5$ に対して

$$x_i = t_i \oplus c_i$$

を計算する。

(3) $1 \le i \le 5$ に対して

$$\begin{cases} b_1 &= c_{((i+3) \bmod 5)+1} \\ b_2 &= c_{((i+2) \bmod 5)+1} \\ a_1 &= x_i \\ a_2 &= x_{(i \bmod 5)+1} \oplus x_{((i+3) \bmod 5)+1} \\ y_{i,1}||y_{i,2} &= \mathrm{DES}_{(b_1', b_2)}(a_1, a_2) \end{cases}$$

を計算する。ただし、$y_{i,1}, y_{i,2}$ は 32 ビットである。

(4) $1 \le i \le 5$ に対して

$$z_i = y_{i,1} \oplus y_{((i+1) \bmod 5)+1,2} \oplus y_{((i+2) \bmod 5)+1,1}$$

を計算する。

(5) $G(t, c) = z_1||z_2||z_3||z_4||z_5$ をメッセージダイジェストとして出力する。

■DSS (FIPS PUB 186-2) の change notice 1 に伴う変更点

DSS (FIPS PUB 186-2) は応用分野におけるデジタル署名の生成と認証時に用いられる DSA を規定している。また、ANSI X9.31 (Digital Signature using Reversible Public Key Cryptography for the Finanicial Services Industry (rDSA)) および ANSI X9.62 (Public Key Cryptography for the Finanicial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)) の使用を認めている。また、現存 DSA の使用するための過渡期間を規定している。

FIPS PUB 186-2 は SHS (FIPS PUB 180-2) と連携して使用される。法としての素数 p の大きさやユーザの秘密鍵 x およびメッセージごとの秘密数 k の生成法を規定している。

(I) 素数 p の大きさ

FIPS PUB 186-2 で規定しているように DSA の継続使用に対する注意や FIPS PUB 186-2 Appendix 3 で規定している乱数生成法に対する修正および DSA の鍵生成以外で用いられる時の技術仕様を与えている。以下の注意は現存システムにある reversible 公開鍵アルゴリズム使用時のガイダンスである。

素数 p は、もともとの FIPS PUB 186-2 の Sec. 4 で、DSA の p を $2^{L-1} < p < 2^L$ を満たす素数として定義し、また L の範囲が $512 \le L \le 1024$ を満たす 64 の倍数であるとしていた。

しかしながら、change notice 1 では、L の取り得る値を $L = 1024$、すなわち $2^{1023} < p < 2^{1024}$ に限定しなければならないと規定した。また、この変更に伴い、現存システムで用いられている RSA や Rabin-Williams アルゴリズムの法 n と n の素因数 p, q のサイズについて、n を最低 1024 ビット、p, q を n の約半分のビット長にしなければならないと規定した。

(II) 乱数生成

FIPS PUB 186-2 Appendix 3 ではユーザの秘密鍵 x およびメッセージごとの秘密数 k を q を法として 0 から 160 ビットの乱数として規定している。最近、Appendix 3 で規定している擬似乱数生成法に見られる {0, 1} の分布の偏りを利用した 2^{64} の計算量と 2^{22} の既知署名を必要とする攻撃法が発表されている [1]。この攻撃法は FIPS PUB 186-2 での擬似乱数あるいはその修正版の擬似乱数の使用時に特定の一つの鍵の使用回数を 200 万回以下に抑えることで防御できるので、生成される署名回数の上限に関するガイダンスを設けなければならない。あるいは FIPS PUB 186-2 Appendix 3 の擬似乱数生成法の代替方式として以下の擬似乱数生成法の更新版を使用しても良い。なお、この変更は {0, 1} の分布の偏りを修正したものであり、互換性を損なうものではない。

以下に掲げる二つのアルゴリズムでは一方向性関数 $G(t, c)$ を用いる。ただし、t は 160 ビット、c は b ビット、$G(t, c)$ は 160 ビットである。FIPS PUB 186-2 で規定されている G の構成法は、FIPS PUB 180-2 で規定されている SHA-1 を用いる方法 ($160 \le b \le 512$) と FIPS PUB 46-3 で規定されている DES を用いる方法 ($b = 160$ 固定) の二つがある。

1. FIPS PUB 186-2 Appendix 3.1 の m 種類の x の計算法の更新版 [FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1]
   x を署名者の秘密鍵とする。m 個の x を以下の方法で生成する。
   (1) 新しい鍵種用の秘密の数 ω_{XKEY} を選択する。
   (2) SHS での 512 ビットの初期ハッシュ値 $H_0||H_1||H_2||H_3||H_4$ と同一の

   $$t = \texttt{67452301||EFCDAB89||98BADCFE||10325476||C3D2E1F0}$$

   を選択する。
   (3) $0 \le j \le m - 1$ に対して以下の (3.a)–(3.c) を実行する。
   (3.a) ω_{XSEED_j} を選択する (ユーザオプション)。
   (3.b) $0 \le i \le 1$ に対して以下の (3.b.i)–(3.b.iii) を実行する。
   (3.b.i) $c = (\omega_{XKEY} + \omega_{XSEED_j}) \bmod 2^b$
   (3.b.ii) $w_i = G(t, c)$
   (3.b.iii) $\omega_{XKEY} = (1 + \omega_{XKEY} + w_i) \bmod 2^b$
   (3.c) $x_j = (w_0||w_1) \bmod q$

2. FIPS PUB 186-2 Appendix 3.2 の m 種類の k, r の事前計算法の更新版 [FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.2]
   m 個のメッセージ用の k, k^{-1}, r を一時に事前計算する方法を与える。
   (1) 鍵種 ω_{KKEY} 用の秘密の初期値を選択する。
   (2) SHS での 512 ビットの初期ハッシュ値 $H_0||H_1||H_2||H_3||H_4$ を左 32 ビット巡回シフトした

   $$t = \texttt{EFCDAB89||98BADCFE||10325476||C3D2E1F0||67452301}$$

   を選択する。

(3) $0 \le j \le m-1$ に対して以下の (3.a)–(3.d) を実行する。
  (3.a) $0 \le i \le 1$ に対して以下の (3.a.i)–(3.a.ii) を実行する。
    (3.a.i) $w_i = G(t, \omega_{KKEY})$
    (3.a.ii) $\omega_{KKEY} = (1 + \omega_{KKEY} + w_i) \bmod 2^b$
  (3.b) $k = (w_0 \| w_1) \bmod q$
  (3.c) $k_j = k^{-1} \bmod q$
  (3.d) $r_j = (g^k \bmod p) \bmod q$
(4) $M_0, \ldots, M_{m-1}$ を次の m 個のメッセージとする。$0 \le j \le m-1$ に対して以下の (4.a)–(4.c) を実行する。
  (4.a) $h = \text{SHA-1}(M_j)$
  (4.b) $s_j = (k_j^{-1}(h + xr_j)) \bmod q$
  (4.c) (r_j, s_j) が M_j の署名とする。
(5) $t = h$
(6) (3) に戻る。

3. 汎用乱数生成法

いくつかの FIPS では、FIPS 推奨あるいは NIST 勧告の乱数生成法を使用することを要求している。その際、上記 FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 または FIPS PUB 186-2 Appendix 3.1 で規定された乱数生成法が他の推奨乱数生成法と並んで使用されるかもしれない。これらの擬似乱数生成法を DSA 用鍵生成以外の目的で使用する場合、mod$q$ 演算を省略しなければならない。これに伴い、以下の変更を行う。

A. FIPS PUB 186-2 Appendix 3.1 のステップ (3.c)
  $x_j = G(t, c_j) \bmod q$ を $x_j = G(t, c_j)$ に変更する。
B. FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 のステップ (3.c)
  $x_j = (w_0 \| w_1) \bmod q$ を $x_j = (w_0 \| w_1)$ に変更する。

5.3.6.3 安全性評価

■総評 FIPS PUB 186-2 Appendix 3 での擬似乱数生成法を用いた DSA に対する攻撃法 [1] が発表されている。報道発表記事などによれば、この攻撃法は、出力される擬似乱数の {0, 1} 分布の偏りを利用して、2^{64} の計算量と 2^{22} の既知署名により DSA の攻撃が行えるとされている。

この攻撃法に対しては、DSA での擬似乱数の使用時に特定の一つの鍵の使用回数を 200 万回以下に抑えることで防御できるため、生成される署名回数の上限に関するガイダンスを設けるなど、生成する擬似乱数の長さや回数を考慮すれば、FIPS PUB 186-2 Appendix 3 のままでも、DSA 用擬似乱数生成法としては安全に利用できるといえる。

しかしながら、一般的な擬似乱数生成法としては、明らかに乱数出力に大きな偏りが生じているので推奨することはできない。change notice 1 による更新版を使用すべきである。

参考文献

[1] CNN.com.SCI-TECH, Cryptologists sees digital signature flaw, fix: http://www.cnn.com/2001/TECH/internet/02/06/DSA.flaw.idg/index.html

[2] Secure Hashing: http://csrc.nist.gov/CryptoToolkit/tkhash.html

[3] New hashing algorithms (SHA-256, SHA-384, and SHA-512): Descrptions of SHA-256, SHA-384, and SHA-512

[4] FIPS Pub 186-2, Digital Signature Standard (DSS) (2000 January 27 更新) Appendix3: Random Number Generation for the DSA

5.4 擬似乱数生成系の検定

5.4.1 擬似乱数検定の概要

共通鍵暗号、公開鍵暗号、暗号プロトコルの安全性は、その根拠を秘密鍵やパラメータの秘匿性、ランダム性等に置いて議論している。従って秘密鍵、イニシャルベクトル等に使われる数列に何らかの偏りがあった場合や、推定されやすい値を用いた場合には、その安全性が必ずしも保証されるとは限らない。

本節では、暗号アルゴリズム、プロトコル等で利用する鍵などのパラメータとしては “利用すべきではない” 擬似乱数あるいは擬似乱数発生装置を検出するための乱数検定法と、それら検定法を基に記述された擬似乱数検定ソフトウェアライブラリで、インターネット上で公開されているものについて述べる。これらの検定法は、理想的な乱数列からは明らかに逸脱した性質を持つことを判定する上で、役に立つと思われる。

なお、本節で取り上げる擬似乱数検定法は、いずれも計算機の利用を前提とし、数列生成装置から発生させた検定対象となる一定長の数列に対して、ある決められた数式に基づいて計算された統計量を組み合わせ、その結果に統計的な処理を施し、確率論、統計論等から導かれる理論値と比較するものである。これらの検定で利用するのは、数列のごく一部であり、あらゆるパターンを調べ尽くしているわけではない。

これらの検定に合格することは、擬似乱数として “最低限満たすべき条件の一つ” にすぎず、その逆つまり生成された擬似乱数の安全性が十分であることを示しているわけでは決してない。ましてや擬似乱数生成アルゴリズムの安全性に対する保証が与えられるわけではない点には、特に注意すべきである。現時点では、これさえ満たせば十分安全であることが保証されるような擬似乱数検定法は知られていない。

また、本節で述べる多くの検定法、検定プログラムでは、入力の与え方、検定結果の解釈および最終判定は検定実施者にまかされている部分が多い。検定方法と検定結果の解釈の説明については、各乱数検定法、ならびに乱数検定プログラムに付属するドキュメント等を参考に、理論的に正しい解釈を与えるよう心がけ、また可能であれば複数の機関で独立に乱数検定を実施するなどにより、独善的な解釈や間違った解釈を与えないよう心がける必要がある。

なお本文章は、本節で紹介する乱数検定ライブラリについて必ずしも正しい動作を保証するものではない。また公開されている乱数検定ライブラリの一部には、ソフトウェアバグが含まれている可能性もあり、それが原因で擬似乱数列が本来持つ性質とは異なる性質が導かれることも十分有り得ることを念頭に入れておくべきである。

5.4.2 NIST : Special Publication 800-22

Special Publication 800-22 (以後、SP 800-22 と略) は、NIST が公開している暗号アプリケーションのための乱数と擬似乱数の統計試験ツール、およびドキュメントである [5, 6]。本ツール

表 5.4: SP800-22 に含まれる乱数検定法 (1/2)

試験番号	検定法	概要
1	The Frequency (Monobit) Test	入力数列に含まれる 0 と 1 の個数の偏りを調べる。
2	Frequency Test within a Block	入力数列を 256 ビットに区切り、その中の 0 と 1 の個数の割合の偏りを調べる。
3, 4	The Cumulative Sums (Cusums) Test	入力数列の 0/1 を −1/1 に変換し、先頭または一番後ろから 1 ビットずつその値を加算していく。加算操作中における絶対値の最大値の偏りを調べる。
5	The Runs Test	入力数列内に連 (1 または 0 が連続している部分) がいくつあるかを数えて、その数の偏りを調べる。
6	Test for the Longest Run of Ones in a Block	入力数列を 256 ビットに区切り、その中の最大の連の長さの偏りを調べる。
7	The Binary Matrix Rank Test	入力数列を 32×32 ビットの部分数列に分割し、それを行列に書き下した時の階数の偏りを調べる。
8	The Discrete Fourier Transform (Spectral) Test	入力数列を離散フーリエ変換によって周波数成分に分解し、各周波数におけるピークの高さが閾値を越える数を数えてその偏りを調べる。
9−156	The Non-overlapping Template Matching Test	9 ビットの Template を用意し、入力数列中にそれらの Template がいくつ現れるかを数えて、その偏りを調べる。Template と同じビット列が出現したら、出現した Template 以降のビットから探索を再開する。SP800-22 内のツールでは、148 個の Template について検定を行う。
157	The Overlapping Template Matching Test	9 ビットがすべて 1 のビット列を Template として用意する。入力数列中にその Template がいくつ現れるかを数えて、その偏りを調べる。Template が出現してもしなくても、観測する場所を 1 ビットずつずらしながら調べる。
158	Maurer's "Universal Statistical" Test	入力数列において、任意の 7 ビットのパターンが現れてから、次に現れるまでの距離の偏りを調べる。
159	The Approximate Entropy Test	入力数列の中に 10 ビット、11 ビットで取り得るパターンがそれぞれいくつ含まれるかを計算し、その数の偏りを調べる。
160−167	The Random Excursions Test	入力数列の 0/1 を −1/1 に変換し、先頭から加算して行く。合計の値が 0 の場所から次に 0 になるまでを一つの cycle と考え、8 種類 (−4~−1, 1~4) の State の出現数の偏りを調べる。この検定法では State 別に 8 種類の試験が用いられる。

表 5.5: SP800-22 に含まれる乱数検定法 (2/2)

試験番号	検定法	概要
168−185	The Random Excursions Variant Test	Random Excursions Test と同様に、数列の 0/1 を −1/1 に変換し、先頭から加算して行く。入力数列の先頭から最後までをまとめて扱い、18 種類 (−9~−1, 1~ 9) の State の出現数の偏りを調べる。この検定法では State 別に 18 種類の試験が用いられる。
186−187	The Serial Test	入力数列の中に、16 ビットで取り得るパターン、15 ビットで取り得るパターン、14 ビットで取り得るパターンがそれぞれいくつ含まれるかを計算し、その数の偏りを調べる。16 ビットと 15 ビットのパターンによる試験と 15 ビットと 14 ビットのパターンによる試験の 2 種類が用いられる。
188	The Lempel-Ziv Compression Test	入力数列の先頭から最後までを Lempel-Ziv データ圧縮アルゴリズムを用いてどの程度まで圧縮可能かを調べる。
189	The Linear Complexity Test	入力数列を 500 ビットのブロックに分割し、それぞれの数列の線形複雑度を求め、その偏りを調べる。

は、AES (Advanced Encryption Standard) の選定時に、候補としてエントリーされた各共通鍵ブロック暗号の暗号文出力への乱数検定法として利用され、その結果が報告されている [7, 8]。

5.4.2.1 乱数検定の概要

SP 800-22 に含まれる検定ツールの 2002 年 10 月現在のバージョンは 1.5 であり、本文章ではこれを元にする。本ツールでは、対象となる擬似乱数に対して、表 5.4、表 5.5 に示す 16 種類の検定法、189 個の試験を行うことができる。

入力する擬似乱数列は、ツールの制限から 1,000,000 ビット程度を複数列用意する必要がある。ちなみに AES レポートでは 1 Mbits (= 2^{20} bits) の数列 300 個を入力としている。本ツールの出力は、各試験について、「合格率 (pass rate)」と「分布 (variant)」の二つがある。各試験では、対象の擬似乱数列が正規分布もしくは χ^2 分布のどのあたりに位置するかを、P-Value という 0 から 1 の小数値で表現する。出力の「合格率」は P-Value が 0.01 以上を合格とし、AES レポートでは、入力 300 系列中の合格率が 0.9633 以上 (1% 棄却) である場合、その試験に対して合格であるとみなしている。一方、「分布」は、入力列の P-Value の分布が一様かどうかを見るもので、各系列の P-Value を 10% ごと 10 区間で集計し、その 10 区間に含まれる個数を χ^2 統計量に変換して P-Value で評価する。なお AES レポートには「分布」に関する検定結果が記載されていない。

5.4.2.2 注意すべき点

■擬似乱数の与え方 SP 800-22 では、与えられた擬似乱数列の統計的な検定法について述べられているが、与えられる擬似乱数の生成方法については、何も述べられていない。AES レポートでは、暗号アルゴリズムの一部を使用して、撹拌過程が検討可能な Partial round test と暗号アルゴリズム全体で擬似乱数を生成する Full round test の 2 種類のテストを行っている。また擬似乱数列の与え方としては、次の 8 種類の方法を用いている。

- Low Density Key
- Low Density Plaintext
- High Density Key
- High Density Plaintext
- Key Avalanche
- Plaintext Avalanche
- Key/Ciphertext Correlation
- Plaintext/Ciphertext Correlation

このように、擬似乱数発生装置の入力パラメータが必要な場合には、そのパラメータとして非常に偏った値を用いて発生させた擬似乱数を利用する方がよりよいと考えられる。

■「分布」の合否判定閾値 「分布」に関する検定結果については、合否判定に用いる P-Value の閾値に関する記述あるいは文献が見当たらない。よって、これらの閾値については、検定ツール利用者が決めなければならないが、どのような値が最適なのかは、現時点では判断できない。ちなみに CRYPTREC が外部に依頼して実施した調査では、「分布」の P-Value について 1% 棄却として 0.01 未満をその試験の不合格として検定を行った。この場合、試験 8 (Discrete Fourier Transfer Test) と試験 188 (Lenpel-Ziv Compression Test) で予想以上に多くの不合格が発生するとの指摘があった。これは入力乱数に偏りがあったか、あるいは棄却域の設定の仕方に問題があった可能性もあるが、あるいはそもそも検定ツール内に埋め込まれている「分布」の既定値が、理想的な値からずれているという可能性も否定はできない。(ソースコードを眺めると、過去に一度、これらの値が変更された形跡が残っている。) こうした点からも「分布」評価には十分な配慮を要すると思われる。

5.4.3 DIEHARD

DIEHARD はフロリダ州立大学 George Marsaglia 教授によって開発された擬似乱数検定ツールである [3]。本ツールは、物理デバイスを用いて発生させた擬似乱数列の検定法としてしばしば用いられている [2]。DIEHARD 検定プログラムはソースコードと DOS、Linux、Sun の上でコンパイルされた実行形式が公開されており、ダウンロードして利用可能である。また関連するものとしては、他者が作成したものではあるが、Java 言語で記述されたもの (論文発表)[4] や

ユーザインタフェースを改良したもの [1, 9] 等 (一部有料) がある。

5.4.3.1 乱数検定の概要

本ツールでは、対象となる擬似乱数に対して、表 5.6、表 5.7 に示す 18 種類の検定法を行うことができる。添付のドキュメントには、これら 18 種類の検定法は runs test を除いてすべて著者が独自に開発したものであると述べられている。

入力する擬似乱数列としては、10MB (= 80 Mbits) から 11MB (= 88 Mbits) 程度を用意する必要がある。より短いデータでも一部動作はするが、データファイルが終わった時点でテストを終了し、次のテストに進む。本ツールの結果としては各試験とも 0 から 1 の小数値である p-value が出力される。p-value は、入力されたデータが理想的な乱数の場合には 0 以上 1 未満で一様に分布し、そうでない場合は、0 あるいは 1 に近い値となる。本ツールに添付されているドキュメント (tests.txt) には、$p < 0.025$ あるいは $p > 0.975$ だった場合、5% の棄却域で検定に不合格であると判断すると記述されている。

5.4.3.2 注意すべき点

■合格率の基準となる閾値 本ツールをすべて適用する場合、全部で 250 種類の p-value を考慮する必要がある。理想的な乱数が入力される場合には、p-value の値は 0 以上 1 未満で一様に分布する。これらすべての検定で合格することが求められるのであれば、5% 棄却による検定の場合には $p < 0.0001$ あるいは $p > 0.9999$ の時に不合格と判断するべきなのではないか、との指摘がある [2]。

参考文献

[1] Balasubramanian, Narasimhan “Diehard GUI”
http://www-stat.stanford.edu/\~{}naras/diehard/snapshots.html)

[2] Intel, “The Intel Random Number Generator,”, 1999.
(available at http://www.intel.com/design/security/rng/rngppr.htm

[3] B. Narasimhan “DIEHARD,” (available at http://stat.fsu.edu/\~{}geo/diehard.html)

[4] B. Narasimhan “JDiehard: An implementation of Diehard in Java,” Proceedings in DSC2001, 2001.
(available at http://www.ci.tuwien.ac.at/Conferences/DSC-2001/Proceedings/,

[5] NIST, Special Publication 800-22, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,”
(available at http://csrc.nist.gov/rng/SP800-22b.pdf,\\http://csrc.nist.gov/rng/errata2.pdf)

[6] NIST, Special Publication 800-22, “NIST Statistical Test Suite,”
(available at http://csrc.nist.gov/rng/sts-1.5.tar,\\http://csrc.nist.gov/

表 5.6: DIEHARD に含まれる乱数検定法 (1/2)

試験番号	検定法	概要
1	birthday spacings test	入力数列を一定に区切り、その値を数字の大きさの順に並べて、各数字の間隔を調べる。理想的な乱数の場合はポアソン分布に従う。
2	overlapping 5-permutation test	入力数列を 1,000,000 個の 32 ビット整数と見た時、連続する 5 整数の順列の偏りを調べる。
3	binary rank test for 31 × 31 matrices	入力数列を 31×31 ビットの部分数列に分割し、それを行列に書き下した時の階数の偏りを調べる。
4	binary rank test for 32 × 32 matrices	入力数列を 32×32 ビットの部分数列に分割し、それを行列に書き下した時の階数の偏りを調べる。
5	binary rank test for 6 × 8 matrices	入力数列を 6×8 ビットの部分数列に分割し、それを行列に書き下した時の階数の偏りを調べる。
6	bitstream test	入力数列 2^{21} ビットから連続する 20 ビット (1 ワード) を重複させながら観測した時に、一度も現れなかったワードの個数を調べる。
7	OPSO	10 ビットを一文字とし、$2^{21} + 1$ 個の入力文字から連続する 2 文字を重複させながら観測した時に、一度も現れなかった 2 文字の個数を調べる。
8	OQSO	5 ビットを一文字とし、$2^{21} + 3$ 個の入力文字から連続する 4 文字を重複させながら観測した時に、一度も現れなかった 4 文字の個数を調べる。
9	DNA	2 ビットを一文字とし、$2^{21} + 9$ 個の入力文字から連続する 10 文字を重複させながら観測した時に、一度も現れなかった 10 文字の個数を調べる。
10	count-the-1's test on a stream of bytes	1 byte を単位としバイト内での 1 の個数で、0~2 なら A を、以下 3、4、5 なら B、C、D を、6~8 なら E を割り当てる。連続する 5 bytes 各々を A~E に置き換えたものに重複させながら観測した時の 5 文字の偏りを調べる。
11	count-the-1's test for specific bytes	1 byte を単位としバイト内での 1 の個数で、0~2 なら A を、以下 3、4、5 なら B、C、D を、6~8 なら E を割り当てる。固定位置の 5 bytes 各々を A~E に置き換えたものに重複させながら観測した時の 5 文字の偏りを調べる。
12	parking lot test	1 辺が 100 の正方形に半径 1 の円を重ねた時、すでに前に置いてある場合は新たな場所を探すという操作を繰り返す。n 個目の円を何回の試行で置くことが出来たかを調べる。
13	minimum distance test	一辺が 10,000 の正方形に 8,000 個の点を置く。これらの点の距離の最小値の偏りを調べる。

表 5.7: DIEHARD に含まれる乱数検定法 (2/2)

試験番号	検定法	概要
14	3dspheres test	一辺が 1,000 の正方形に 4,000 個の点を置く。これらの点の距離の最小値の偏りを調べる。
15	squeeze test	入力数列を 32 ビットごとに各々 [0,1) の小数と見て U とする。k の初期値を 2^{31} とし、$k = ceiling(k * U)$ を繰り返した時、$k = 1$ となるまでの回数を調べる。
16	overlapping sums test	入力数列を 32 ビットごとに各々 [0,1) の小数と見て $U(1), U(2), \ldots$ とする。$S(1) = U(1) + \cdots + U(100)$, $S(2) = U(2) + \cdots + U(101), \ldots$ の値の分布を調べる。
17	runs test	入力数列を 32 ビットごとに各々 [0,1) の小数と見て、増加あるいは、減少し続けた個数を調べる。
18	craps test	200,000 回の craps ゲームを行い、勝った回数と試合を投げた回数を調べる。

rng/rng2.html)

[7] NIST, “Randomness Testing of the Advanced Encryption Standard Candidate Algorithms,” IR 6390, Sep. 1999. (available at http://csrc.nist.gov/rng/AES-REPORT2.doc)

[8] NIST, “Randomness Testing of the Advanced Encryption Standard Finalist Candidates,” IR 6483, Apr. 2000. (available at http://csrc.nist.gov/rng/aes-report-final.doc)

[9] Ronin Software Group “DieHard randomness tester result analyzer,” (available at http://www.roninsg.com/dhrslt.htm)

第 6 章

暗号技術の実装に関わる攻撃

暗号アルゴリズムをハードウェアやソフトウェアで実装したものを暗号デバイスという。暗号デバイスに埋め込まれている暗号鍵や復号鍵などの秘密情報を暴露したり推定する攻撃を実装攻撃という。この攻撃は平文・暗号文だけから秘密情報を推定するのではなく暗号デバイスから得られる別の情報を用いて秘密情報を特定するためにサイドチャンネル攻撃と呼ばれる。近年、暗号が多方面で実用化されるにつれ実装上の安全性の観点から関心が高まっている。

本章で紹介しているような、現在提案されている実装攻撃について、そのすべてが有効であるとはいえないまでも、有効性が認められるような実装攻撃もありその対策は必要である。また、実装攻撃に関する研究は日進月歩で進んでおり、今後も新たな解読技術や防御対策が提案されると考えられるので、研究動向に十分配慮し、適切な対応が必要である。

しかしながら、ある実装攻撃による解読論文などが発表されたとしても、暗号アルゴリズム自体の安全性に直接影響するようなことはほとんどないうえ、利用環境下における適切な実装攻撃対策を施せば実用上十分な安全性が確保されると考えられるので、過度の反応をすべきではない。実装攻撃に対する安全性は、実装方式や利用環境に深く依存するため、操作性や効率をも見据えた防御対策を注意深く検討すべきである。

6.1　実装攻撃とその対策に関する調査報告書のまとめ

6.1.1　はじめに

この節では、これまでに知られている実装攻撃とその実現可能性およびそれに対する対策を述べる。本節の内容は主に、平成 14 年度に CRYPTREC より調査委託した報告書 [11] の概要を示すものである。また、まとめにあたっては、IC カード (スマートカード) の安全性に関する調査報告書 [8] も参考にした。本節に関する詳細はこれら報告書を参照にされたい。

なお、本報告書では暗号デバイスの実装上のセキュリティのみを検討対象としたので、システム全体のセキュリティは別途検討すべき課題である。

6.1.2　IC カードの概要

暗号アルゴリズムを実行する代表的な暗号デバイスは IC カードである。IC カードはプロセッサと ROM と EEPROM と少量の RAM からなる。主な目的は秘密パラメータ (鍵) を含む暗号処理を行うことであり、この秘密鍵が外部に暴露されないことが必要となる。一方攻撃者はこの秘密鍵を暴露することを目的とする。

プロセッサはチップに埋め込まれており外部とのインタフェイスは標準化されている。電源およびクロックが外部から供給され、測定可能であることが実装攻撃上重要なポイントとなる。チップ内部の回路動作およびデータが外部からは容易に測定されないような物理的な保護メカニズムを備えている。

6.1.3　実装攻撃の分類

実装攻撃は破壊的攻撃と非破壊的攻撃とに分類される。破壊的解析はパッケージを破壊して秘密データにアクセスするものであり、非破壊的解析は実行時間や電力消費量など外部から利用できる情報のみを用いる。また、実装攻撃は能動的と受動的とにも分類される。能動的攻撃はカードの通常の動作を変更して攻撃を行うものである。一方受動的攻撃はカードの通常動作を観測することによる攻撃である。

後述のプローブ攻撃や故障利用攻撃は破壊的攻撃であり、タイミング攻撃、電力解析攻撃、電磁解析攻撃は非破壊的攻撃である。破壊的攻撃よりも非破壊的攻撃のほうが低コストでできるので非破壊的攻撃への対策がより重要となる。

6.1.4　プローブ攻撃

代表的な破壊的攻撃がプローブ攻撃である。IC カードのパッケージを開封し、チップ表面のデータバスに探針 (プローブ) を当て回路を動作させながらバス上のビットの変化を観察し解析して秘密情報を得る。このために最小限必要な機器は 1 万ドルで入手できる。解析を容易にするためにクロック周波数を落とすこともある。攻撃者がこのような状況を作ってしまえば秘密情報は暗号方式を問わずほとんどのものに対して秘密情報が暴露されてしまう。

このために IC カードのチップには外部から動作を観測することを妨げるような保護層を備えたり、さらに回路を取り囲むように金属層を設け、そこに流れる電流をモニターし異常を検出したとき機密データを破壊するような保護メカニズムが備えられる。さらに攻撃者が保護層を取り除こうとするとチップ自体が破壊されるような高度な保護機構が備えられる場合もある。しかし攻撃者の能力次第で、これらの保護メカニズムを打ち破ることが不可能であるとはいえない。

6.1.5 故障利用攻撃

■攻撃の概要

故障利用攻撃とは、暗号モジュールに故障を引き起こした状態で暗号処理を行わせ、その結果から秘密データを推定する攻撃方法であり、一種の能動的攻撃である。

一般に故障は次のように分類される。故障利用攻撃はこれらの分類のうちのあるものを仮定して行われる。

永久故障と一時的故障: 将来にわたってずっと故障が続くか、ある特定の計算の時にのみ故障が起こるか。

エラーの位置: 特定の位置にエラーが起こるか、任意の位置にエラーが起こるか。

頻度: 計算中の特定時点でエラーが起こるか、任意の時点でエラーが起こるか。

エラーのタイプ: ビットまたはバイト単位で他の値に置き換わる、メモリーセルが 0 または 1 に固定化される、1 から 0 のように一方向のみにエラーが起こる、実行中にジャンプを無効化する、命令デコーダを無効化する、など。

故障モデル、すなわちどのような故障を仮定するか、によって、故障利用攻撃の実現性は大きく左右される。本節では、ある故障モデルを仮定したときに故障がどのように秘密パラメータの暴露につながるかを解析する攻撃理論と、実際の暗号デバイスにおいて故障を引き起こすための技術を述べる。

■公開鍵暗号に対する故障利用攻撃

(a) RSA 暗号系に対する故障利用攻撃

RSA 暗号系における署名生成および復号の際は、高速化のために中国人剰余定理 (Chinese Remainder Theorem、CRT) がしばしば利用される。この CRT 演算において、計算誤りを起こさせることによる攻撃法が提案されている [3, 6]。秘密鍵を p、q、d、公開鍵を $n(= pq)$、e とすると、RSA 署名生成プリミティブでは、平文 m に対する署名 s は CRT を用いて次のように計算される。

$$
\begin{aligned}
x_p &= m^d \bmod (p-1) \bmod p \\
x_q &= m^d \bmod (q-1) \bmod q \\
s &= q\left(q^{-1} \bmod p\right) x_p + p\left(p^{-1} \bmod q\right) x_q \bmod n
\end{aligned}
$$

この時、2 つのべき乗剰余演算 (x_p と x_q の計算) のうち一方に計算誤りが起こすことができたと仮定すると、次のようにして n の素因数分解が可能となる。例えば x_p の計算に誤りが起きて計算結果が x'_p になったとする。x'_p と x_q から得られる誤った署名 s' は、高い確率で次式を満たす。

$$
\begin{aligned}
s'^e &\equiv m \bmod q \\
s'^e &\not\equiv m \bmod p
\end{aligned}
$$

つまり、$s'^e - m$ は q で割り切れるが p では割り切れない。したがって、$s'^e - m$ と n との最大公約数を計算すると q となり、n の素因数である秘密鍵が容易に計算できる。

また、べき乗剰余演算中に故障を起こす攻撃だけでなく、CRT 演算における “*Recombination*” 中の故障利用攻撃も提案されている [13]。

■故障差分解析

故障差分解析は、Biham と Shamir[4] によって提案された故障利用攻撃の一種であり、暗号モジュール内で一時的故障を発生させて、暗号鍵を推定する解析法である。攻撃者は、正常動作時と誤動作時の出力暗号文の差を観測することにより、暗号鍵として取りえない鍵を特定し、鍵の絞り込みを行うことが出来る。故障差分解析は、主に共通鍵暗号に適用される。解析の手順は以下のように行う。

(1)　一つの平文に対して、正しい処理を行った際の正規暗号文を得る。また、それと同一の平文を用いて、特定の場所で一時故障を起こした際の暗号文を取得する。
(2)　両者の出力暗号文の差を観測することにより、ありえない鍵候補を見つけることができ、鍵候補から除外する。
(3)　(1)–(2) を繰り返し行い、鍵の探索空間を限定していき、最終的に鍵を一意的に求める。

彼らは適用例として DES を挙げている。DES が実装された暗号モジュール内で、最終段 (16 段目) への入力の右半分の 1 ビットに一時的故障 (ビット反転) を発生させられると仮定した場合、すべての鍵が推定可能であることを示している。DES は、S-box 参照演算の出力値に影響を与える部分鍵のビット数が少ないため、50 ~ 200 個の暗号文に対して正常動作時と誤動作時の出力暗号文の差を観測することで解析可能である。なお、14 段目、15 段目に一時的故障が発生させられるような攻撃者を仮定した際の解析方法も同様に提案されている。さらに、Triple DES においても、同様な手法で解析可能である。

楕円曲線暗号系に対する故障差分攻撃

楕円曲線暗号系の演算中に、あるレジスタにビットエラーを起こさせることによる攻撃が提案されている [2]。楕円曲線上の点 P を入力すると、内部に秘密に保持されている整数 d を用いて、スカラー倍点 dP を計算するデバイスがあるとする。このデバイスがスカラー倍演算を行う際に、楕円曲線パラメータが保持されているレジスタのあるビットを反転させる。すると、本来の曲線上の演算ではなく、別な曲線上での演算となる。その別な曲線が、もし暗号学的に弱い曲線であった場合、計算された点 dP から秘密情報 d が求められる。

■故障を引き起こすための技術

IC カードを以下のような異常環境の下で動作させることにより、故障を引き起こすことが知られている。故障を引き起こすためにパッケージの破壊を伴うかどうかにより破壊的攻撃であるか非破壊的攻撃であるかに分かれる。

- 定格外電圧の印加

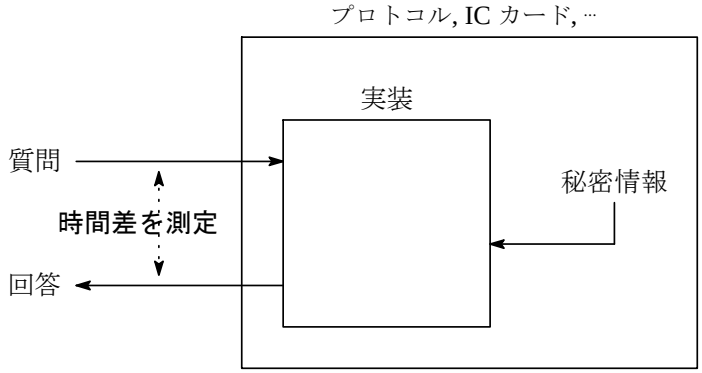

図 6.1: タイミング攻撃の原理

- 定格外のクロック周波数
- 電磁波・放射線・強力な光の照射

しかし、暗号処理の特定の個所に故障を生じさせることは容易ではない。

6.1.6 タイミング攻撃

■攻撃の概要

タイミング攻撃は、暗号デバイスの実行時間を測定することにより、暗号デバイスに格納されている秘密パラメータ (鍵) を求める攻撃である。パッケージの破壊を伴わない、代表的な非破壊的攻撃である。最初に Kocher により考案され、モンゴメリ法による RSA 暗号計算を行うデバイスに対して適用された。

攻撃者は対象となる暗号デバイスの入出力データと処理時間が入手できると仮定する。攻撃者の目的は秘密パラメータの特定である。図 6.1 にタイミング攻撃の原理図を示す。

■モンゴメリ法による RSA 暗号に対するタイミング攻撃

Montgomery のアルゴリズム [10] は、法乗算および法自乗算における剰余演算において除算命令を必要としないため、べき乗剰余演算が高速に行える。したがって、RSA 暗号系では Montgomery のアルゴリズムがしばしば用いられる。Montgomery のアルゴリズムを用いてべき乗剰余演算を行う場合に対するタイミング攻撃が提案されている [5]。

Montgomery 乗算では、演算の最後の段階で、中間結果が法 n よりも小さいか大きいかを検査し、大きい場合は法 n による剰余算を行う必要がある (この剰余算は、中間結果から n を 1 回だけ減算すればよい)。小さい場合は何もする必要が無い。したがって、与えられた法 n による Montgomery 乗算を行う場合、Montgomery 乗算への入力データによって、減算分の演算速度差が生じることになる。これを利用して、RSA 暗号系へのタイミング攻撃が可能となる [5]。

この攻撃では、攻撃者はべき乗剰余演算 $y \equiv x^d \bmod n$ が (例えば) Binary 法で計算されること

と、法乗算および法自乗算が Montgomery のアルゴリズムにより計算されることを知っていると仮定する。この仮定の元で、攻撃者はべき指数 d を最上位ビット (または最下位ビット) から 1 ビットづつ求めていく。d の i 番目のビットを求める時は、i 番目のビットで Montgomery 乗算における減算が行われる x と、行われない x とに分類して x をいくつか用意する。それらの x に対するべき乗剰余演算時間差を統計的に解析し (例えば平均時間を比較し)、ビット i が 1 か 0 かを判定する。2 種類の x を用意することは、べき乗剰余演算アルゴリズムを知っている攻撃者ならば、計算をシミュレートすることにより可能である。

■共通鍵暗号に対するタイミング攻撃

公開鍵暗号に比べて数は少ないものの、共通鍵暗号 (ブロック暗号) に対するタイミング攻撃も提案されている。4,000 回の測定により AES 暗号デバイスに格納される鍵の値を推定する方法が提案されている。

6.1.7　電力解析攻撃

■攻撃の概要

暗号デバイスの実行時間に加えて、その消費電力も暗号処理内容と秘密パラメータに関して多くの情報を与える。この考えに基づき Kocher が単純電力解析および差分電力解析による攻撃を考案した [9]。非破壊的攻撃の一種である。

IC カード接続の端末から IC カードに対してクロックと共に電源が供給されるため IC カードの消費電力は容易に測定できる。環境の整った実験室でこの電力を超高速 (1GHz 以上) かつ非常に精密 (誤差 1% 以下) にデジタル化することができる。20MHz 以上でサンプルし PC にデータを送るデバイスは 400 ドル未満で入手できる。

■単純電力解析

単純電力解析 (SPA) は暗号処理中の消費電力を直接測定する手法である。ある暗号処理のある区間に測定した消費電力データをトレースという。例えば 5MHz でサンプルされた 1 ミリ秒の動作は 5,000 ポイントのトレースを与える。図 6.2 は DES 動作を行う IC カードの SPA トレースの一例である [1]。

SPA によれば実行命令系列が明らかになるので、処理されるデータにより実行パスが異なるように実装された暗号が攻撃できる。以下に例を示す。

DES 鍵スケジュール:　DES の鍵スケジュール計算は 28 ビットの鍵レジスタの回転を含む。1 ビットシフトして外に出された 1 ビットデータの値によって条件分岐が行われる。もし実行パスが分岐によって異なるならばそれぞれの電力消費のトレースは異なる SPA 特性を示す。

DES 置換:　ビット置換にともない、ソフトウェアまたはマイクロコードでの条件付分岐によっ

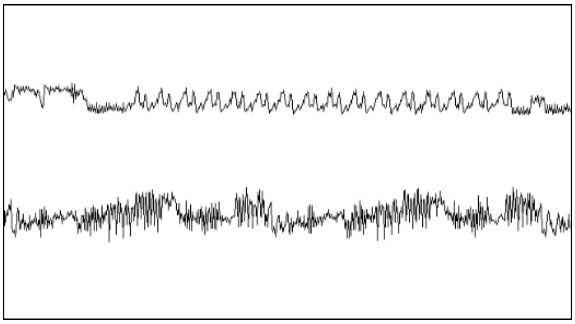

図 6.2: 典型的な IC カードで実行した DES 動作 1 段の電力測定結果

上側のトレースは、暗号化処理全体を示し、初期転置、16 段のラウンド処理、最終転置を示している。

下側のトレースは、第 2 および第 3 ラウンドの詳細電力波形である。

て 0 または 1 のビットにより消費電力が全く異なる。

比較: 系列またはメモリ内の値との比較でミスマッチが起こったときには通常条件付分岐が伴う。これから SPA (あるいはタイミング) 特性が生じる。

べき乗剰余演算: RSA 暗号系では、法 $n(= pq)$ によるべき剰余演算 $y \equiv x^d \bmod n$ (CRT を用いる場合は法 p と q によるべき乗剰余演算) が計算される。べき乗剰余演算においては、法 n での乗算と自乗算が繰り返し演算される。乗算が演算中のどの時点で行われるかは、べき指数 d (署名生成および復号における秘密鍵) に依存して決まる。Binary 法と呼ばれるべき乗剰余演算の基本的アルゴリズムでは、d を最上位ビットあるいは最下位ビットから 1 ビットずつ調べ、ビットが 1 であった場合は乗算を行い、ビットが 0 であった場合は乗算は行わない。乗算の電力波形と、自乗算の電力波形とで異なった特徴を示すことがある。したがって、電力波形を解析することにより乗算が行われたかどうか知ることができれば、秘密鍵 d の全ビットが求まることになる。また、べき乗剰余演算は、Binary 法以外にも Window 法など様々なアルゴリズムが研究されているが、これらのべき乗剰余演算アルゴリズムに対する電力解析攻撃も研究されている。

楕円曲線上のスカラ倍演算: 楕円曲線暗号系では、曲線上の点のスカラー倍演算 $Y = dG$ が行われる。スカラー倍演算は、べき乗剰余演算と同様のアルゴリズムにより行われ、点の加算と 2 倍算が繰り返し演算される。点の加算は、整数 d のビットが 1 の時に演算される。したがって RSA 暗号系への攻撃の場合と同様、電力波形を解析することにより、点加算と点 2 倍算の波形の特徴の違いを識別できれば、秘密鍵 d の全ビットが求まる。

■電力差分解析

暗号デバイスの実行時の消費電力の波形は、暗号処理内容 (実行命令系列) の情報に加え、暗号鍵に関連し、暗号デバイスに一時的に蓄えられる秘密パラメータ (内部変数) に関する情報も与える。一般的に、それらの情報はノイズ等により打ち消される場合が多い。しかし、その情報を統計的に解析することにより、暗号鍵の推定が可能な場合がある。そのような解析手法を、電力

差分解析 (DPA) と呼ぶ。攻撃者は、暗号デバイス内の一つの内部変数に着目し、その内部変数に関連する暗号鍵としてある値に仮定する。そして、その仮定した鍵の値に対応する内部変数の値を推測する。もし、仮定した鍵が正しい鍵ならば、着目する内部変数の推測が正しくなり、秘密パラメータに関する情報が観測電力の統計量に現れる。このようにして、正しい鍵を推定する。具体的な解析の流れは以下のとおりである。

(1)　異なる平文に対して暗号化処理を実施し、暗号文と消費電力の波形を取得する。
(2)　暗号鍵の一部 (数ビット) をある値に仮定する。
(3)　暗号デバイス内のある内部変数 1 ビットを、上記で仮定した暗号鍵の値から推定する。
(4)　その推定した内部変数の値が 1 か 0 かによって、暗号文を 2 つのグループに分類する。
(5)　2 つのグループの電力波形の平均値の差を計算する。
(6)　暗号鍵の一部を別の値に変えて (2)–(5) を繰り返し、一番電力差が大きいものを正しい鍵として出力する。
(7)　他の暗号鍵の一部に対しても (2)–(6) と同様の処理を行い、最終的に暗号鍵を一意的に求める。

DES は広く利用されているため、解析が進んでいる。DES は、最終段の各 S-box 参照演算の出力値に関連する暗号鍵のビット数が 6 ビットと少ないため、1,000 個の暗号文の消費電力の波形を測定することにより、すべての鍵が推定可能であると報告されている。具体的には、最終段のある一つの S-box に着目し、その S-box の入力の前に排他的論理和演算が行われる部分鍵の値 6 ビットを仮定する。そして、その仮定した鍵の値を用いて、S-box の出力の 1 ビット目を計算し、暗号文を 2 つのグループに分類し、消費電力波形の平均値の差を計算する。鍵として他の値を仮定し、同様の処理を繰り返し、もっとも消費電力波形の平均値の差が大きいものを、正しい鍵として出力する。他の最終段の S-box に関しても同様の操作を行うことで、最終的にはすべての暗号鍵を推定することが可能となる。なお、Triple DES に対しても、同様の手法で解析が可能である。

6.1.8　電磁解析攻撃

電磁解析攻撃は非破壊的攻撃を意図したものであり、暗号チップの周辺にコイルを近づけ電磁放射を測定して暗号チップの各部分の動作を推測する。得られた電磁放射量は電力波形量と同様の処理が行われる。実際の測定にあたってはチップのパッケージと保護層をはがす必要がある。この意味では破壊的攻撃の一種と呼ぶべきである。

6.1.9　対策

■プローブ攻撃への対策

プローブ攻撃と密接に関連するため、プローブ攻撃を説明する 6.1.4 節に示した。

■故障利用攻撃への対策

故障攻撃に対する自明な対策は再計算により計算結果を確かめることであるが、時間またはハードウェアのコストが増加し、永久故障の場合には対策にならない場合がある。**(a) ソフトウェア実装における対策**

公開鍵暗号系の署名生成における故障攻撃対策の 1 つは、生成された署名を、検証してみることである。故障攻撃された場合は誤った署名が生成されるので、検証不合格となり、攻撃されたことを検出できる。復号の場合も同様、復号された平文を再び暗号化し、元の暗号文と一致するかどうかを検査すればよい。RSA 暗号系においては、署名検証および暗号化で用いる公開鍵 e は小さな値 (例えば $e = 2^{16} + 1$) にできるので、e が小さい場合はこの対策を実施することによる演算量増加は比較的少ない。より効率的な対策も提案されている [12, 7]。

(b) ハードウェア実装における対策

ある演算の逆演算を行うハードウェアを追加し演算結果を確認して故障攻撃に対抗する方法が提案されているが、ハードウェアコストが増加するとともにすべての故障利用攻撃に対する対策にはなっていない。

■タイミング攻撃への対策

タイミング攻撃対策は、秘密鍵や入力データに依存した計算時間差を無くすことと、攻撃者が計算をシミュレートできないように計算の中間状態を隠蔽すること、の 2 つの基本方針に分類される。

計算時間差を無くす例として、Montgomery 乗算において剰余算 (法 n での減算) が起きるかどうかを利用した攻撃に対する対策がある。剰余算を常に (ダミー演算として) 行うことで、計算時間差を無くす方法である。ダミー演算を行う対策は、演算時間増加を伴うため、様々な改良法が提案されている。

べき乗剰余演算の計算の中間状態を隠蔽する対策としては、次のような方法がある。べき乗剰余演算 $y \equiv x^d \bmod n$ を行う前に、ランダムな組 (v_i, v_f) を $v^{-1} \equiv v_i^e \bmod n$ を満たすように選択し、x に v_i を乗じてからべき乗剰余演算し、結果に v_f を乗じる。べき乗剰余演算は、x と異なる数に基づいて進行するため、攻撃者は計算をシミュレートできない。

シフト演算を有するブロック暗号の場合、シフト命令による実行時間が、シフト量に依存しないような実装をすることが効果的である。

■電力解析攻撃および電磁解析攻撃への対策　ソフトウェア実装に関しては電力解析攻撃に対しても電磁解析攻撃に対しても対策法はほぼ同じであり、内部の秘密パラメータを隠蔽することにより外部で得られる情報を無意味にしようとするものである。

ハードウェア実装に関する対策は電力解析攻撃と電磁解析攻撃の両方に有効なものでなければならない。

(a) ソフトウェア実装における対策

ソフトウェア実装における電力解析および電磁解析攻撃への対策は以下のような方法が考えられている。しかし、以下のような対策を施すことによって攻撃を困難にすることは可能でも、完璧に阻止することは難しい。

(I)　入力値、演算ごとに消費電力差が計測出来ないようにする。
(II)　電力計測出来ないように、ノイズを発生させる。
(III)　内部変数と消費電力の相関を小さくする。
(IV)　演算命令と消費電力の相関を小さくする。
(V)　演算実行の順番をランダムにし、その演算と消費電力との相関を小さくする。
(VI)　各演算の消費電力を均一にする。
特に、RSA 暗号系のべき乗剰余演算の場合、べき指数のビットが 0 の時も 1 の時も常に乗算を実行する方法がある。しかし、この方法は計算時間の増加を生じるという問題があり、また、様々な電力解析攻撃方法に対して必ずしも安全でない。Binary 法によるべき乗演算だけでなく、Window 法など様々なべき乗剰余演算アルゴリズムに対する攻撃および対策は、数多く研究されている。
(VII)　秘密分散法を用いて計算途中の内部変数を複数の場所に格納し、内部変数を予測しずらくする。

(b) ハードウェア実装における対策

ハードウェア実装における電力解析または電磁解析攻撃への対策として以下のような方法が提案されている。しかしながら、どれも完全な対策とは言えない。

(I)　ハードウェアを非公開にする。
(II)　実装時の演算の順番をランダム化し、演算と消費電力の相関を小さくする。
(III)　ブロック暗号の場合、S-box を参照する順番を平文によって変化させる。

■楕円曲線固有の対策

SPA への対策としては、Montgomery 型、Hessian 型、Jacobi 型など、スカラ倍演算中常に加算と 2 倍算を行う曲線や、加算と 2 倍算の計算量 (定義体上の乗算回数) が同じ曲線を用いる対策が提案されている。しかし、すべての楕円曲線がこれらの型の曲線に変換できるわけではないことに注意を要する。また、通常の Weierstrass 型楕円曲線において点の加算公式を工夫する対策や、Montgomery ladder に基づく対策なども提案されている。

DPA への対策としては、スカラ倍演算ごとに鍵の展開法 (符号無し展開法や符号付展開法) や、点の表示法 (射影座標系における表示法) を変える対策が提案されている。また、同型写像を利用して、演算する曲線や定義体をランダムに変換する対策法も提案されている。

参考文献

[1] R. Anderson and M. Kuhn, Tamper resistance – a cautionary note, Proc. of the second USENIX workshop on electronic commerce (Oakland, California), Nov. 18-21 1996, pp. 1–11.

[2] I. Biehl, B. Meyer, V. Muller, "Differencial Faults Attacks on Elliptic Curve Cryptosystems," *Advances in Cryptology – CRYPTO'2000*, LNCS, **1880** (2000), Springer-Verlag, 131–146.

[3] D. Boneh, R.A. DeMilo, R.J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," *Advances in Cryptology – EUROCRYPT'97*, LNCS, **1233** (1997), Springer-Verlag, 37–51.

[4] E. Biham, and A. Shamir, "Differential fault analysis of secret key cryptsystems," *Advances in Cryptology – CRYPTO'97*, LNCS, **1294** (1997), Springer-Verlag, 513–525.

[5] J.-F. Dhen, F. Koeune, P.-A. Leroux, P. Mestré, J.-J. Quisquater, J.-L. Willems, "A Practical Implementation of the Timing Attack," *CARDIS 1998*, LNCS, (1998), Springer-Verlag.

[6] M. Joye, A.K. Lenstra, J.-J. Quisquater, "Chinese Remaindering Based Cryptosystems in the Presence of Faults," *Journal of Cryptology*, **12** (1999), No. 4, 241–245.

[7] M. Joye, P. Paillier, S.M. Yen, "Secure Evaluation of Modular Functions," *International Workshop on Cryptology and Network Security*, (2001).

[8] 情報処理振興事業協会, "平成 11 年度　スマートカードの安全性に関する調査　調査報告書," http://www.ipa.go.jp/security/fy11/report/contents/crypto/crypto/report/SmartCard/, 平成 12 年 2 月 29 日.

[9] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology – CRYPTO '99*, LNCS, **1666** (1999), Springer-Verlag, 388–397.

[10] P.L. Montgomery, "Modular Multiplication without Trial Division," *Math. Comp.*, **44** (1985), no. 170, 519–521.

[11] J.-J. Quisquater, "Side channel attacks – State-of-the-art –," 2002.

[12] A. Shamir, "How to Check Modular Exponentiation," Presented at the rump session of EUROCRYPT'97.

[13] L.Y. Wang, C.S. Laih, H.G. Tsai, N.M. Hunag, "On the Hardware Design for DES Cipher in Tamper Resistant Devices Against Differencial Fault Analysys," *IEEE international symposium on circuits and systems*, (2000).

6.2 実装攻撃に関する最近のトピック

本節では、共通鍵暗号の実装攻撃に関する最近のトピックとして 2003 年暗号と情報セキュリティシンポジウムで発表された実装攻撃と CHES2002 の紹介を行う。

6.2.1　近年の実装攻撃に関する研究動向

CHES ワークショップ (Workshop on Cryptographic Hardware and Embedded Systems) は、暗号学研究コミュニティと暗号学のアプリケーション・エリアの間のギャップを埋めるワークショップとして、1999 年に始まり 2002 年 8 月のサンフランシスコでの CHES2002 まで 4 回開催され、参加者は 200 名以上となっている。CHES では、暗号を実装したハードウェアやシステムのセキュリティ全般に関する研究成果が発表されている。例えば、いろいろな暗号技術の効率的なハードウェア実装方式、高速のソフトウェア実装方式や乱数生成器の実装と並んで暗号の解析も行われている。特に実装攻撃に関する論文は、毎年増加傾向にあり CHES2002 では発表総数 42 件の 4 割が該当する。ただし、共通鍵暗号の実装攻撃に関する報告は少なく、CHES2002 では AES 関連の解析が 2 件と DES の解析が 1 件のみである。

SCIS (Symposium on Cryptography and Information Security) は、国内で開催される情報セキュリティに関するシンポジウムとしては最大規模のものである。2003 年 1 月に浜松で開かれた SCIS2003 で 20 回目の開催となり、参加者は 400 名を越え発表件数も 200 件を越えている。SCIS2003 の発表のうち、約 1 割にあたる 20 件が暗号の実装攻撃に関するものであり、12 件が共通鍵暗号の解析である。

6.2.2　共通鍵ブロック暗号に関する攻撃のまとめ

前述のように、SCIS2003 では共通鍵ブロック暗号に対する実装攻撃が 12 件発表され、そのうち 11 件はキャッシュ攻撃に関連するものである。結果としては、電子政府推奨暗号リストに掲載された共通鍵暗号のうち、CIPHERUNICORN-E、CIPHERUNICORN-A、Hierocrypt-3、SC2000 を除くすべての暗号がある種の特殊な条件のもとで実用的な時間で秘密鍵のすべてを解読できることが報告されている[*1]。これらの報告は、第三者による評価であることから、ある程度の信頼性をもつ攻撃評価とみなすことが可能である。しかしながら、攻撃自身の適用場面が想定しにくいことと暗号実装における適切な対策を施せば防御が可能であると考えられることから、暗号アルゴリズムの安全性を直接脅かすものではないと考えるべきである。

参考文献

[1] 峯松　一彦, 角尾　幸保, 辻原　悦子, “ キャッシュに基づくサイドチャネル攻撃の理論的評価,” *SCIS2003,2D-1,*(2003).

[2] 大熊建司, 川村信一, 清水秀夫, 村谷博文, 佐野文彦, “ キャッシュミス利用実装攻撃における鍵推定の解析,” *SCIS2003,2D-2,*(2003).

[*1] フルラウンド暗号で秘密鍵が求まったものに関しては実装攻撃の評価コメントとして各々の暗号技術の部分に概要が記述されている

[3] 鶴丸豊広, 酒井康行, 反町亨, 松井充, “ 64 ビットブロック暗号に対するタイミング攻撃,” *SCIS2003,2D-3*,(2003).

[4] 青木和麻呂, 山本剛, 植田広樹, 盛合志帆, “ 128 ビットブロック暗号に対するキャッシュ攻撃,” *SCIS2003,2D-4*,(2003).

[5] 角尾幸保, 久保博靖, 茂真紀, 辻原悦子, 宮内宏, “ S-box におけるキャッシュ遅延を利用した AES へのタイミング攻撃,” *SCIS2003,3D-1*,(2003).

[6] 齊藤照夫, 角尾幸保, 洲崎智保, 宮内宏, “ S-box におけるキャッシュ遅延を利用した DES へのタイミング攻撃,” *SCIS2003,3D-2*,(2003).

[7] 角尾幸保, 川幡剛嗣, 辻原悦子, 峯松一彦, 宮内宏, “ S-box におけるキャッシュ遅延を利用した KASUMI へのタイミング攻撃,” *SCIS2003,3D-3*,(2003).

[8] 角尾幸保, 齊藤照夫, 洲崎智保, 川幡剛嗣, 宮内宏, “ S-box におけるキャッシュ遅延を利用した Camellia へのタイミング攻撃,” *SCIS2003,3D-4*,(2003).

[9] 角尾幸保, 茂真紀, 辻原悦子, 宮内宏, “ SC2000 へのタイミング攻撃,” *SCIS2003,4D-1*,(2003).

[10] 川幡剛嗣, 角尾幸保, 齊藤照夫, 辻原悦子, 宮内宏, “ Hierocrypt-L1/-3 へのタイミング攻撃,” *SCIS2003,4D-2*,(2003).

[11] 青木和麻呂, 古屋聡一, 盛合志帆, “ CIPHERUNICORN-A 実装に対する乗算の時間差を使ったタイミング攻撃,” *SCIS2003,4D-3*,(2003).

[12] 角尾幸保, 洲崎智保, 久保博靖, 辻原悦子, 宮内宏, “ S-box におけるキャッシュ遅延を利用した CIPERUNICORN-E/-A へのタイミング攻撃,” *SCIS2003,4D-4*,(2003).

[13] E. Trichina, D. De Seta, L. Germani, “ Simplified adaptive multiplicative masking for AES and its secure implementation ,” *CHES2002,pp187-197*,(2002).

[14] J. Dj. Golic, C. Tymen, “ Multiplicative masking and power analysis of AES ,” *CHES2002,pp198-212*,(2002).

[15] R. Clayton, M. Bond, “ Experience Using a Low-Cost FPGA Design to Crack DES Keys ,” *CHES2002,pp582-595*,(2002).

[16] F-X. Standaert, G. Rouvroy, J-J. Quisquater, J-D. Legat, “ A Time-Memory Tradeoff using Distinguished Points: New Analysis & FPGA Results ,” *CHES2002,pp596-611*,(2002).

[17] I. Biehl, B. Meyer, V. Muller, “Differencial Faults Attacks on Elliptic Curve Cryptosystems,” *Advances in Cryptology – CRYPTO'2000*, LNCS, **1880** (2000), Springer-Verlag, 131–146.

[18] A. Shamir, “How to Check Modular Exponentiation,” Presented at the rump session of EUROCRYPT'97.

[19] L.Y. Wang, C.S. Laih, H.G. Tsai, N.M. Hunag, “On the Hardware Design for DES Cipher in Tamper Resistant Devices Against Differencial Fault Analysys,” *IEEE international symposium on circuits and systems*, (2000).

第 7 章

2002 年度電子政府推奨暗号リストに掲載された暗号技術の知的財産情報・ライセンス方針・問い合わせ先一覧

本章の内容は、応募暗号については、応募者から修正願いがでて、受け付けられたものを除いて、すべて 2002 年 5 月時点で応募者から提出された応募書類からの抜粋である。従って、問い合わせ窓口担当者等に変更のある場合もある。

その他評価が必要とされる暗号技術については、暗号仕様などを入手するのに役立つと考えられる情報を掲載した。

7.1　公開鍵暗号技術

■DSA

仕様　ANSI X9.30:1-1997, Public Key Cryptography for The Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA) で規程されたもの。

参照 URL　http://www.x9.org/

なお、国内では同規格書は日本規格協会 (http://www.jsa.or.jp/) から入手可能である。

■ECDSA (Elliptic Curve Digital Signature Algorithm)

公開ホームページ URL

和文　http://www.labs.fujitsu.com/techinfo/crypto/ecc/

英文　http://www.labs.fujitsu.com/en/techinfo/crypto/ecc/

学会発表等　SECG(Standards for Efficient Cryptography Group)、“SECG standards” ウェブページで公開 http://www.secg.org/、2000 年 9 月 20 日

調達窓口

担当者氏名　栗田　享佳

所属・役職　ソフトウェア事業本部ミドルウェアプラットフォーム事業部　第一開発部　プロジェクト課長

企業・団体名　富士通株式会社

所在地　〒222-0033 神奈川県横浜市港北区新横浜 3-9-18(TECH ビル)

TEL　045-474-1927(内:4460)

FAX　045-474-1954

e-mail　crypto-ml@ml.soft.fujitsu.com

関連する特許権　SECG member patent letters(下記アドレス) をご参照ください。http://www.secg.org/collateral/certicom_secg_patent.pdf

関連する著作権　License to copy the document is granted provided it is identified as “Standards for Efficient Cryptography (SEC)”, in all material mentioning or referencing it.

関連する特許とその扱い　特許の扱いについては、下記の SECG Patent Policy をご参照ください。http://www.secg.org/patent_policy.htm

電子政府で使用する際のライセンス方針　富士通株式会社の所有する特許は、合理的な条件で、提供先を差別することなく、実施権を供与する。

■RSA Public-Key Cryptosystem with Probabilistic Signature Scheme (RSA-PSS)

仕様　PKCS#1 RSA Cryptography Standard (Ver.2.1)

参照 URL　http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html

公開ホームページ URL

和文　http://www.rsasecurity.com/rsalabs/submissions/index.html

英文　http://www.rsasecurity.com/rsalabs/submissions/index.html

学会発表等　Phillip Rogaway, “PSS/PSS-R (an encoding method for RSA or RW signatures)” IEEE P1363 Working Group, 1998 年 8 月

調達窓口

担当者氏名　荒井 英治

所属・役職　デベロッパ営業本部 部長

企業・団体名　RSA セキュリティ株式会社

所在地　〒100-0005 東京都千代田区丸の内 1-3-1 東京銀行協会ビルヂング 13F

TEL　03-5222-5210

FAX　03-5222-5270

e-mail　earai@rsasecurity.com

関連する特許権

関連する著作権 応募提出物にあるサンプルコードは RSA Security が著作権を所有しています。

関連する特許とその扱い

電子政府で使用する際のライセンス方針 RSA-PSS アルゴリズムに関する特許ライセンスを有しません。

■RSASSA-PKCS1-v1_5

仕様 PKCS#1 RSA Cryptography Standard (Ver.2.1)

参照 URL http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html

■RSA Public-Key Cryptosystem with Optimal Asymmetric Encryption Padding (RSA-OAEP)

仕様 PKCS#1 RSA Cryptography Standard (Ver.2.1)

参照 URL http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html

公開ホームページ URL

和文 http://www.rsasecurity.com/rsalabs/submissions/index.html

英文 http://www.rsasecurity.com/rsalabs/submissions/index.html

学会発表等 M. Bellare and P. Rogaway, “Optimal asymmetric encryption – How to encrypt with RSA” Eurocrypt’94, 1994 年

調達窓口

担当者氏名 荒井 英治

所属・役職 デベロッパ営業本部 部長

企業・団体名 RSA セキュリティ株式会社

所在地 〒100-0005 東京都千代田区丸の内 1-3-1 東京銀行協会ビルヂング 13F

TEL 03-5222-5210

FAX 03-5222-5270

e-mail earai@rsasecurity.com

関連する特許権

関連する著作権 応募提出物にあるサンプルコードは RSA Security が著作権を所有しています。

関連する特許とその扱い

電子政府で使用する際のライセンス方針 RSA-OAEP アルゴリズムに関する特許ライセンスを有しません。

■RSAES-PKCS1-v1_5

仕様　PKCS#1 RSA Cryptography Standard (Ver.2.1)

参照 URL　http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html

■DH

仕様　ANSI X9.42-2001, Public Key Cryptography for The Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography で規定されたもの。

参照 URL　http://www.x9.org/

なお、国内では同規格書は日本規格協会 (http://www.jsa.or.jp/) から入手可能である。

■ECDH (Elliptic Curve Diffie-Hellman Scheme)

公開ホームページ URL

和文　http://www.labs.fujitsu.com/techinfo/crypto/ecc/

英文　http://www.labs.fujitsu.com/en/techinfo/crypto/ecc/

学会発表等　SECG(Standards for Efficient Cryptography Group)、“SECG standards” ウェブページで公開 http://www.secg.org/、2000 年 9 月 20 日

調達窓口

担当者氏名　栗田　享佳

所属・役職　ソフトウェア事業本部ミドルウェアプラットフォーム事業部 第一開発部 プロジェクト課長

企業・団体名　富士通株式会社

所在地　〒222-0033 神奈川県横浜市港北区新横浜 3-9-18(TECH ビル)

TEL　045-474-1927(内:4460)

FAX　045-474-1954

e-mail　crypto-ml@ml.soft.fujitsu.com

関連する特許権　SECG member patent letters(下記アドレス) をご参照ください。http://www.secg.org/collateral/certicom_secg_patent.pdf

関連する著作権　License to copy the document is granted provided it is identified as “Standards for Efficient Cryptography (SEC)”, in all material mentioning or referencing it.

関連する特許とその扱い　特許の扱いについては、下記の SECG Patent Policy をご参照ください。http://www.secg.org/patent_policy.htm

電子政府で使用する際のライセンス方針　富士通株式会社の所有する特許は、合理的な条件で、提供先を差別することなく、実施権を供与する。

■PSEC-KEM Key agreement

公開ホームページ URL

和文 http://info.isl.ntt.co.jp/

英文 http://info.isl.ntt.co.jp/

学会発表等 岡本 龍明、“公開鍵暗号「EPOC」および「PSEC」” ISEC 研究会、2000 年 5 月 25 日

調達窓口

担当者氏名 中尾 昌善

所属・役職 NTT 情報流通プラットフォーム研究所主幹研究員

企業・団体名 日本電信電話株式会社 (NTT)

所在地 〒239-0847 神奈川県横須賀市光の丘 1-1-609A

TEL 0468-59-3334

FAX 0468-59-3365

e-mail nakao@isl.ntt.co.jp

関連する特許権

1. 特許出願番号 (公開番号) 10-320172
   名称 ランダム関数利用公開鍵暗号の暗号装置、復号装置
   出願日 1998 年 11 月 11 日
2. 特許出願番号 (公開番号) 2000-32461
   名称 暗号化装置、方法、復号装置、方法、暗号システム及びプログラムを記憶した記憶媒体
   出願日 2000 年 2 月 9 日

関連する著作権 「(2) 暗号技術仕様書」「(3) 自己評価書」「(5) 参照プログラムおよび その仕様書, テストベクトル生成プログラムおよびその仕様書」「(7) 応募暗号説明会発表資料」の著作権は日本電信電話株式会社に帰属します。

関連する特許とその扱い 弊社の知る範囲では、関連する他社の特許はありません。

電子政府で使用する際のライセンス方針 上記特許に関しては、応募暗号技術 (PSEC-KEM) を使用するものに対して、相互主義の下に、非排他的に、無償で実施許諾する方針です。

7.2 共通鍵暗号技術

■CIPHERUNICORN-E

公開ホームページ URL

和文 http://www.hnes.co.jp/products/security/index.html

英文 http://www.hnes.co.jp/products/security/index-e.html

学会発表等 日本電気株式会社、“登録番号 19、登録日 1998.7.6、アルゴリズム公開登録” ISO/IEC 9979 暗号アルゴリズム登録制度、1998 年 7 月 6 日

日本電気株式会社、“統計的手法により安全性が評価された暗号” 1998 年暗号と情報セキュリティシンポジウム、SCIS’98-4.2.B、1998 年 1 月 29 日

調達窓口

担当者氏名 セキュリティ技術センター

所属・役職 インターネットソフトウェア事業部

企業・団体名 日本電気株式会社

所在地 〒108-8557 港区芝浦 2-11-5

TEL 03-5476-1913

FAX 03-6576-1678

e-mail sec@isd.nec.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 出願平 9-213274

   名称 暗号装置及び暗号装置を実現するプログラムを記録したコンピューターが読みとり可能な記録媒体

   出願日 平成 9 年 (1997)8 月 7 日

関連する著作権 著作物 CIPHERUNICORN-E のプログラム

商標 登録番号 第 4221077 号

関連する特許とその扱い 現時点までに発行されている特許公報からは、関連する他社先行特許は発見できませんでした。

電子政府で使用する際のライセンス方針 民間企業等による営利目的の使用の場合を除き、無償とする考えです。

■Hierocrypt-L1

公開ホームページ URL

和文 http://www.toshiba.co.jp/rdc/security/hierocrypt

英文 http://www.toshiba.co.jp/rdc/security/hierocrypt

学会発表等 大熊建司、“ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 に対する強度／性能評価” 電子情報通信学会 信学技法 ISEC2000-71 pp.71-100、2000 年 9 月 29 日

調達窓口

担当者氏名 大熊 建司

所属・役職 研究開発センター コンピュータ・ネットワーク ラボラトリー 主任研究員

企業・団体名 (株) 東芝

所在地 〒212-8582 神奈川県川崎市幸区小向東芝町 1

TEL 044-549-2156

FAX 044-520-1841

e-mail kenji.ohkuma@toshiba.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 特願 2000-210484
   名称 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに演算装置」
   出願日 2001/03/06
2. **特許出願番号 (公開番号)** 特願 2000-211686
   名称 「暗号化装置、復号装置及び拡大鍵生成装置、拡大鍵生成方法並びに記録媒体」
   出願日 2000/07/12
3. **特許出願番号 (公開番号)** 特願 2000-212175
   名称 「パラメータ決定装置、パラメータ決定方法、暗号化装置、および復号装置」
   出願日 2000/07/13
4. **特許出願番号 (公開番号)** 特願 2001-68742
   名称 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに記憶媒体」
   出願日 2001/06/30

関連する著作権

関連する特許とその扱い 非排他的かつ妥当な条件で実施を許諾します。

電子政府で使用する際のライセンス方針 非排他的かつ妥当な条件で実施を許諾します。

■MISTY1

公開ホームページ URL

和文 http://www.security.melco.co.jp/misty

英文 http://www.security.melco.co.jp/misty

学会発表等 松井 充、“ブロック暗号アルゴリズム MISTY” 電子情報通信学会 ISEC 研究会、1996 年 7 月 22 日

調達窓口

担当者氏名 小松田 敏二

所属・役職 インフォメーションシステム事業推進本部 情報セキュリティ推進センター 副センター長

企業・団体名 三菱電機株式会社

所在地 〒100-8301 東京都千代田区丸の内 2-2-3 (三菱電機ビル)

TEL 03-3218-3221

FAX 03-3218-3638

e-mail Binji.Komatsuda@hq.melco.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 特許第 3035358 号
   名称 データ変換装置及びデータ変換方法
   出願日 2000.2.18

上記特許は、PCT/JP96/01254 (PCT 出願 1996 年 7 月 31 日) にも出願中。

関連する著作権　提出書類の著作権は三菱電機株式会社に帰属します。

関連する特許とその扱い　弊社の知る範囲では、関連する他社特許はありません。

電子政府で使用する際のライセンス方針　上記特許に関しては、応募暗号技術 (MISTY1) を使用するものに対して相互主義の下に、非排他的に、無償で実施許諾する方針です。

■Triple DES

仕様　FIPS PUB 46-3, Data Encryption Standard (DES)

参照 URL　http://csrc.nist.gov/CryptoToolkit/tkencryption.html

■AES

仕様　FIPS PUB 197, Advanced Encryption Standard (AES)

参照 URL　http://csrc.nist.gov/CryptoToolkit/tkencryption.html

■Camellia

公開ホームページ URL

和文　http://info.isl.ntt.co.jp/camellia/

英文　http://info.isl.ntt.co.jp/camellia/

学会発表等　神田雅透、“128 ビットブロック暗号 Camellia” ISEC 研究会、2000 年 5 月 25 日

調達窓口

担当者氏名　中尾 昌善

所属・役職　情報流通プラットフォーム研究所 情報セキュリティプロジェクト (セ制 G) グループリーダ

企業・団体名　日本電信電話株式会社

所在地　〒239-0847 横須賀市光の丘 1-1-609A

TEL　0468-59-3334

FAX　0468-59-3365

e-mail　nakao@isl.ntt.co.jp

担当者氏名　豊嶋 淳

所属・役職　NTT 事業部第一部長

企業・団体名 三菱電機株式会社
所在地 〒104-6212 中央区晴海 1-8-12 オフィスタワー Z13 階
TEL 03-6221-2634
FAX 03-6221-2770
e-mail toshima@npd.hon.melco.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 2000-064614
   名称 データ変換装置及びデータ変換方法及びデータ変換方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体
   出願日 2000 年 3 月 9 日

上記特許は、PCT/JP01/01796 (PCT 出願, 2001 年 3 月 8 日) および第 90105464 号 (台湾, 2001 年 3 月 8 日) にも出願中。

関連する著作権 「(2) 暗号技術仕様書」「(3) 自己評価書」「(7) 応募暗号説明会発表資料」の著作権は日本電信電話株式会社および三菱電機株式会社に,「(5) 参照プログラムおよびその仕様書, テストベクトル生成プログラムおよびその仕様書」の著作権は三菱電機株式会社に帰属します。

関連する特許とその扱い 弊社の知る範囲では、関連する他社の特許はありません。

電子政府で使用する際のライセンス方針 上記特許に関しては、応募暗号技術 (Camellia) を使用するものに対して、相互主義の下に、非排他的に、無償で実施許諾する方針です。

■CIPHERUNICORN-A

公開ホームページ URL

和文 http://www.hnes.co.jp/products/security/index.html
英文 http://www.hnes.co.jp/products/security/index-e.html

学会発表等 日本電気株式会社、“信学技報 Vol.100 No.76 pp23-46, ISEC2000-5「A New 128-bit Block Cipher CIPHERUNICORN-A」” 電子情報通信学会 ISEC 研究会、2000 年 5 月 26 日

調達窓口

担当者氏名 セキュリティ技術センター
所属・役職 インターネットソフトウェア事業部
企業・団体名 日本電気株式会社
所在地 〒108-8557 港区芝浦 2-11-5
TEL 03-5476-1913
FAX 03-6576-1678
e-mail sec@isd.nec.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 出願平 9-213274
   名称 暗号装置及び暗号装置を実現するプログラムを記録したコンピューターが読み

とり可能な記録媒体

出願日 平成 9 年 (1997)8 月 7 日

関連する著作権 著作物 CIPHERUNICORN-A のプログラム

商標 登録番号 第 4221077 号

関連する特許とその扱い 現時点までに発行されている特許公報からは、関連する他社先行特許は発見できませんでした。

電子政府で使用する際のライセンス方針 民間企業等による営利目的の使用の場合を除き、無償とする考えです。

■Hierocrypt-3

公開ホームページ URL

和文 http://www.toshiba.co.jp/rdc/security/hierocrypt

英文 http://www.toshiba.co.jp/rdc/security/hierocrypt

学会発表等 大熊建司、“ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 に対する強度／性能評価” 電子情報通信学会 信学技法 ISEC2000-71 pp.71-100、2000 年 9 月 29 日

調達窓口

担当者氏名 大熊 建司

所属・役職 研究開発センター コンピュータ・ネットワーク ラボラトリー 主任研究員

企業・団体名 (株) 東芝

所在地 〒212-8582 神奈川県川崎市幸区小向東芝町 1

TEL 044-549-2156

FAX 044-520-1841

e-mail kenji.ohkuma@toshiba.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 特願 2000-210484
   名称 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに演算装置」
   出願日 2001/03/06
2. **特許出願番号 (公開番号)** 特願 2000-211686
   名称 「暗号化装置、復号装置及び拡大鍵生成装置、拡大鍵生成方法並びに記録媒体」
   出願日 2000/07/12
3. **特許出願番号 (公開番号)** 特願 2000-212175
   名称 「パラメータ決定装置、パラメータ決定方法、暗号化装置、および復号装置」
   出願日 2000/07/13
4. **特許出願番号 (公開番号)** 特願 2001-68742
   名称 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに記憶媒体」
   出願日 2001/06/30

関連する著作権

関連する特許とその扱い 非排他的かつ妥当な条件で実施を許諾します。

電子政府で使用する際のライセンス方針 非排他的かつ妥当な条件で実施を許諾します。

■SC2000

公開ホームページ URL

和文 http://www.labs.fujitsu.com/theme/crypto/sc2000_j.html

英文 http://www.labs.fujitsu.com/theme/crypto/sc2000.html

学会発表等 下山 武司、“共通鍵ブロック暗号 SC2000 (ISEC2000 - 72)” 電子情報通信学会情報セキュリティ研究会、2000 年 9 月 29 日

調達窓口

担当者氏名 栗田 享佳

所属・役職 ソフトウェア事業本部運用管理ミドルウェアプラットフォーム事業部 第一開発部 プロジェクト課長

企業・団体名 富士通株式会社

所在地 〒222-0033 神奈川県横浜市港北区新横浜 3-9-18(TECH ビル)

TEL 045-474-1927(内:4460)

FAX 045-474-1954

e-mail crypto-ml@ml.soft.fujitsu.com

関連する特許権

1. **特許出願番号 (公開番号)** 2001-018016
   名称 暗号設計装置、暗号設計プログラム及び記録媒体
   出願日 2000.1.26
2. **特許出願番号 (公開番号)** 2000-212813
   名称 F 関数内部に SPN 構造を用いた演算装置及び演算方法
   出願日 2000.7.13
3. **特許出願番号 (公開番号)** 2000-212814
   名称 Feistel 構造と SPN 構造とを組み合わせた演算装置及び演算方法
   出願日 2000.7.13
4. **特許出願番号 (公開番号)** 2000-212482
   名称 拡大鍵生成装置、拡大鍵生成プログラムおよび記録媒体
   出願日 2000.7.13

関連する著作権 富士通株式会社

関連する特許とその扱い なし

電子政府で使用する際のライセンス方針 上記特許、及び著作権は、合理的な条件で、提供先を差別することなく、実施権を供与する

■MUGI

公開ホームページ URL

和文 http://www.sdl.hitachi.co.jp/crypto/mugi/

英文 http://www.sdl.hitachi.co.jp/crypto/mugi/index-e.html

学会発表等 渡辺大、“PANAMA 型擬似乱数生成器の乱数性” 電子情報通信学会 ISEC 研究会、2001 年 9 月 17 日

調達窓口

担当者氏名 原野 紳一郎

所属・役職 (株) 日立製作所ソフトウェア事業部 システム管理ソフトウェア本部 推進部長

企業・団体名 (株) 日立製作所ソフトウェア事業部 システム管理ソフトウェア本部

所在地 〒244-8555 神奈川県横浜市戸塚区戸塚町 5030 番地

TEL (045)862-8715

FAX (045)865-9010

e-mail harano@itg.hitachi.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 特願 2001-013959
   名称 疑似乱数生成方法またはそれを用いた暗号復号処理装置
   出願日 2000 年 1 月 23 日
2. **特許出願番号 (公開番号)** 特願 2001-145783
   名称 疑似乱数生成方法またはそれを用いた暗号復号処理装置
   出願日 2000 年 5 月 16 日
3. **特許出願番号 (公開番号)** 特願 2001-274433
   名称 疑似乱数生成方法またはそれを用いた暗号復号処理装置
   出願日 2000 年 9 月 11 日

関連する著作権 本暗号技術 MUGI を応募するにあたり提出される文書、ソースコードなどの著作権はすべて日立製作所に帰属します。

関連する特許とその扱い 弊社は、MUGI の技術に関し、弊社による上記出願特許および登録特許が関連すると考えられます。弊社と致しましては、MUGI が本提案に採用された場合には、上記特許を非差別的、かつ適正な対価条件でライセンス致します。ただし、相手方が相手方の MUGI 関連特許を非差別的、かつ適正な対価条件でライセンスしない場合はこの限りではございません。

電子政府で使用する際のライセンス方針 同上

■MULTI-S01

公開ホームページ URL

和文 http://www.sdl.hitachi.co.jp/crypto/s01/index-j.html

英文 http://www.sdl.hitachi.co.jp/crypto/s01/index.html

学会発表等 古屋 聡一、“MULTI-S01 のパディングと安全性についての考察” 電子情報通信学会 ISEC 研究会、2000 年 9 月 29 日

調達窓口

担当者氏名 原野 紳一郎

所属・役職 (株) 日立製作所ソフトウェア開発本部 推進部長

企業・団体名 (株) 日立製作所ソフトウェア開発本部

所在地 〒244-8555 神奈川県横浜市戸塚区戸塚町 5030 番地

TEL (045)866-8140

FAX (045)865-9036

e-mail harano@itg.hitachi.co.jp

関連する特許権

1. **特許出願番号 (公開番号)** 特願 2000-108334 号 (特開 2001-007800)
   名称 「暗号化装置および方法」
   出願日 2000 年 4 月 10 日
2. **特許出願番号 (公開番号)** 特願 2000-070994 号
   名称 「共通鍵暗号方法及び装置」
   出願日 2000 年 3 月 9 日
3. **特許出願番号 (公開番号)** 特願 2000-210690 号
   名称 「共通鍵暗号方法及び装置」
   出願日 2000 年 7 月 6 日

関連する著作権 本暗号技術 MULTI-S01 を応募するにあたり提出される文書、ソースコードなどの著作権はすべて日立製作所に帰属します。

関連する特許とその扱い 弊社は、MULTI-S01 の技術に関し、弊社による上記の出願特許および登録特許が関連すると考えられます。弊社と致しましては、MULTI-S01 が本提案に採用された場合には、上記特許を非差別的、かつ適正な対価条件でライセンス致します。ただし、相手方が相手方の MULTI-S01 関連特許を非差別的、かつ適正な対価条件でライセンスしない場合はこの限りではございません。

電子政府で使用する際のライセンス方針 上記載の内容に同じです。

■RC4

問い合わせ先 RSA セキュリティ社 (http://www.rsasecurity.co.jp/)

仕様 RC4 のアルゴリズムについては、RSA Laboratories が発行した CryptoBytes 誌 (Volume 5, No.2, Summer/Fall 2002) に掲載された次の論文に記載されているもの。

Fluhrer, Scott, Itsik Mantin, and Adi Shamir, “Attacks On RC4 and WEP”, CryptoBytes,

Volume 5, No.2, Summer/Fall 2002

参照 URL　http://www.rsasecurity.com/rsalabs/cryptobytes/index.html

7.3　ハッシュ関数

■RIPEMD-160

参照 URL　http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html

■SHA-1, SHA-256, SHA-384, SHA-512

仕様　FIPS PUB 186-2, Secure Hash Standard (SHS)

参照 URL　http://csrc.nist.gov/CryptoToolkit/tkhash.html

7.4　擬似乱数生成系

■PRNG in ANSI X9.42-2001 Annex C.1/C.2

仕様　ANSI X9.42-2001, Public Key Cryptography for The Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography

参照 URL　http://www.x9.org/

なお、国内では同規格書は日本規格協会 (http://www.jsa.or.jp/) から入手可能である。

■PRNG in ANSI X9.62-1998 Annex A.4

仕様　ANSI X9.62-1998, Public Key Cryptography for The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)

参照 URL　http://www.x9.org/

なお、国内では同規格書は日本規格協会 (http://www.jsa.or.jp/) から入手可能である。

■PRNG in ANSI X9.63-2001 Annex A.4

仕様 ANSI X9.63-2001, Public Key Cryptography for The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography

参照 URL http://www.x9.org/

なお、国内では同規格書は日本規格協会 (http://www.jsa.or.jp/) から入手可能である。

■PRNG for DSA in FIPS PUB 186-2 Appendix 3

仕様 FIPS PUB 186-2, Digital Signature Standard (DSS)

参照 URL http://csrc.nist.gov/CryptoToolkit/tkrng.html

■PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1

仕様 FIPS PUB 186-2, Digital Signature Standard (DSS)

参照 URL http://csrc.nist.gov/CryptoToolkit/tkrng.html

■PRNG in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1/3.2

仕様 FIPS PUB 186-2, Digital Signature Standard (DSS)

参照 URL http://csrc.nist.gov/CryptoToolkit/tkrng.html

第 8 章

評価対象暗号技術一覧

1. 公開鍵暗号技術
   （a）署名
      i. DSA
      2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、電子署名法等で利用される暗号技術として詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。
      ii. ECDSA (ANSI X9.62)
      2001 年度、電子署名法等で利用される暗号技術として詳細評価を実施し、電子政府暗号候補とされた。2002 年度、楕円曲線の選択法の検討において、ECDSA in SEC 1 に一本化された。評価結果については本報告書第 2 章を参照。
      iii. ECDSA (Elliptic Curve Digital Signature Algorithm) in SEC 1 [ECDSA]
      2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子署名法等で利用される暗号技術として詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。
      iv. ESIGN [ESIGN]
      2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、仕様の変更に伴い、公募により提案された暗号技術であり、電子署名法等で利用される暗号技術として詳細評価を実施し、2002 年度詳細評価対象暗号候補とされた。2002 年度、詳細評価を実施し、証明可能安全性を持たないと判断されたため、評価を終了した。評価結果については本報告書第 2 章を参照。
      v. TSH-ESIGN
      2002 年度、暗号技術評価委員会において参考のため評価が必要であると判断された暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。
      vi. RSA Public-Key Cryptosystem with Probabilistic Signature Scheme (RSA-PSS) [RSA-PSS]

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、かつ、公募により提案された暗号技術であり、詳細評価を実施した。2001 年度、電子署名法等で利用される暗号技術として詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

vii. RSASSA-PKCS1-v1_5

2001 年度、電子署名法等で利用される暗号技術として詳細評価を実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

（b）守秘

i. ECIES (Elliptic Curve Integrated Encryption Scheme) in SEC 1 [ECIES]

2000 年度、ECAES という名称で公募により提案された暗号技術で詳細評価を実施した。2001 年度、ECIES という名称に変更された。ISO/IEC 18033-2 における審議を皮切りに、2002 年度詳細評価対象暗号候補とされた。2002 年度、詳細評価を実施し、証明可能安全性を持たないと判断されたため、評価を終了した。評価結果については本報告書第 2 章を参照。

ii. HIME(R)(High Performance Modular Squaring Based Public Key Encryption (Revised version)) [HIME(R)]

2001 年度、公募により提案された暗号技術であり、スクリーニング評価を実施し、2002 年度詳細評価対象暗号候補とされた。2002 年度、詳細評価を実施し、2002 年 9 月時点で証明可能安全性を有すると断定できないと判断されたため、評価を終了した。評価結果については本報告書第 2 章を参照。

iii. RSA Public-Key Cryptosystem with Optimal Asymmetric Encryption Padding (RSA-OAEP) [RSA-OAEP]

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、かつ、公募により提案された暗号技術であり、詳細評価を実施した。2001 年度、詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

iv. RSAES-PKCS1-v1_5

2002 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施し、注釈付きで電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

（c）鍵共有

i. DH

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

ii. ECDH (Elliptic Curve Diffie-Hellman Scheme) in SEC 1 [ECDH]

2000 年度、ECDHS という名称で公募により提案された暗号技術で詳細評価を実施した。2001 年度、ECDH という名称に変更された。電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

iii. PSEC-KEM Key agreement [PSEC-KEM]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、仕様の変更に伴い、公募により提案された暗号技術であり、スクリーニング評価を実施し、2002 年度詳細評価対象暗号候補とされた。2002 年度、詳細評価を実施し、注釈付きで電子政府推奨暗号リストに記載された。評価結果については本報告書第 2 章を参照。

2. 共通鍵暗号技術

（a）64 ビットブロック暗号

i. CIPHERUNICORN-E [UNI-E]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

ii. Hierocrypt-L1 [HC-L1]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

iii. MISTY1 [MISTY1]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

iv. Triple DES

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、注釈付きで電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

（b）128 ビットブロック暗号

i. Advanced Encryption Standard (AES)

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、FIPS 化されたことを受け、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

ii. Camellia [Camellia]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

iii. CIPHERUNICORN-A [UNI-A]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

iv. Hierocrypt-3 [HC-3]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

v. RC6 Block Cipher [RC6]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、応募者からの通知により 2002 年 10 月で評価を終了した。評価結果については本報告書第 3 章を参照。

vi. SC2000 [SC2000]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

（c）ストリーム暗号

i. MUGI

2001 年度、公募により提案された暗号技術でスクリーニング評価を実施し、2002 年度詳細評価対象暗号候補とされた。2002 年度、詳細評価を実施し、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

ii. MULTI-S01 [S01]

2000 年度、公募により提案された暗号技術で詳細評価を実施した。2001 年度、詳細評価を継続して実施し、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

iii. RC4

2001 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、2002 年度詳細評価対象暗号候補とされた。2002 年度、詳細評価を実施し、注釈付きで電子政府推奨暗号リストに記載された。評価結果については本報告書第 3 章を参照。

3. ハッシュ関数

（a）RIPEMD-160

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 4 章を参照。

（b）SHA-1

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、電子政府暗号候補とされた。2002 年度、電子政府推奨暗号リストに記載された。評価結果については本報告書第 4 章を参照。

（c）SHA-256, SHA-384, SHA-512

2001 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術で

あり、詳細評価を実施し、電子政府暗号候補とされた。2002 年度、ドラフト版の仕様と同一のものが FIPS 化されたことを受け、電子政府推奨暗号リストに記載された。評価結果については本報告書第 4 章を参照。

4. 擬似乱数生成系

（a）PRNG in ANSI X9.42-2001 Annex C.1/C.2

2002 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施し、電子政府推奨暗号リストにおいて SHA-1 ベースの Annex C.1 が例示として記載された。評価結果については本報告書第 5 章を参照。

（b）PRNG in ANSI X9.62-1998 Annex A.4

2002 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。評価結果については本報告書第 5 章を参照。

（c）PRNG in ANSI X9.63-2001 Annex A.4

2002 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。評価結果については本報告書第 5 章を参照。

（d）PRNG for DSA in FIPS PUB 186-2 Appendix 3

2000 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施した。2001 年度、電子政府暗号候補とされた。評価結果については本報告書第 5 章を参照。

（e）PRNG for general purpose in FIPS PUB 186-2 (+ change notice 1) Appendix 3.1

2002 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施し、電子政府推奨暗号リストにおいて SHA-1 ベースのものが例示として記載された。評価結果については本報告書第 5 章を参照。

（f）PRNG in FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1/3.2

2002 年度、暗号技術評価委員会において評価が必要であると判断された暗号技術であり、詳細評価を実施し、電子政府推奨暗号リストにおいて汎用目的の SHA-1 ベースの Appendix 3.1 が例示として記載された。評価結果については本報告書第 5 章を参照。

索引

発行日 2003(平成 15) 年 3 月

発行者

- 〒113-6591
  東京都文京区本駒込二丁目 28 番 8 号
  (文京グリーンコートセンターオフィス 16 階)
  情報処理振興事業協会 (セキュリティセンター)
  電話: 03-5978-7508, FAX: 03-5978-7518

  INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
  BUNKYO GREEN COURT CENTER OFFICE
  2-28-8 HONKOMAGOME, BUNKYO-KU
  TOKYO, 113-6591 JAPAN
  TEL: +81-3-5978-7508, FAX: +81-3-5978-7518 (IT SECURITY CENTER)
  EMAIL: cryptrec@ipa.go.jp
- 〒105-0014
  東京都港区芝二丁目 31 番 19 号
  (バンザイビル)
  通信・放送機構 研究企画管理部研究企画課
  電話: 03-3769-6810, FAX: 03-5441-7584

  TELECOMMUNICATIONS ADVANCEMENT ORGANIZATRION of JAPAN
  BANZAI Bld. 2-31-19 SHIBA, MINATO-KU
  TOKYO, 105-0014 JAPAN
  TEL: +81-3-3769-6810, FAX: +81-3-5441-7584
  EMAIL: cryptrec@shiba.tao.go.jp