暗号技術評価報告書

CRYPTREC Report 2000

平成１３年３月
情報処理振興事業協会
セキュリティセンター

目次

はじめに

ミレニアム・プロジェクトとして 2003 年度までにその基盤が構築される世界最高水準の電子政府は、行政の効率化や国民負担の軽減を目標に、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ものである。

電子政府の機能を実現するためには、様々なところで暗号技術を使う必要がある。暗号技術については、現在その大多数が計算量的困難性に立脚したアルゴリズム公開型のものとなっているが、そのため、コンピュータの能力の向上に伴い、安全性に対するマージンが年々低下するという宿命がある。また、暗号技術の進歩に伴い、新たな攻撃方法が発見され、解読困難性が設計上の想定よりも低いことが明らかになるケースもある。さらに、実用上から安全性以外にも、速度やプログラムの大きさなど、実装性能も考慮して使うべき暗号の選択を行う必要がある。しかしながら、我が国においては、これまで各種の暗号技術について、十分な評価を体系的に行った経験がなかった。

このようなことから、電子政府構築において、先導的官庁の役割を担う経済産業省（旧通商産業省）は、電子政府において利用可能な暗号技術のリストを作成することが必要であるとして、情報処理振興事業協会（IPA）にその評価の業務を委託した。

我が国の暗号技術の最高水準の研究者等の協力を得て、我が国初の試みである暗号技術の評価という事業を行うために、東京大学今井秀樹教授を委員長とし、我が国の最高水準の暗号研究者とセキュリティ技術関係者からなる暗号技術評価委員会（CRYPTREC）を組織し、約 1 年の期間、密度の濃い評価プログラムを実施してきた。

委員会は、経済産業省（旧通商産業省）だけでなく、総務省（旧総務庁、及び旧郵政省）、防衛庁が最初からオブザーバとして参加し、その後、内閣官房、警察庁、法務省、財務省がオブザーバとして加わった、政府横断的なものとなった。

暗号技術をめぐる世界の動きを見ると、米国では DES 暗号に代わる次世代の米国政府標準暗号を定める AES プログラムが進行中である。また、欧州でも欧州版暗号評価プロジェクト(NESSIE)が開始された。さらに ISO/IEC の場でも、暗号技術の国際標準を策定しようとの動きがある。

このような世界情勢の中で、今回の評価事業は、大きな意味を持つ画期的なことであった。さらに我が国暗号技術研究の向上にも資するものであったと考えている。

本報告書が我が国政府の電子政府構築に有益な指針を与えることを期待するとともに、電子商取引などにも価値ある情報を提供できたと考えている。

最後に、一年間の長きにわたって暗号技術評価委員会（CRYPTREC）の推進にご尽力いただいた今井委員長、大所高所から貴重な意見を頂いた辻井顧問、小委員長の重責を担っていただいた金子小委員長と松本小委員長、ほか委員の皆様、そして本事業を進めるにあたってご指導いただいたオブザーバの皆様に感謝するものである。

平成 13 年 3 月

情報処理振興事業協会
セキュリティセンター所長　小林　正彦

報告書作成にあたって

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（CRYPTREC）の１年間の活動の成果である。多くの暗号技術を短期間で厳正に評価することは決して容易な作業ではない。しかし、このような評価が、電子政府の構築、ひいては２１世紀におけるわが国のネットワーク社会の健全な発展に不可欠であり、歴史的な意義を持つ事業であるとの共通の認識のもとに、関係者の方々には献身的なご協力を頂いた。本報告書には、電子政府で利用可能な暗号技術の評価に関し現時点で望み得る最も適正な情報が盛り込まれていると考えている。今後の電子政府の構築に向けて、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委員会の活動に先立ち、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（提案者：山岸篤弘氏)」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。

そこで、この報告を受けた形で、平成 12 年 5 月に情報処理振興事業協会の「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、暗号技術の信頼性等を評価することになった。

ネットワーク社会における電子政府システムは、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。情報セキュリティ技術が電子政府を支える基盤技術であることは紛れもない事実なのである。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である。

特に、暗号技術の安全性評価については、暗号アルゴリズムを完全に公開して行わない限り意味のある結果は得られないが、たとえ暗号アルゴリズムを完全に公開し、一定期間内にそれに対する攻撃法の公表が無かったとしても、それで安全性が保証される訳でもない。しかし、電子政府システムを構築するために、現在の技術によって、安全性のレベルを明らかにすることは必須である。しかも、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評価を行わなければならないのは、当然の責務である。暗号技術評価委員会は、この責務の一翼を担うべく設立されたものであり、その役割は極めて重いといえよう。

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非常に困難である。また、将来にわたった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だろう。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。

本報告書は、電子政府に利用可能な暗号技術に関し、現時点で望み得る最良の評価結果が示されており、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。しかし、今回の暗号技術評価事業は、米国 AES の公募選定事業という範があったとはいえ、我が国では初めての事業であり、今回の成果で全てが完了したという訳ではない。1 年間という異例に短い期間で評価作業を行ったため、現時点で利用可能な技術による評価であり、その内容には至らぬ点もあろうと考える。この不足分を補完するとともに、日々進歩を遂げる暗号技術に追随することが今後の重要な課題である。

また、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。

今回の暗号技術評価委員会および共通鍵暗号評価小委員会、公開鍵暗号評価小委員会には、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、この暗号技術評価委員会を自らのものとされ、その事業に献身的に参画いただいた。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、小委員会委員長として本報告書の取りまとめに多大なご尽力いただいた。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。この紙面を借りて各委員に謝意を表する。また、本委員会および小委員会には、評価対象となった暗号の設計者も含まれていた。これは、我が国における極めて限られた数の暗号研究者の中から、本事業に欠かすことのできない人材を選ぶ場合避け難い状況である。しかし、これらの委員の方々は、難しい立場であるにもかかわらず、本事業の目的のために、自らの利害を超え、公正な観点からご協力いただいた。重ねて謝意を表したい。

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。

平成 13 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章から3章には、本暗号技術評価事業の概要や暗号評価方法を、4章から7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4章、ブロック暗号やストリーム暗号の共通鍵暗号技術は5章、ハッシュ関数は6章、擬似乱数生成法は7章に記述した。また、参考情報として、8章に世界の暗号標準化の動向について簡単にまとめた。

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、その安全性評価に関して、将来にわたっての保証をしたものではなく、今後とも継続して評価を実施することが必要であると考える。

今回の暗号技術評価は、2000年6月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、同一名称の製品版の暗号技術やISO/IECなど他機関への提案暗号技術とは異なる部分がある場合がある。

また、今回公募した暗号技術は、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ので、評価対象暗号の技術仕様については、応募者のWebサイトから情報を得ることができるが、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。

更に、本報告書で評価対象となった暗号技術を実装する場合には、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、情報処理振興事業協会セキュリティセンター（問い合わせ先　FAX：03-5978-7510、e-mail：crypt-kobo@ipa.go.jp）までご連絡していただけると幸いです。

1. 暗号技術評価の概要

平成 15 年度（2003 年度）を目途としてその基盤を構築することとされている電子政府において、セキュリティの共通基盤の確保は重要な課題とされている。中でも、暗号技術は、電子化された情報の秘匿性及び非改ざん性の確保の他、電子認証を実現する技術であり、電子政府のセキュリティ確保のための重要な基盤技術である。

本報告書は、電子政府における適切な暗号技術利用をはかるために、我が国の電子政府システムに適用可能と想定される暗号技術について、技術的・専門的見地から、安全性、実装性等の特徴を評価したものであり､電子政府関連の調達や電子署名法･GPKI などの参考資料として活用されることを目的に作成した。

国際的に関連する動きとしては、ISO/IEC JTC1 による暗号アルゴリズムの標準化活動、米国 NIST による AES の選定活動、欧州の NESSIE プロジェクトがある。

なお、本事業は経済産業省の委託により実施するものであり、平成 12 年 4 月に経済産業省が策定した「情報セキュリティ政策実行プログラム-電子政府のセキュアな基盤構築に向けての通商産業省の貢献-」の重要な一部をなすものである。

今年度の暗号技術評価は、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（CRYPTREC)」を組織し、委員会の検討結果を踏まえて実施した。

評価対象暗号としては、2003 年度にその基盤を構築することとされている電子政府に適用できると想定される暗号技術を一般から公募するとともに、委員会で評価すべき暗号技術としてリストアップし、本報告書では詳細評価対象となった 35 件（応募暗号技術 27 件、その他 8 件）の評価結果を報告する。

暗号技術の公募を 2000 年 6 月 13 日から 2000 年 7 月 14 日の期間に、公開鍵暗号、共通鍵暗号、ハッシュ関数、擬似乱数生成のカテゴリーで実施した結果 48 件の暗号技術の応募があった。

暗号技術評価の外部評価委託や評価基準などの評価実施方法および評価結果については、「暗号技術評価委員会」の個別評価小委員会として、公開鍵暗号に関する評価方法・評価結果を検討する「公開鍵暗号評価小委員会」と共通鍵暗号、ハッシュ関数、擬似乱数生成に関する評価方法・評価結果を検討する「共通鍵暗号評価小委員会」を設置して検討し、「暗号技術評価委員会」に諮問した。

暗号評価は、大きく 2000 年 7 月から 9 月のスクリーニング評価と 2000 年 10 月から 2001 年 3 月の詳細評価とに分けて実施した。

表１　評価スケジュール

	平成１２年							平成１３年			
	6 月	7 月	8 月	9 月	10 月	11 月	12 月	1 月	2 月	3 月	4 月
公募	◀	――	▶								
ｽｸﾘｰﾆﾝｸﾞ評価			◀	――	――	▶					
詳細評価					◀	――	――	――	――	▶	
結果の公表											☆

評価対象となった暗号技術の評価は、国内外の専門家に評価して頂いた外部評価報告書、国内外学会論文、さらにソフトウェア実装評価結果を踏まえて、評価報告書にまとめた。

また、本暗号技術評価活動内容を国内外の暗号技術者や暗号利用者など広く理解していただくためのイベ

ントとして、2000 年 10 月 20 日に暗号技術シンポジウムを開催した。

さらに、本報告書の公開にあたって、2001 年 4 月に暗号技術評価報告会（仮称）の開催も予定している。

暗号技術評価委員会 委員（肩書き等は 2001 年 3 月末現在）

委員長	今井　秀樹	東京大学生産技術研究所教授
委員	岩下　直行	日本銀行金融研究所研究第２課調査役
委員	岡本　栄司	東邦大学理学部情報科学科教授
委員	岡本　龍明	日本電信電話株式会社情報流通プラットフォーム研究所主席研究員
委員	金子　敏信	東京理科大学理工学部電気工学科教授
委員	櫻井　幸一	九州大学大学院システム情報科学研究院情報工学部門助教授
委員	佐々木　良一	株式会社日立製作所システム開発研究所主管研究長
顧問	辻井　重男	中央大学理工学部情報工学科教授
特別委員	苗村　憲司	慶應義塾大学大学院政策・メディア研究科教授
委員	松井　充	三菱電機株式会社情報技術総合研究所情報セキュリティ技術部チームリーダー
委員	松本　勉	横浜国立大学工学部教授

（オブザーバ）

須永　和男	内閣官房情報セキュリティ対策推進室内閣参事官
武市　一幸	警察庁情報通信局技術対策課長
中島　明彦	防衛庁運用局指揮通信課長
高森　國臣	総務省行政管理局情報システム企画官
喜安　拓	総務省情報通信政策局通信規格課長
大森　慎吾	総務省通信総合研究所通信システム部長
後藤　博	法務省民事局商事課長
中田　悟	財務省大臣官房審議官室長
東井　芳隆	経済産業省商務情報政策局情報経済課情報セキュリティ政策室長
八田　勲	経済産業省産業技術環境局標準課情報電気標準化推進室長
大蒔　和仁	経済産業省産業技術総合研究所電子技術総合研究所情報アーキテクチャ部長

公開鍵暗号評価小委員会 委員(肩書き等は 2001 年 3 月末現在)

委員長	松本　勉	横浜国立大学工学部教授
委員	有田　正剛	日本電気株式会社情報通信メディア研究本部主任
委員	小暮　淳	株式会社富士通研究所コンピュータシステム研究所 セキュアコンピューティング研究部主任研究員
委員	酒井　康行	三菱電機株式会社情報技術総合研究所情報セキュリティ技術部
委員	静谷　啓樹	東北大学情報処理教育センター教授
委員	新保　淳	株式会社東芝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
委員	高橋　昌史	株式会社日立製作所システム開発研究所セキュリティシステム研究センタ
委員	趙　晋輝	中央大学理工学部電気電子情報通信工学科教授
委員	藤岡　淳	日本電信電話株式会社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主任研究員
委員	松崎　なつめ	松下電器産業株式会社マルチメディア開発センター
委員	宮地　充子	北陸先端科学技術大学院大学情報科学研究科助教授

共通鍵暗号評価小委員会 委員(肩書き等は 2001 年 3 月末現在)

委員長	金子　敏信	東京理科大学理工学部電気工学科教授
委員	荒木　純道	東京工業大学大学院理工学研究科電気電子工学専攻教授
委員	香田　徹	九州大学大学院システム情報科学院情報工学部門教授
委員	川村　信一	株式会社東芝研究開発センター コンピュータ・ネットワークラボラトリー主任研究員
委員	神田　雅透	日本電信電話株式会社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研究主任
委員	古原　和邦	東京大学生産技術研究所助手
委員	櫻井　幸一	九州大学大学院システム情報科学研究院情報工学部門助教授
委員	下山　武司	株式会社富士通研究所コンピュータシステム研究所 セキュアコンピューティング研究部
委員	宝木　和夫	株式会社日立製作所システム開発研究所セキュリティシステム研究センタ長
委員	館林　誠	松下電器産業株式会社マルチメディア開発センター主幹技師
委員	角尾　幸保	日本電気株式会社情報通信メディア研究本部主任研究員
委員	時田　俊雄	三菱電機株式会社情報技術総合研究所情報セキュリティ技術部
委員	森井　昌克	徳島大学工学部知能情報工学科教授

公開鍵暗号評価小委員会及び共通鍵暗号評価小委員会オブザーバ(肩書き等は 2001 年 3 月末現在)

桝賀　政浩	防衛庁運用局指揮通信課
松井　教安	防衛庁運用局指揮通信課
今住　秀孝	総務省行政管理局行政情報システム企画課
丹代　武	総務省情報通信政策局通信規格課
今井　清春	総務省情報通信政策局通信規格課
山村　明弘	総務省通信総合研究所通信システム部非常通信研究室
桑原　敦	通商産業省機械情報産業局情報処理振興課情報セキュリティ政策室（当時）
山本　文土	経済産業省商務情報政策局情報経済課情報セキュリティ政策室
田辺　雄史	経済産業省商務情報政策局情報経済課情報セキュリティ政策室
石井　伸治	経済産業省商務情報政策局情報経済課情報セキュリティ政策室
平野　芳行	経済産業省産業技術環境局標準課情報電気標準化推進室
渡辺　創	経済産業省産業技術総合研究所電子技術総合研究所
青木　和麻呂	通信・放送機構研究企画管理部研究企画課

（事務局）

情報処理振興事業協会セキュリティセンター暗号技術調査室

2. 公募対象とした暗号技術

2.1 公募対象とした暗号技術の分類

暗号技術は、長い歴史を持っているが、いわゆる電子政府で使用される暗号としては、1970 年代後半に研究が活発化した「現代暗号」技術である。「現代暗号」技術の最大の特徴は、「暗号化・復号」に関する手続きを広く公開している点にある。

現代暗号技術は、1976 年に米国の政府標準として公表された DES(Data Encryption Standard)に代表される共通鍵暗号と、Rivest-Shamir-Adleman により提案された RSA 暗号に代表される公開鍵暗号に大別される。これらの暗号が提供するセキュリティ機能は、守秘、署名、認証の 3 つである。

守秘の目的では、主に、共通鍵暗号が使われ、署名、認証には、公開鍵暗号が使われることが主流になりつつある。この 2 種類の暗号は、通常、両暗号の特徴を補完しあう形でシステムでは使われる。例えば、共通鍵暗号における秘密鍵（共通鍵）を送信者、受信者で共有するために、公開鍵暗号の鍵共有の機能が使われ、公開鍵暗号を使って電子署名を作る際には、平文を、暗号学的なハッシュ関数で処理の後、署名が作成される。公開鍵暗号の鍵対（公開鍵：パブリック鍵と、秘密鍵：プライベート鍵）、秘密鍵（共通鍵）、又は、その種を生成するために、（擬似）乱数生成法が必要である。また、この擬似乱数生成法は多くの公開鍵暗号において、守秘、署名の機能を実現する際の乱数源としても必要とされる。

共通鍵暗号は、そのデータ処理の構造により、ブロック単位にデータを処理するブロック暗号とデータを系列と考えて処理するストリーム暗号に分かれる。

図 2.1.1 に今年度公募対象とした暗号技術の分類を示す。

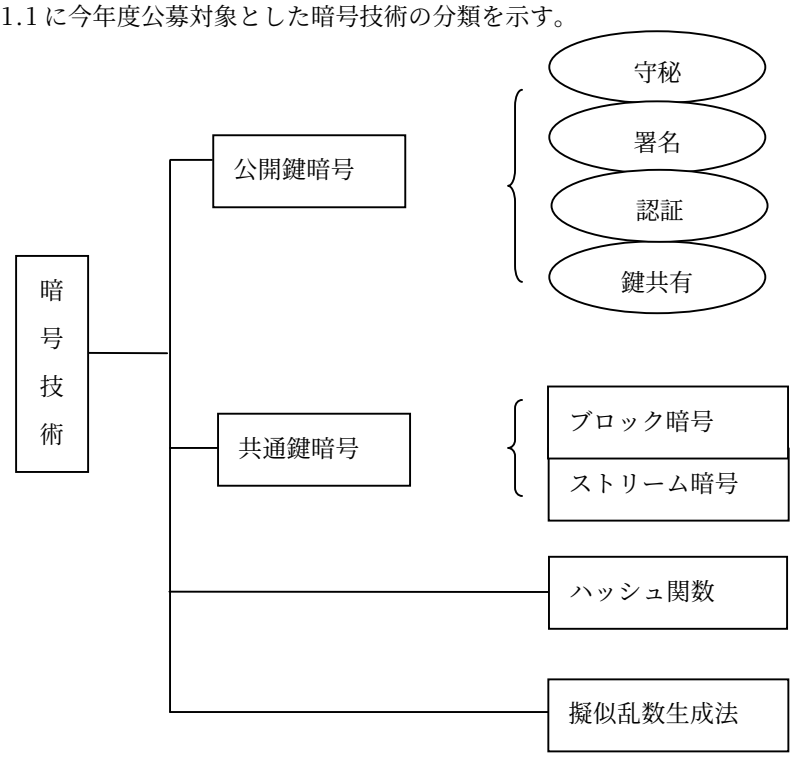


図 2.1.1　公募対象暗号技術の分類

2.2　公開鍵暗号技術について

公開鍵暗号は、安全性の根拠としては素因数分解問題、離散対数問題、楕円曲線上の離散対数問題に、機能面では署名機能、守秘機能、鍵共有機能、認証機能のいずれかに大きく分類できる。
今回の暗号評価では、機能面での分類で応募された暗号技術（スキームとプリミティブ・補助関数の組み合わせ）の評価を実施した。

ここで、暗号方式（暗号スキーム）とは、基本暗号（暗号プリミティブ）と補助関数（暗号補助関数）とを用いて機能を発揮させるアルゴリズムを指し、基本暗号の要件と、補助関数の要件と、アルゴリズムの記述とからなる。

基本暗号とは、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、その他の安全性根拠に基づく安全性を有する要素暗号アルゴリズムで、補助関数とは、ハッシュ関数や乱数（擬似乱数）等、暗号方式が機能を発揮する上で、基本暗号の外に必要となる要素を指す。

2.2.1　守秘機能を実現する公開鍵暗号

送信者と受信者の間で、秘密裏に任意の情報を共有する機能が、守秘機能である。大容量データの守秘には共通鍵暗号方式を用いて守秘機能を実現するのが一般的であるが、短い情報などを暗号通信する場合には、公開鍵暗号を利用することがある。この手順を以下に示す。

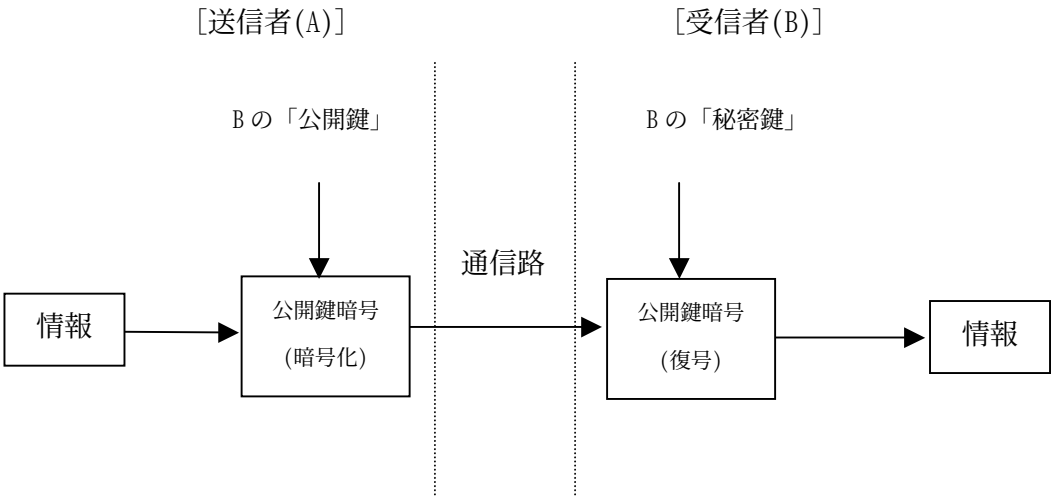


図 2.2.1　公開鍵暗号技術を用いた守秘機能の例

ステップ１：受信者(B)は、鍵対（公開鍵と秘密鍵）を生成し、公開鍵を公開する。
ステップ２：送信者（A）はBの公開鍵を入手し、送信情報（平文）に対し、Bの公開鍵を用いて暗号化し送付する。
ステップ３：受信者(B)は受信した暗号文を自分の秘密鍵を用いて復号処理を行い、情報（平文）を入手する。

2.2.2　署名機能を実現する公開鍵暗号

署名機能とは、電子情報の正当性を確認する機能である。電子情報に対する正当性とは、署名作成者の確認機能と電子情報自体の改ざんの有無の確認機能の両方を意味している。以下にその概略の手順を示す。

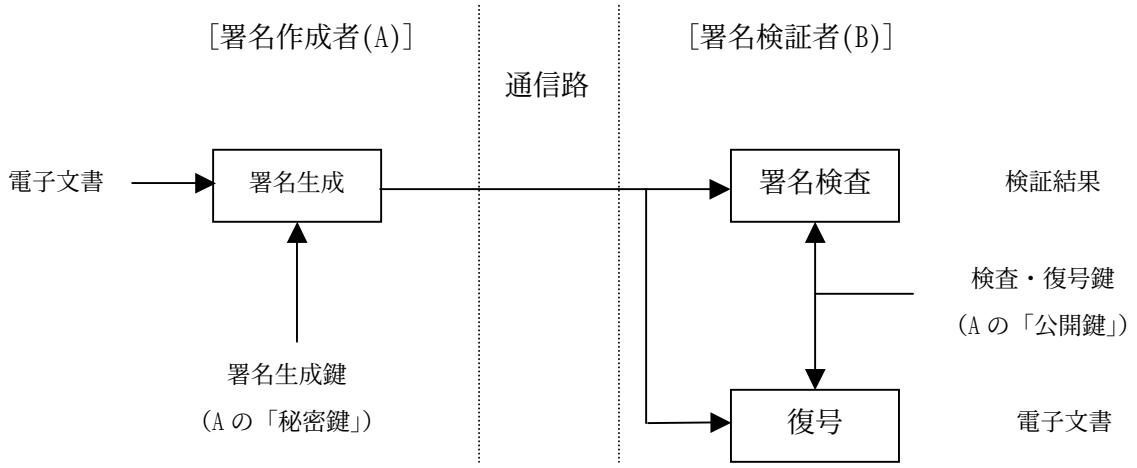


図 2.2.2　公開鍵暗号技術を用いた署名機能の例

ステップ１： A は、鍵対（公開鍵と秘密鍵）を生成し、公開鍵を公開する。

ステップ２： A は秘密鍵を用いて、電子文書に対して署名付き電子文書を作成する。

ステップ３： A は B に署名付き電子文書を送付する。

ステップ４： B は、A の公開鍵を用いて署名付き電子文書を処理し、電子文書を復号するとともに、検証結果を得る。

2.2.3　認証機能を実現する公開鍵暗号

認証機能とは、被認証者の正当性を検証者が確認する機能である。この機能は、共通鍵暗号を用いても実現可能であるが、鍵管理の煩雑性を回避するために公開鍵暗号も用いることがある。公開鍵暗号を用いた（相手）認証とは、署名機能の公開鍵暗号技術か守秘機能の公開鍵暗号技術を利用し、被認証者が秘密に保持する秘密鍵（プライベート鍵）とそれに対応した検証用の公開鍵を用いて、被認証者と検証者の間で、前もって定められた手順（認証プロトコル）に従って、情報を送受することで、公開鍵に対応した秘密鍵を持っていることを証明する技術である。公開鍵暗号技術を用いた認証機能の例を図 2.2.3 に示す。

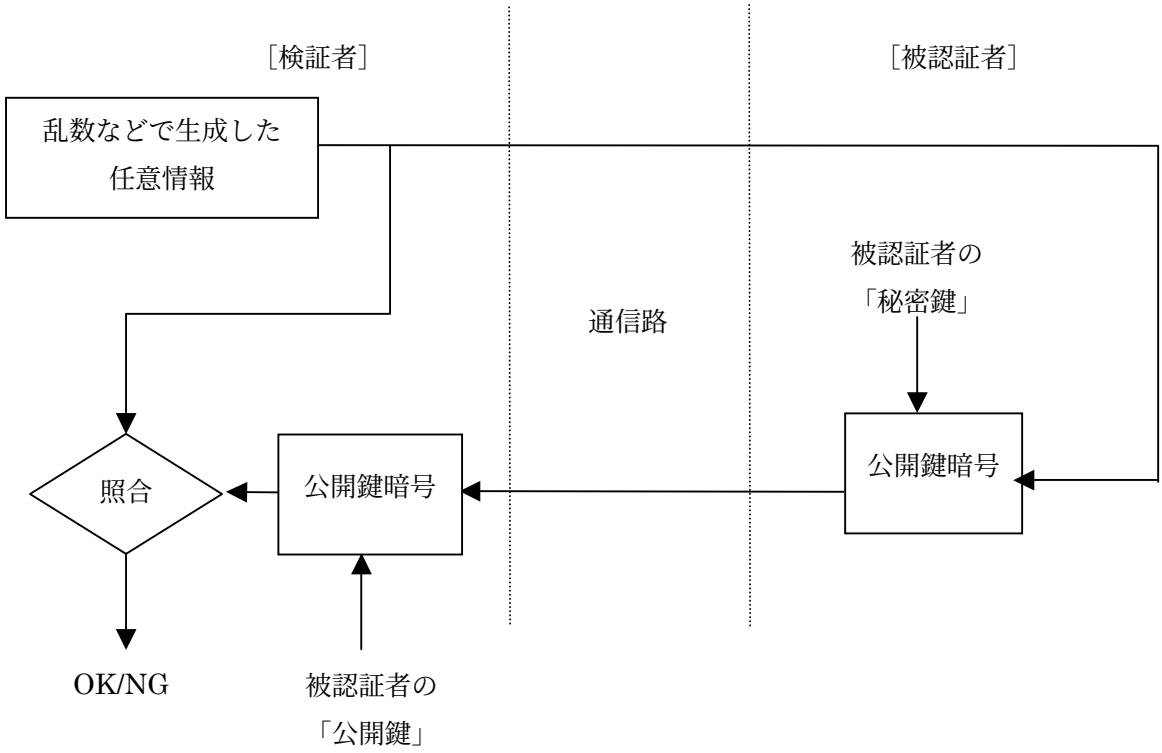


図 2.2.3　公開鍵暗号技術を用いた認証機能の例（2move 型）

ステップ１：被認証者は、鍵対（公開鍵と秘密鍵の対）を生成し、公開鍵を公開する。
ステップ２：検証者より、乱数生成等により発生させた任意の情報を、被認証者に送付する。
ステップ３：被認証者は、被認証者の秘密鍵を用いて、送られてきた情報の暗号文（署名）を生成する。
ステップ４：暗号文を検証者に送付する。
ステップ５：検証者は、被認証者の公開鍵を用い、送られてきた暗号文を復号し、平文を得る。
ステップ６：受信した平文と生成した情報を照合し、一致すれば、被認証者が“正当”であると判断し、
　　一致しなければ、“正当でない”と判断する。

2.2.4　鍵共有機能を実現する公開鍵暗号

鍵共有（鍵配送）機能とは、共通鍵暗号技術を利用する際に送信者と受信者の間で「鍵」情報を共有（配送）する機能である。特に、大規模なシステム（ユーザー数が多いシステム）では公開鍵暗号技術を用いることが多い。公開鍵暗号技術で鍵共有を行う場合においては、前述の守秘機能を持った公開鍵暗号技術で実現することもできる。以下には、Diffie と Hellman により提案された DH 鍵共有型の手順を示す。

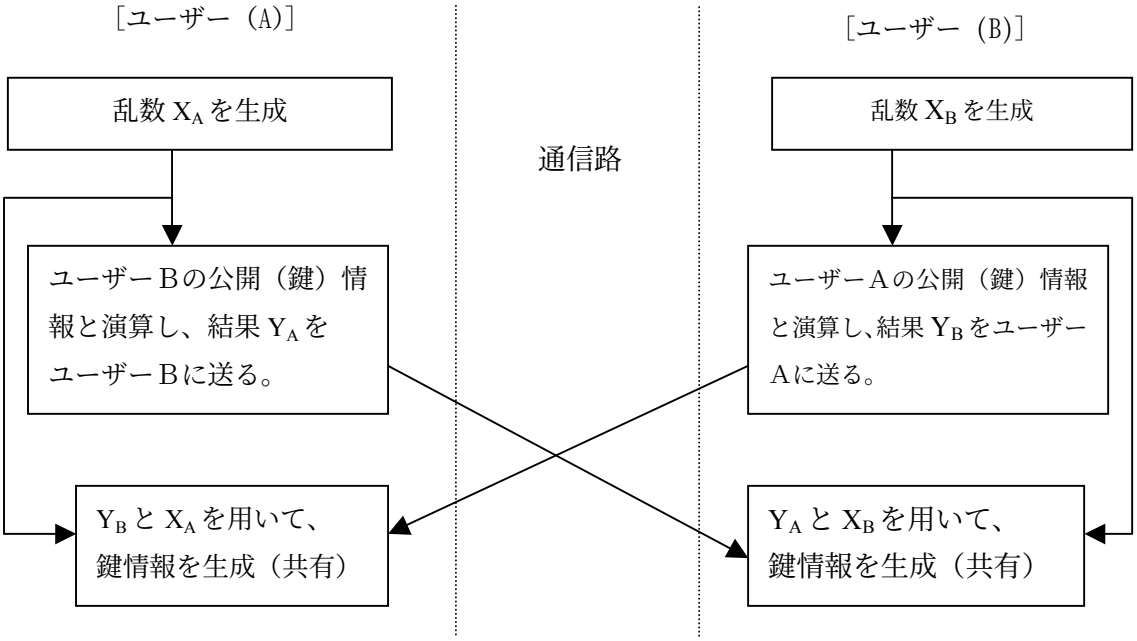


図 2.2.4　公開鍵暗号技術を用いた鍵共有機能の例（DH 型）

ステップ１：システム全体に共通な公開情報を生成し、各ユーザーに配布する。

（DH 鍵共有方式で有れば、素数 p と原始根 α を各ユーザーに公開する。）

ステップ２：ユーザーＡは、乱数 X_A を生成する。（X_A がユーザーA の秘密鍵となる。）

ステップ３：乱数 X_A とシステムに共通な公開情報から、ユーザーA の公開鍵 Y_A を生成する。

（DH 鍵共有方式で有れば、$Y_A = \alpha^{X_A} \bmod p$ をユーザーA の公開鍵として公開する。）

ステップ４：ユーザーＢは、乱数 X_B を生成する。（X_B がユーザーB の秘密鍵となる。）

ステップ５：乱数 X_B とシステムに共通な公開情報から、ユーザーＡの公開鍵 Y_B を生成する。

（DH 鍵共有方式で有れば、$Y_B = \alpha^{X_B} \bmod p$ をユーザーB の公開鍵として公開する。）

ステップ６：ユーザーＡ、ユーザーＢは相互に公開鍵 Y_A と Y_B を相互に交換する。

ステップ７：秘密鍵と受信した公開鍵から共有鍵 K を生成する。

（DH 鍵共有方式で有れば、ユーザーＡ、ユーザーＢはそれぞれ $K = (Y_B)^{X_A} \bmod p = \alpha^{X_A X_B} \bmod p$ 及び $K = (Y_A)^{X_B} \bmod p = \alpha^{X_B X_A} \bmod p$ を計算し、その結果 K を共有鍵とする。）

2.3 共通鍵暗号技術

2.3.1 ブロック暗号

データをブロックに分割し、ブロック単位で処理を行うものがブロック暗号である。分割するブロックのサイズは、64 ビットと 128 ビットとされるブロック暗号が一般的である。分割するブロックサイズが 64 ビットのものを 64 ビットブロック暗号、分割するブロックサイズが 128 ビットのものを 128 ビットブロック暗号という。ブロック暗号はデータランダム化部と鍵スケジュール部で構成される(図 2.3.1)。

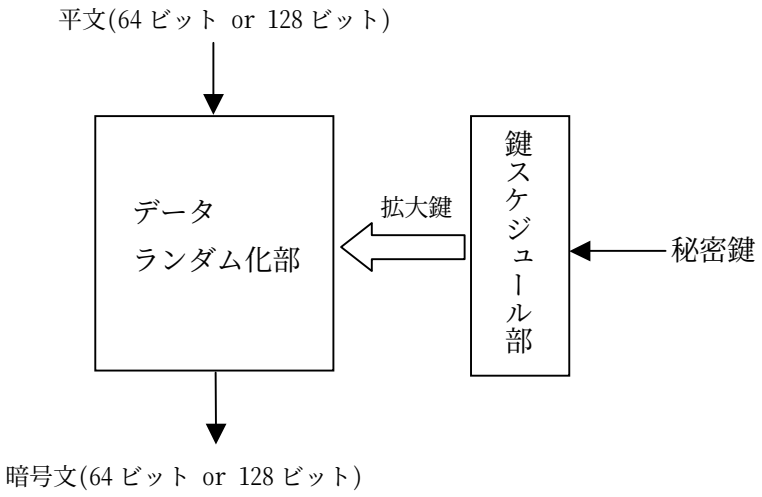
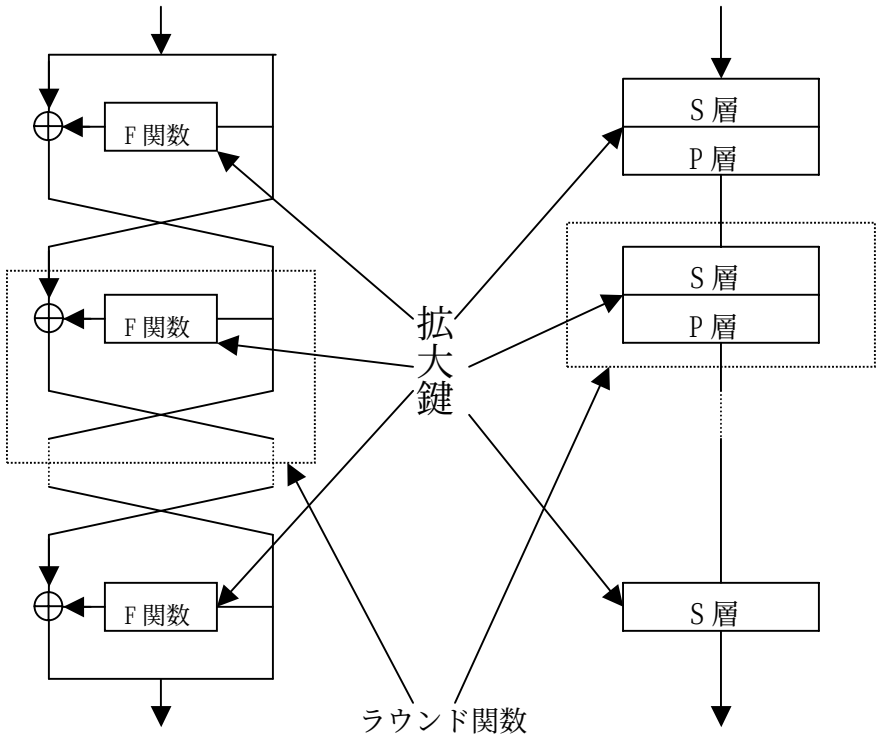


図 2.3.1 ブロック暗号の概略図

今回は、DES-ChallengeⅢ等に見られるコンピュータの処理能力の現状と、進化のスピードを考慮し、秘密鍵が 128 ビット以上のものを公募対象とした。鍵スケジュール部は、秘密鍵からデータランダム化部で用いる拡大鍵を生成する。データランダム化部は、拡大鍵が入力され、それに基づいて平文を暗号文に変換する。復号は暗号化に用いた鍵と同一のものを使用し、暗号文を平文に変換する。

データランダム化部の代表的な構造として Feistel 構造と SPN 構造がある(図 2.3.2)。



(i) Feistel 構造　　(ii) SPN 構造

図 2.3.2　データランダム化部の代表的な構造

Feistel 構造は入力を 2 分割し、片方が F 関数に入力され、他方に影響を与える構造を繰り返す。この F 関数は、S-box 等の非線形関数などを用いて構成される。SPN 構造は入力を分割せず、S 層(非線形層)と P 層(線形層)で構成されるラウンド関数を繰り返す。この F 関数やラウンド関数の繰り返し回数を段数と呼ぶ。また、Feistel 構造は様々に拡張され、図 2.3.3(i)や図 2.3.3(ii)の様なものもある。ここでは、これらを変形 Feistel 構造と呼ぶ。ブロック暗号にはいくつかの利用モード(mode of operation)が存在する。図 2.3.1 は ECB モードと呼ばれている。

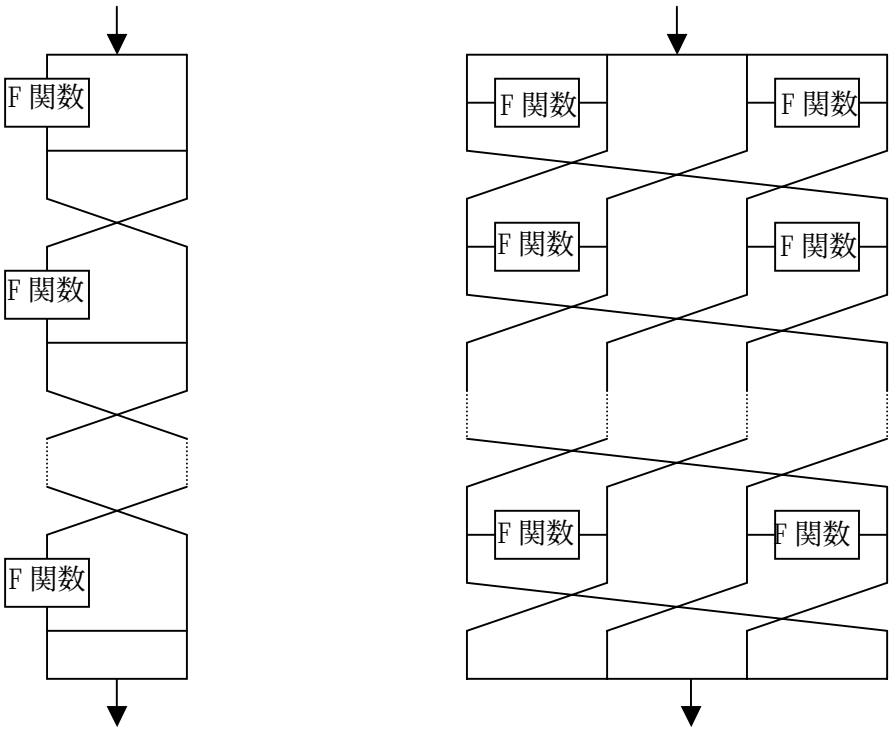


図 2.3.3　変形 Feistel 構造の例

2.3.2　ストリーム暗号

ストリーム暗号はデータをブロック暗号のように区切るのではなく、系列のまま処理する。暗号文系列は、平文系列に鍵系列を排他的論理和して得られる(図 2.3.4)。

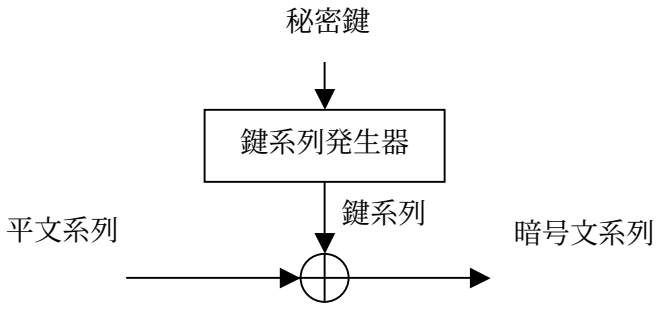


図 2.3.4　ストリーム暗号の構造例

鍵系列発生器は擬似乱数生成器であり、秘密鍵による決定的アルゴリズムに基づいて鍵系列である乱数列を生成する。今回の公募では、ブロック暗号と同様の考慮により、秘密鍵が 128 ビット以上のものを公募対象とした。ストリーム暗号の構造は多種多様であり、ブロック暗号のような代表的構造は存在しない。復号は、同一の秘密鍵、鍵系列発生器を用い、暗号文系列に鍵系列を排他的論理和して平文系列を得る。

2.4 ハッシュ関数

公募対象としたハッシュ関数は暗号用途であり、パスワード認証、電子署名、メッセージ認証等の使用を想定した。従って暗号用途のハッシュ関数に求められる性能は、一方向性と無衝突性である。一方向性とは、出力から入力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つの入力に対し同じハッシュ値を出力することである。今回の公募では、ブロック暗号と同様の考慮により、ハッシュ値が 128 ビット以上のものを公募対象とした。

2.5 擬似乱数生成

擬似乱数生成は、ストリーム暗号用途のものではなく、暗号の鍵または鍵の種等の用途を想定している。性質は真性乱数に近いが、例えばユーザーのキーボード入力間隔を利用した方法などでは、このような用途に対して一様性に乏しく十分大きい乱数値を期待できない。公募対象の擬似乱数生成は、上述の目的を満たす程度に大きな乱数値を出力し、特定の値が出力される確率が充分小さく一様に分布していることが求められる。さらに暗号用途であるから、次の乱数値が計算量的に予測不可能である必要がある。尚、特殊な機器を必要とする物理的乱数は公募対象外とした。

3. 評価方法の概要

3.1 評価の目的

暗号技術に関する評価は、安全性評価と実装評価を実施した。実装評価にはソフトウェア(SW)実装評価とハードウェア(HW)実装評価がある。

暗号の安全性評価は、安全性証明の検証や仮定の妥当性など設計段階の評価と設計された後に行われる各種の攻撃に対する評価結果により判断される。

設計段階、あるいは攻撃評価段階での安全性評価に関しては、従来知られている統計量的な評価だけでは十分ではなく、暗号解析技術からの評価が必須である。この暗号解析技術は暗号設計技術と表裏をなす関係にある。言い換えれば、暗号強度評価とは、解読に対する条件（利用環境や攻撃者の計算能力等）を仮定し、その仮定の下で適当な暗号解読技術を適用して暗号解読を行う際に、必要な情報量（収集する必要のある平文ないし暗号文の量）と集めた情報から使用された暗号鍵を計算する為の計算量を示すことであり、この情報量や計算量を用いて、暗号強度を評価することが一般的である。この際、適用される暗号解析技術は唯一ではあり得ず、複数の暗号解析技術により評価することが必須である。暗号技術の使用される環境や条件により、評価尺度は様々に変化する。従って、安全性評価に際しては、複数の暗号解読技術に基づいた安全性の評価を行うべきである。評価された暗号技術を使用する際には、安全性評価の結果、適用されるシステムの情報セキュリティポリシー、実際に使用する環境や条件を勘案した「総合的な判断」が必要である。

図 3.1 に暗号解析技術、暗号設計技術と安全な暗号の関係を模式的に示す。

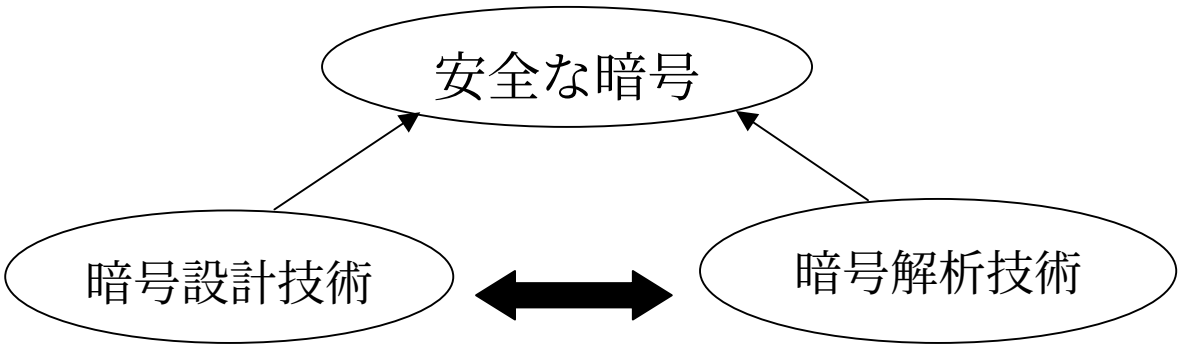


図 3.1　安全な暗号と設計／解析技術

また、安全性の評価としては、本報告書の結果だけではなく、システムとしての安全性の評価も重要である。システム的な評価は、ISO/IEC 15408 共通評価基準（CC: Common Criteria）等により評価が行われる。さらにシステムで実装された状態での暗号技術の評価も必要である。

一方、実装性の評価については、適用されるシステムのセキュリティポリシーに基づき、実装性能を考慮すべきである。

3.2　スクリーニング評価と詳細評価

評価作業は、スクリーニング評価と詳細評価の２段階に分けて実施した。

(i)　スクリーニング評価

スクリーニング評価の目的は、提出された応募書類に基づいて詳細評価を行うに値するかを判断した。スクリーニング評価項目は、以下の通りである。

- 詳細に評価するための情報（記述の有無、記述内容の論理的整合性/自己完結性）が整っていることの確認した。
- 書面上で容易に判明するような欠点（解読手法など）の検査を行った。
- 応募時点で提出された暗号技術仕様書、自己評価の内容の点検と正当性（内容の妥当性）の確認した。

(ii) 詳細評価

詳細評価は、安全性評価及び実装性評価（ソフトウェアによる実装、及びハードウェアによる実装）を行った。

安全性評価においては、国内外の暗号研究者に委託して既存の解析技術（攻撃法）による統一的な評価と応募された暗号技術に固有な欠陥（解読法）の存在の点検という２通りの評価を行った。

ソフトウェア実装評価では、統一したプラットフォーム上で応募者が実装したコードを用いて、速度を評価した。

ハードウェア評価では、FPGA[1]ないし GA[2]での実現を想定して、国内の暗号実装研究者に委託して、ゲート数や処理速度の大まかな見積もりを行った。

3.3　公開鍵暗号の評価方法

3.3.1　安全性評価

応募された守秘、認証、署名、鍵共有いずれかの機能を実現するための手続き（暗号プロトコル）を含んだ広義の暗号方式（以下、暗号スキーム[3]と略記する）に関して、その暗号スキームで要求された要件を満たす基本となる暗号アルゴリズム（以下、暗号プリミティブ[4]と略記する）、補助関数の存在を仮定して、暗号スキームとしての安全性を評価した。、応募された実現例で使用した暗号プリミティブや補助関数を、提案された暗号プリミティブや補助関数の満たすべき要件との適合性、妥当性の観点から評価した。また、提案されたパラメータと暗号プリミティブは、最適と思われる攻撃法により評価した。

(a)暗号スキームに関する安全性評価項目

応募された暗号スキームに対して、最適と思われる攻撃方法を選択し、暗号スキームとしての安全性の目標に応じた暗号スキームの振舞いを評価した。攻撃方法と攻撃の目標の組み合わせ毎に、応募方式が、安全性の証明を有するか、あるいは、ヒューリスティックな根拠を有するか等の観点で評価した。

---

[1] FPGA：Field Programable Gate Array の略

[2] GA：Gate Array の略。今回のプラットフォームとしては、0.3μm 前後の GA を想定した。

[3] 暗号スキーム：署名や鍵共有等の機能を実現するための手続き（暗号プロトコル）を含んだ広義の暗号方式

[4] 暗号プリミティブ：情報セキュリティ機能を提供する基本的な暗号アルゴリズム。

攻撃の方法：受動的攻撃、能動的攻撃、その他の攻撃

攻撃の目標：実現すべき機能（守秘、認証、署名、鍵共有）への影響

(b)暗号プリミティブに関する安全性評価項目

各暗号プリミティブに対し，それぞれの安全性根拠となる素因数分解問題，離散対数問題，楕円曲線離散対数問題に基づいた既知の攻撃法に対する安全性について評価を行った。

（ｉ）整数の素因数分解問題に基づく暗号プリミティブの評価項目

既知の攻撃法に対する計算量的耐性、および、暗号プリミティブに固有なその他の方法

（ii）有限体上の離散対数問題に基づく暗号プリミティブの評価項目

既知の攻撃法に対する計算量的耐性、および、暗号プリミティブに固有なその他の方法

（iii）楕円曲線上の離散対数問題に基づく暗号プリミティブの評価項目

既知の攻撃法に対する計算量的耐性、および、暗号プリミティブに固有なその他の方法。

（iv）その他の安全性根拠に基づく暗号プリミティブの評価項目

安全性根拠に対する既知の攻撃法、暗号プリミティブに固有の攻撃法等

3.3.2 ソフトウェア実装評価

今回の評価では、実装評価対象コードは応募者側で作成し、測定用メインプログラムの正当性を確認した。また、安全性に関しては、RSA（素因数分解問題に基づく暗号） 1,024 ビット相当以上の強度とした。ソフトウェア実装評価項目は以下の通りである。

- 提出された暗号技術仕様書の記述内容の評価（第三者による実装が可能となる正確で十分な技術情報が提供されているかどうかの点検)。
- 指定したプラットフォームで実行可能性の評価（特殊なハードウェアや巨大な量の記憶装置が仮定されていないこと)。
- 指定したプラットフォームでの、処理速度、メモリ等のリソース量の評価。

指定したプラットフォーム

CPU	：PentiumIII (650MHz)
OS	：Windows98 SE
搭載メモリ	：64MB
コンパイラ	：Visual C++ Ver.6.0 SP3

3.4 共通鍵暗号の評価方法

3.4.1 安全性評価

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る。これを満たすには鍵の量が平文の量以上である必要があるが、これは運用上、現実的ではない。共通鍵暗号の安全性は情報量的安全性でなく、計算量的安全性で示される。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方法は現在のところ存在しない。そこで実際に攻撃を行い、それが必要とするコスト（計算量、データ量、メモリ量）で安全性を評価する。実際に攻撃を行うことから、暗号のアルゴリズムが公開され既知であることが前提である。攻撃を行う側の使用可能な情報の条件で、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

暗号文のみが利用できる場合

・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

攻撃者は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が利用可能

使用できる情報により攻撃には難易が存在し、後者ほど攻撃側に有利となる。暗号文単独攻撃で攻撃可能な暗号は十分な安全性を有するとは判断し難い。

十分な量の暗号文に対応する平文が既知である、という条件を持つこれら攻撃では、攻撃側が無限の計算量を持つとすれば、全数探索で必ず秘密鍵を発見できる。学術的には、ある攻撃方法が必要とするコストが、この全数探索よりも少なくて済む時、攻撃が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、その暗号は安全ではないと判断する。DES-ChallengeⅢ(1999 年 1 月に行われた DES 解読コンテスト。ユーザー鍵(56 ビット)の解読に 22 時間要した)等に見られる現状や、処理向上の進歩の速さを考慮すると 2^{64} 程度のコストは処理可能と見積もられる。従って 2^k $(>2^{64})$ よりも少ないコストで攻撃可能であっても実用上問題ない場合もあるが、上記の状況を考えると長期の使用は薦められない。

攻撃方法には、そのカテゴリーに含まれるものに適用可能な汎用の攻撃方法と、その暗号に特化した攻撃方法がある。同一カテゴリーに含まれる応募暗号に対しては、そのカテゴリーに対する汎用攻撃を同一評価者が適用し評価する「横並び評価」と、各暗号特有の弱点の発見を目的とする「個別評価」の 2 種類の評価方法を実行し、それらの結果を総合的にまとめた。

共通鍵暗号には 64 ビットブロック暗号、128 ビットブロック暗号、ストリーム暗号がある。5.1.1 節に 64 ビットブロック暗号、5.1.2 に 128 ビットブロック暗号に関する安全性評価について、5.1.3 にストリーム暗号に関する安全性評価について記す。

(1)ブロック暗号

ブロック暗号に対しては、今回の評価では、汎用の攻撃方法として以下に対する強度を評価した。

・差分解読法

・線形解読法

・高階差分解読法

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行した。

差分解読法/線形解読法

差分解読法は Biham と Shamir によって、1990 年に提案された。DES に対して公開された攻撃方法であるが、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、適用可能な選択平文攻撃の一つである。

線形解読法は 1993 年に三菱電機の松井によって提案された。差分解読法と同様に DES に対する攻撃方法として提案されたが、ブロック暗号全体に適用可能な汎用的攻撃方法である。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、適用可能な既知平文攻撃の一つである。

これらに対する耐性は、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

1.構成部品ごとに評価を行い確率の上界を求める方法

2.計算機探索より求める方法

などによって見積もられる。

差分解読法・線形解読法に対する証明可能安全性

応募暗号によっては差分解読法・線形解読法に対し、これら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 構造のブロック暗号に対して、F 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、差分解読法・線形解読法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、差分解読法と線形解読法に対してのみ有効な議論であることを留意されたい。

高階差分解読法

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 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、適用可能な選択平文攻撃の一つである。KN 暗号は、上記の差分/線形解読法に対する証明可能安全性を有するが、F 関数の代数次数が小さいことから高階差分解読法で攻撃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、入力ビットを全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。この評価方法は暗号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。

全ての共通鍵型暗号は

・ラウンド関数
・データランダム化部
・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も

（1） ラウンド関数単体
（2） データランダム化部
（3） 鍵スケジュール部単体

について検討評価を行い、それを踏まえて

（4） 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

(1) 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⊿X、鍵差分⊿Kのハミング重みは m =1、2について実施。
(2) AVD（差分の拡散値） ： 出力差分値のハミング重みの平均値。
(3) 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。
(4) 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。

の項目が調査対象である。

また共通鍵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る。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

(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-140-1/2 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 等汎用な攻撃方法に対する耐性を、ブロック暗号における個別評価と同様に評価した。

3.4.2　ソフトウェア実装評価

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、全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全ての暗号に対して行った。残りの環境は各暗号の設計思想を尊重し応募者の選択とした。また、実際には8ビットCPU等のロースペック環境における評価も行うべきではあるが、評価期間が限られていることから、今回は見送った。なお、ハッシュ関数と擬似乱数生成に対してはソフトウェア評価を実施しなかった。

暗号プログラムは応募者が実装したものであり、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。尚、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等にかなり影響される。従って、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以下に示す。

1.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

2.サーバ環境

CPU	：Ultra SPARC IIi (400MHz)
OS	：Solaris 7
搭載メモリ	：256MB
コンパイラ	：Forte C 6

3.ハイエンド環境

CPU	：Alpha21264 (463MHz)
OS	：Tru64 UNIX V5.1
搭載メモリ	：512MB
コンパイラ	：DEC C

ブロック暗号に関しては、

1.データランダム化部

2.鍵スケジュール部＋データランダム化部

の 2 種類の測定を実施した。データランダム化部の測定は、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 のサイクル数(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である。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。従って、鍵のセットアップ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の測定では、1 ブロックの暗号化(復号)毎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の測定と同じである。ただし、上述したデータランダム化部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速度とはならない場合もある。

ストリーム暗号の測定においては、一般的には鍵のセットアップが無いため、データランダム化部の測定のみ行った。ただし、64 ビットブロック暗号と同一の測定プログラムを用いたため、ストリーム暗号の実装性を犠牲にしている場合がある。使用目的からすれば、ストリーム暗号はブロック暗号と比較してハードウェア指向が強いので、ソフトウェア評価よりもハードウェア評価を主眼にすべきである。従って、ストリーム暗号に対するソフトウェア評価は、ソフトウェア実装したとしても最低条件の使用に耐えうる性能を実現しているかどうかの確認を目的とした。

3.4.3　ハードウェア実装評価

今回のハードウェア実装評価では、「利用可能な状態」を確認する目的で、共通鍵暗号のハードウェア実装評価を行った。使用するプロセス（FPGA、GA）別に、処理速度評価、リソース使用数量（FPGA の場合には、使用セル数、GA 等の場合に、使用ゲート数等）を評価する。

なお、今年度のハードウェア評価に関しては、応募書類にハードウェア実装情報（結果）が記載されている方式を対象とし、シミュレーション評価結果をもって、処理速度、リソース消費量を評価し、応募書類に記載された情報の妥当性を検証するにとどめた。

なお、「利用可能な状態」とは、応募された暗号技術が単なる理論だけではなく、実際に実装可能で、応募のカテゴリーに対応した機能を実現出来ている状態にあることを言う。

［ブロック暗号の評価の手法］

評価のための対象ディバイスとしては、0.25～0.35μm の ASIC ライブラリであり、設計記述言語は Verilog-HDL、回路合成には Design Compiler を使用している。また、今回のハードウェア実装評価にあたっては、その他評価が必要な暗号技術となっている Triple DES(3-Key)を相対評価の指標として評価することとした。但し、実行速度、ゲート規模等に関しては、実装アーキテクチャの違いや最適化の状況が異なるため、あくまでも“目安”でしかあり得ない。また、実装者の経験に左右されるところが大きいが、実行速度、ゲート規模としては、実際の実装時点では、概ね性能の向上（高速化、小型化）が期待できる。

［ストリーム暗号の評価の手法］

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、ハードウェアで実現することが多いので、妥当なな回路規模であれば処理速度優先の設計条件を優先した評価とした。

ハードウェア実装評価のために使用した開発環境は、下記の通り。

・ModelSim VHDL/Verilog Version 5.4e（Model Technology）

・Synplify（Synplicity Inc.）

3.5　ハッシュ関数の評価方法

ハッシュ関数は、n ビット入力 m ビット出力（$n>m$）の関数である。ハッシュ関数に求められる性能は、「一方向性」と「無衝突性」である。一方向性とは、出力から入力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よりも入力の方が大きいため、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。

3.6　擬似乱数生成の評価方法

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。安全性評価として、出力される乱数には、前述のストリーム暗号用擬似乱数生成器に対する評価である FIPS140-1/2 等の評価を行った。

・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

また、さらに暗号用途であることから出力が予測不可能であり、入力空間が出力空間に比べ十分大きい必要がある。

4. 公開鍵暗号の評価

4.1 総評

4.1.1 詳細評価対象公開鍵暗号技術一覧

詳細評価を行った公開鍵暗号技術は機能と安全性の根拠とから以下のように分類できる。

機能 安全性の根拠	署名	守秘	鍵共有
素因数分解問題 IF	ACE Sign ESIGN-signature　註 1) RSA-PSS	EPOC-1　　註 3) EPOC-2　　註 3) EPOC-3　　註 3) HIME-1　　註 4) HIME-2 RSA-OAEP	
離散対数問題 DL	DSA	ACE Encrypt	DH
楕円曲線 離散対数問題 ECDL	ECDSA in SEC1 MY-ELLTY ECMR-160/192/OEF-h 註 2)	ECAES in SEC1 PSEC-1　　註 3) PSEC-2　　註 3) PSEC-3　　註 3)	ECDHS in SEC1 ECMQVS in SEC1 HDEF-ECDH

註

1) ESIGN-identification というスキームが「認証」機能の項目で応募されたが、認証プロトコルとしての記述がなく、「署名」機能の項目で応募された ESIGN-signature と名称以外は同一であるので、ESIGN-signature に合併した。

2) MY-ELLTY ECMR-160/192/OEF-h は、MY-ELLTY ECMR-160-h, MY-ELLTY ECMR-192-h, MY-ELLTY ECMR-OEF-h という異なる３件の応募であったが、使用する体が異なるだけで、署名スキームとしては同一のもとの見なせるため、３件をまとめて評価することとした。

3) EPOC-1, EPOC-2, EPOC-3 は EPOC という名称の暗号の組として、また PSEC-1, PSEC-2, PSEC-3 は PSEC という名称の暗号の組として、それぞれ応募されたが、個々に異なる方式であるので、このように分類した。

4) HIME-1 は「鍵共有」機能の項目で応募されたが、暗号の形式としては共有すべき鍵を一方のエンティティが公開鍵方式で暗号化して他方のエンティティに送るというものであり、このような使い方は「守秘」機能の項目に分類される全ての暗号に適用できるものであるから、HIME-1 は「守秘」機能の項目に配置した。すなわち,「鍵共有」機能の項目には共有される鍵の生成に両エンティティが関与するものだけを配置することにした。

なお、本報告書における、個々の公開鍵暗号の記載順は、上表の署名－素因数分解、署名－離散対数、署名－楕円曲線離散対数、守秘－素因数分解、守秘－離散対数、守秘－楕円曲線離散対数、鍵共有－離散対数、鍵共有－楕円曲線離散対数の順であり、各欄の中は英語のアルファベット順に配置している。

4.1.2 公開鍵暗号技術の総評

本プロジェクトで詳細評価を行った公開鍵暗号技術は、実装性についてはいずれの暗号も概ね許容できる処理性能を有していることが、ソフトウェア実装評価の結果から確認できる。本節では、安全性に重点をおいて、署名、守秘、鍵共有の機能別に、各々の暗号技術に対する本プロジェクトにおける評価をまとめている。安全性に関して暗号技術の提案者の主張とは異なる指摘がなされ、その妥当性について本プロジェクトで結論を出すに至らなかった事項については、その旨を明記している。これらの事項については、本プロジェクトの後継プロジェクト等で評価を続けることが望ましい。

さて、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、直ちに安全でなくなる危険性も小さいであろうと、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であろうが、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ろう。

ただし、この概念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。本節では、「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。

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。

また、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。なお、証明可能安全性を達成する暗号を構成する方法は、署名や守秘のための暗号に対しては確立されつつあるが、鍵共有のための暗号に対しては必ずしもそのような状況には至っていないと考えられる。

署名

（1） ACE Sign

ACE Sign は、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。若干強い数学的仮定が必要なものの，証明可能安全性を有することの証明が、他の署名機能をもつ暗号と異なり、補助関数を仮想的なものに置き換えることなく行えることが特徴である。また法（モジュラス）である合成数ｎと素因数ｐのサイズには以下の条件がある：1024≦｜ｎ｜≦16384、｜p｜≧512。さらに素因数の形も限定されている。なお、補助関数として用いられている共通鍵暗号はMARS に限定された仕様となっている。

参考情報：本応募暗号の提案者は，同名であるが仕様が本応募とは異なる暗号を暗号技術公募への応募後に発表している。

（2） ESIGN-signature

ESIGN-signature は、現時点においては安全性に大きな脅威を与える問題点は解消されている。ある仮定の下での証明可能安全性を有する。この仮定を満たすためには適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 署名方式（教科書的 RSA 署名方式およびRSA-PSS）と異なるため、法が RSA 署名方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。さらに、安全性は法 n における e 乗根近似問題の困難性に依存しており、これはｎの値と eの値に依存するという特徴がある。この点で、提案者の仕様書中の条件(ｅ≧5)、提案者の推奨パラメータ(ｅ≧8,｜ｐ｜＝｜ｑ｜≧320, ｜ｎ｜≧960)、ソフトウェア実装評価時の提案者による採用パラメータ(ｅ＝2^{10}, ｜ｐ｜＝｜ｑ｜＝384, ｜ｎ｜＝1152)、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案の条件(ｅ≧8, ｜ｎ｜≧1024)のように各所であげられた条件やパラメータ設定例が異なるので使用においては、十分な吟味が必要である。

（3） RSA-PSS

RSA-PSS は、現時点において安全性の上で特段の問題点は指摘されていない。現在多く使われている教科書的 RSA 署名方式は証明可能安全性を有することは確認されていないのに対し，RSA-PSS はある仮定の下での証明可能安全性を有していることが利点であるが、その仮定を満たすために適切なパラメータを選択することに注意を払う必要がある。

（4） DSA

FIPS 186-2 の DSA は、現時点において安全性の上で特段の問題点は指摘されていない。ただし、パラメータの選択にあたっては離散対数問題が困難になるように注意を払う必要がある。法のサイズは 512 ビット以上 1,024 ビット以下と規定されていて 1025 ビット以上にはできないため、達成しうる安全性には上限があることにも注意を払う必要がある。証明可能安全性を有することは確認されていない。なお、乱数生成方法の例として FIPS 186-2 Appendix 3 に記述されている方法は、最近その有効性については疑問が提

示されており、更なる検討が必要である。

（５） ECDSA in SEC1

ECDSA in SEC1 は、現時点において安全性の上で特段の問題点は指摘されていない。ECDSA in SEC1 の推奨パラメータのクラスである SEC2 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 SEC2 に含まれている Koblitz 曲線とよばれる楕円曲線は、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。証明可能安全性を有することは確認されていなかったが、最近、スキームとしての ECDSA がある仮定の下での証明可能安全性を有するという主張がある論文によりなされている。ただし、この主張の妥当性と、証明の方法の現実的効果とについて、本評価では結論を得るに至っていないため、更なる検討が必要である。

（６） MY-ELLTY ECMR-160/192/OEF-h

MY-ELLTY ECMR-160/192/OEF-h は、ハッシュ値のサイズが短すぎるため、2^{40} (または 2^{48})の計算量により、Birthday 攻撃による署名の存在的偽造が可能であるという点において安全性に問題なしとはいえず、長期間有効性を保つことが求められる署名方式としては推奨できない。なお、提案者の証明可能安全性の論述には誤りがあるため、証明可能安全性を有すると現時点においては認められていない。

守秘

（７） EPOC-1

EPOC-1 は、以下に示す点を除けば、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 方式と異なることなどにより、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。また、証明で十分な安全性を導くためには、パラメータの条件として提案者が示していない条件を追加する必要があるとの指摘がある。ただし、この指摘の妥当性について本評価では結論を得るに至っていないため、パラメータ選択条件の明確化について更なる検討が必要である。

（８）EPOC-2

EPOC-2 は、以下に示す点を除けば、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 方式と異なることなどにより、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。また、証明で十分な安全性を導くためには、パラメータの条件として提案者が示した条件とは異なる条件が必要であり、特に提案者の推奨パラメータでは十分でないとの指摘がある。ただし、この指摘の妥当性について本評価では結論を得るに

至っていないため、パラメータ選択条件の明確化について更なる検討が必要である。

（９）EPOC-3

EPOC-3 は、以下に示す点を除けば、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 方式と異なることなどにより、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。また、証明で十分な安全性を導くためには、パラメータの条件として提案者が示した条件とは異なる条件が必要であり、特に提案者の推奨パラメータでは十分でないとの指摘がある。さらに、拠り所とする GAP-素因数分解問題という比較的新しい問題の採用が適当であるかまだ見極められていないという指摘もある。ただし、これらの指摘の妥当性について本評価では結論を得るに至っていないため、更なる検討が必要である。

（１０） HIME-1

HIME-1 は仕様に曖昧さが存在し、そのままでは第三者が適切に実装することができない。提案者の証明可能安全性の論述には誤りがあるため、証明可能安全性を有すると現時点においては認められていない。HIME-1 の法である合成数の素因数分解の形が RSA 方式と異なるため、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。特に、パラメータの選択にあたっては、楕円曲線を使用した素因数分解法（ECM）が数体ふるい法（NFS）よりも効率的になる可能性があることに十分注意を払う必要がある。

（１１） HIME-2

HIME-2 は仕様に曖昧さが存在し、そのままでは第三者が適切に実装することができない。提案者の証明可能安全性の論述には誤りがあるため、証明可能安全性を有すると現時点においては認められていない。HIME-2 の法である合成数の素因数分解の形が RSA 方式と異なるため、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。特に、パラメータの選択にあたっては、楕円曲線を使用した素因数分解法（ECM）が数体ふるい法（NFS）よりも効率的になる可能性があることに十分注意を払う必要がある。

（１２）RSA-OAEP

RSA-OAEP は、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することが必要である。一般的な OAEP 変換は、適応的選択暗号文攻撃に対して強秘匿であるという性質を示すために十分でなかったが、RSA-OAEP については、適応的選択暗号文攻撃に対して強秘匿であることをある仮定の下で証明できることが確認されている。

（１３） ACE Encrypt

ACE Encrypt は、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。証明可能安全性を有することの証明が、他の守秘機能をもつ暗号と異なり、補助関数を仮想的なものに置き換えることなく行えることが特徴である。パラメータの条件としては、法 p のサイズに制限があること：1024≦｜p｜≦16384、および、パラメータ q のサイズが｜q｜= 256 と固定されていることがあげられる。なお、補助関数として用いられている共通鍵暗号は MARS に限定された仕様となっている。

参考情報：本応募暗号の提案者は，同名であるが仕様が本応募とは異なる暗号を暗号技術公募への応募後に発表している。

（１４） ECAES in SEC1

ECAES in SEC1 は、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有する。SEC2 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 SEC2 に含まれている Koblitz 曲線とよばれる楕円曲線は、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。

（１５） PSEC-1

PSEC-1 は、プリミティブ暗号化関数の問題で、証明可能安全性は確認されていないないという指摘がある。また、証明可能安全性が成り立つためには提案者の示していない条件がパラメータに課せられるという指摘があり、これが正しい場合、１回の暗号処理で安全に扱える平文の長さが著しく制限されることになる。ただし、これらの指摘の妥当性について本評価では結論を得るに至っていないため、更なる検討が必要である。

（１６） PSEC-2

PSEC-2 は、以下に示す点を除けば、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有している。しかし、その証明で十分な安全性を導くためには、パラメータの条件として提案者が示した条件とは異なる条件が必要であり、特に提案者の推奨パラメータでは十分でないとの指摘がある。ただし、この指摘の妥当性について本評価では結論を得るに至っていないため、パラメータ選択条件の明確化について更なる検討が必要である。

（１７） PSEC-3

PSEC-3 は、以下に示す点を除けば、現時点において安全性の上で特段の問題点は指摘されていない。ある仮定の下での証明可能安全性を有している。しかし、その証明で十分な安全性を導くためには、パラメータのサイズを明確に指定しなければならないとの指摘がある。また、復号アルゴリズムにある処理を追加しなければならないという指摘がある。さらに、拠り所とする楕円曲線上の GAP-DH 問題という比較的新

しい問題の採用が適当であるかまだ見極められていないという指摘もある。ただし、これらの指摘の妥当性について本評価では結論を得るに至っていないため、更なる検討が必要である。

鍵共有

（１８） DH

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/IS11770-3, Oakley, PGP)。今年度の評価対象は、基本的なスキームのみである。基本的スキームの使用に際しては、現時点において、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、最低限以下の３点に注意を払う必要がある。

A. 公開鍵とエンティティとの結びつきを保証する手段を確保する。
B. （更新を前提とする）セッション鍵の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。
C. 共有される鍵が乱数と見分けがつかなくするためには鍵導出関数を使用する。

（１９） ECDHS in SEC1

ECDHS in SEC1 は、現時点において、受動的攻撃に対して問題点は指摘されていないが、能動的攻撃に対して、最低限以下の２点に注意を払う必要がある。

A. 公開鍵とエンティティとの結びつきを保証する手段を確保する。
B. （更新を前提とする）セッション鍵の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。

SEC2 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 SEC2 に含まれている Koblitz 曲線とよばれる楕円曲線は、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。

（２０） ECMQVS in SEC1

ECMQVS in SEC1 は、現時点において、受動的攻撃に対して問題点は指摘されていないが、能動的攻撃に対して、最低限以下の２点に注意を払う必要がある。

A. 公開鍵とエンティティとの結びつきを保証する手段を確保する。
B. （更新を前提とする）セッション鍵の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。

SEC2 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 SEC2 に含まれている Koblitz 曲線とよばれる

楕円曲線は、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。

（２１）HDEF-ECDH

本提案は、スキームとしては、楕円曲線版 DH 方式（ECDH）の基本形(各エンティティが同じ楕円曲線を使うスキーム)とその変形版(エンティティ毎に異なる楕円曲線を使うスキーム)とから成り立っているが、提案の中心は、ある限定されたクラスの楕円曲線パラメータの生成法を示すことに重点が置かれている。ECDH の基本形の評価は、(18)DH の評価に準じる。基本系における基本的なスキームの使用に際しては、現時点において、受動的攻撃に対して問題点は指摘されていないが、能動的攻撃に対して、最低限以下の３点に注意を払う必要がある。

A. 公開鍵とエンティティとの結びつきを保証する手段を確保する。

B. （更新を前提とする）セッション鍵の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。

C. 共有される鍵が乱数と見分けがつかなくするためには鍵導出関数を使用する。

また、ECDH の変形版は基本系より安全性が向上するという提案者の主張が妥当であるかどうか、本評価においては結論が得られなかったため、更なる検討が必要である。

提案方法で生成された楕円曲線は、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。

4.2 機能による分類評価

4.2.1 署名

【署名用途の公開鍵暗号ソフトウェア実装結果】

詳細評価の対象となった署名用途の公開鍵暗号技術のうち、ソフトウェア実装評価を行った応募は、

（1）ESIGN 署名

（2）ECDSA in SEC1

（3）MY-ELLITY ECMR-160/192/OEF-h

の 3 方式である。詳細評価対象となった署名用途の暗号技術のうち RSA-PSS、DSA に関しては、その他評価が必要と判断した暗号技術であり、多くの実装実績を有しているため、今回のソフトウェア実装評価の必要がないと判断した。また、ACE Sign に関しては、応募された暗号技術ではあったが、応募書類提出後に仕様が変更された。応募された技術に対応したソフトウェア実装評価を行わなかった。

ソフトウェア実装評価の対象とした上記の 3 方式に関しては、実装可能なレベルにあると判断される。この 3 方式に関しては、実装パラメータ選択にあたって、1,024 ビットの RSA 暗号相当以上の安全性を有することを目安として、応募者に選択いただいた。

鍵生成、署名生成、署名検証の実行速度の目安（測定結果）及び目安となるコードサイズは以下の通りである。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠		実装パラメータと位置づけ	その他
ESIGN 署名	素因数分解問題(IF)		合成数のサイズは、384 bit×3＝1152 bit	ESIGN のセキュリティパラメータは、10 ビット、2 の 10 乗(1024)である。仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。
MY-ELLITY ECMR-192-h	ECDLP	192 bit 素体	RSA 1536 bit 相当	MOV、FR 帰着攻撃、SSSA 攻撃について安全性を検証(判定条件をクリア)。パラメータ固定。バイナリ組み込み。予備計算結果をバイナリ組み込み。テーブル参照。
MY-ELLITY ECMR-OEF-h		OEF	RSA 1024 bit 相当	
MY-ELLITY ECMR-160-h		160 bit 素体	RSA 1024 bit 相当	
ECDSA in SEC1		160 bit 素体 163bit 標数 2 の体	RSA 1024 bit 相当	・ 楕円曲線の 2 つの係数が、(コントロールすることのできない)SHA-1 の出力値によって関係付けられているため、任意に選択されたものであることが検証可能(randomly verifiable)。 ・ X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。

〔鍵対生成の速度（目安)〕

測定対象		平均実行時間	備考
ESIGN 署名		50.8 ms	素数生成、乱数生成は含まれていない。合成数の長さは、384 bit×3＝1152 bit。
		452.8 ms	乱数生成、素数生成を含めた測定値。
MY-ELLTY ECMR -192-h		0.8 ms	乱数生成を含む。 但し、実使用時には、乱数生成部の入れ替えが必要。当然時間も変化する。
MY-ELLTY ECMR -OEF-h		0.7 ms	
MY-ELLTY ECMR -160-h		0.7 ms	
ECDSA in SEC1	ECDSA 160 bit 素体	1.9 ms	鍵対生成
		5.8 ms	鍵対の検証。但し、鍵対の検証処理は、自分で鍵を生成したときには不必要。
	ECDSA 163 bit 標数 2 の体	3.2 ms	鍵対生成
		8.0 ms	鍵対の検証。但し、鍵対の検証処理は、自分で鍵を生成したときには不必要。

［署名生成の速度（目安)］

測定対象		平均実行時間	補助関数	備考	
ESIGN-Signature		9.2 ms	SHA-1	署名対象データのサイズ : 31 KB	
		49.1 ms		署名対象データのサイズ : 178 KB	
MY-ELLTY ECMR-192-h		2.0 ms	SHA-1	署名対象データのサイズ : 31 KB	乱数生成を含む。SHA-1 を補助関数として使用。ファイルの長さに比例して、SHA-1 の処理時間が増加する。
		9.2 ms		署名対象データのサイズ : 178 KB	
MY-ELLTY ECMR-OEF-h		1.9 ms		署名対象データのサイズ : 31 KB	
		9.6 ms		署名対象データのサイズ : 178 KB	
MY-ELLTY ECMR-160-h		1.9 ms		署名対象データのサイズ : 31 KB	
		9.7 ms		署名対象データのサイズ : 178 KB	
ECDSA in SEC1	ECDSA 160bit 素体	3.7 ms	SHA-1	署名対象データのサイズ : 31 KB	
		11.1 ms		署名対象データのサイズ : 178 KB	
	ECDSA163 bit 標数 2 の体	5.0 ms		署名対象データのサイズ : 31 KB	
		13.1 ms		署名対象データのサイズ : 178 KB	

［署名検証の速度（目安)］

測定対象		平均実行時間	備考	
ESIGN 署名		6.6 ms	署名対象データのサイズ : 31 KB	
		44.3 ms	署名対象データのサイズ : 178 KB	
MY-ELLITY ECMR-192-h		5.4 ms	署名対象データのサイズ : 31 KB	検証の正当性は、チェックするようになっているが、今回の実装では、その結果は出力していない。
		12.8 ms	署名対象データのサイズ : 178 KB	
MY-ELLITY ECMR-OEF-h		4.4 ms	署名対象データのサイズ : 31 KB	
		11.9 ms	署名対象データのサイズ : 178 KB	
MY-ELLITY ECMR-160-h		4.3 ms	署名対象データのサイズ : 31 KB	
		11.9 ms	署名対象データのサイズ : 178 KB	
ECDSA in SEC1	160bit 素体	9.7 ms	署名対象データのサイズ : 31 KB	
		17.2 ms	署名対象データのサイズ : 178 KB	
	163bit 標数 2 の体	13.6 ms	署名対象データのサイズ : 31 KB	
		21.4 ms	署名対象データのサイズ : 178 KB	

4.2.2 守秘

守秘のカテゴリーは、EPOC、HIME-1、HIME-2、RSA-OAEP、ACE Encrypt、ECAES in SEC1、PSEC の７方式が評価の対象であるが、EPOC と PSEC は方式(スキーム)としてそれぞれ 3 つが提案されているので実質的には 11 方式が対象である。次頁の表に各方式の特徴と安全性をまとめる。

特徴欄

- 次の視点で各方式の機能を分類した。各方式は、スキーム処理 1 回に対して暗号化できる平文ブロック長に制限のあるもの(L1)、制限のないもの(L2)、さらに鍵配送フェーズと暗号通信フェーズを分けたセッション利用を想定したもの(L3)に分類できる。L1 は共通鍵暗号や MAC 関数に利用する秘密鍵の配送用途を考慮した設計であり、L2･L3 は共通鍵暗号系とのハイブリッド方式であって、一般の暗号通信用途(メッセージ認証機能付き)までを考慮した設計と、それぞれとらえることができる。
- 平文長、暗号文長、公開鍵長、秘密鍵長の各実効長は、仕様書にて推奨されているパラメータサイズを基に、RSA 1024 bit 相当の強度を持つと想定される典型的なパラメータサイズを一覧にした。なお、推奨パラメータサイズが明記されていない方式は、仕様書内の実装性能の記述部にて利用されているパラメータサイズを推奨値とみなした。

  さらに、なるべく統一的に比較できるように次の点を考慮した。
  - 各種パラメータサイズを可変にしている方式では、そのパラメータサイズを特定するために必要なパラメータは実効長からはずした。関数を指定する関数 ID 等のパラメータも同様。
  - 機能で L2･L3 を実現している方式の共通鍵暗号や MAC 関数は仕様書で推奨されている関数を前提とした。共通鍵暗号にバーナム暗号が推奨されている方式は、バイト単位のバーナム暗号を前提とした。
  - 楕円曲線は素体上で定義された曲線を前提とした。余因子は 1 とした。

安全性欄

- 各方式の基本関数(プリミティブ)とそれがベースとしている安全性の仮定を一覧にした。
- 各方式(スキーム)の秘匿機能に関する安全性は、想定する攻撃方法とそれに対する安全性目標の組で議論される。暗号理論上は、最強の攻撃法である「適応的選択暗号文攻撃」に対して、「強秘匿」であるならば、守秘用途には最高の安全性レベルを実現していると判断できる(この性質を「IND-CCA2」と略記する)。各方式は、使用する関数のランダム性を仮定し、さらにパラメータサイズでの数論問題の困難性を前提にして、IND-CCA2 であることの証明が試みられている。表ではこのランダム性仮定と前提とする数論問題を一覧にした。

方式		EPOC-1 EPOC-2 EPOC-3	HIME-1	HIME-2	RSA-OAEP	ACE Encrypt	ECAES in SEC1	PSEC-1 PSEC-2 PSEC-3
特徴	機能分類	EPOC-1/2/3は順にL1/L2/L3。	L1	L1	L1	L2	L2	PSEC-1/2/3は順にL1/L2/L3。
	平文実効長	[EPOC-1] 128 bit [EPOC-2] 任意長 L(M) byte [EPOC-3] 任意長 L(M) byte	510 bit	768 bit	688 bit	任意長 L(M) byte	任意長 L(M) byte	[PSEC-1] 128 bit [PSEC-2] 任意長 L(M) byte [PSEC-3] 任意長 L(M) byte
	暗号文実効長	[EPOC-1] 1152 bit [EPOC-2] C_1:1152bit C_2:L(M)byte [EPOC-3] C_1:1152bit C_2:L(M)byte C_3:128bit	1024 bit	C:1024 bit a:2 bit	1024bit	s:128bit u_1:1024bit u_2:1024bit v:1024bit e:L(M)+16*(L(M)/1024)byte	R:320bit EM:L(M)byte D:160bit	[PSEC-1] C_1:320bit c_2:160bit [PSEC-2] C_1:320bit c_2:160bit c_3:L(M)byte [PSEC-3] C_1:320bit c_2:160bit c_3:L(M)byte c_4:128bit
	公開鍵実効長	n:1152bit g:1152bit h:1152bit	n:1024bit	n:1024bit	n:1024bit e:17bit	P:1024bit q:256bit g_1:1024bit g_2:1024bit c:1024bit d:1024bit h_1:1024bit h_2:1024bit	Q_V:320bit 楕円曲線パラメータ: 160bit×6	W:320bit 楕円曲線パラメータ: 160bit×6
	秘密鍵実効長	p:384bit g_p:768bit	p:256bit q:256bit α:256bit β:256bit z:256bit	p_1:256bit p_2:256bit p_3:256bit p_4:256bit z_1:256bit z_2:256bit z_3:256bit	p:512bit q:512bit dP:512bit dQ:512bit qInv:512bit	w:256bit x:256bit y:256bit z_1:256bit z_2:256bit	D_V:160bit	s:160bit

方式		EPOC-1 EPOC-2 EPOC-3	HIME-1	HIME-2	RSA-OAEP	ACE Encrypt	ECAES In SEC1	PSEC-1 PSEC-2 PSEC-3
安全性	暗号プリミティブ	Okamoto-Uchiyama関数を基本関数とする。 素因数分解(p^2q 型)の安全性に基づく。	Rabin 関数を基本関数とする。 素因数分解(p^kq 型)の安全性に基づく。	Rabin 関数を基本関数とする。 素因数分解($\prod_{i=1}^{d} p_i$ 型)の安全性に基づく。	RSA 関数を基本関数とする。 素因数分解(pq 型)の安全性に基づく。	Cramer-Shoup 関数(ElGamal 関数の変形)を基本関数とする。 離散対数問題の安全性に基づく。	楕円曲線上の ElGamal 関数を基本関数とする。楕円曲線上の離散対数問題の安全性に基づく。	楕円曲線上の ElGamal 関数を基本関数とする。 楕円曲線上の離散対数問題の安全性に基づく。
	暗号スキーム	以下に挙げる問題の困難性を前提に IND-CCA2。 (ランダムオラクルモデル) [EPOC-1] p-部分群問題 [EPOC-2] 素因数分解問題 [EPOC-3] GAP 素因数分解問題	仕様の不備や証明上の問題があり、証明可能安全性を有することは確認されていない。	仕様の不備や証明上の問題があり、証明可能安全性を有することは確認されていない。	RSA 関数の逆変換問題の困難性を前提に IND-CCA2。 (ランダムオラクルモデル)	決定 DH 問題の困難性を前提に IND-CCA2。 (汎用ハッシュ関数の仮定、共通鍵暗号の擬似ランダム性仮定)	楕円曲線上の適応的ハッシュ DH 問題の困難性を前提に IND-CCA2。 (共通鍵暗号と MAC 関数のランダム性仮定)。 なお、ランダムオラクル仮定の下では以下になるとの指摘がある。 楕円曲線上の GAP DH 問題の困難性を前提に IND-CCA2。 (ランダムオラクルモデル)	以下に挙げる問題の困難性を前提に IND-CCA2。 (ランダムオラクルモデル) [PSEC-1] 楕円曲線上の部分決定 DH 問題 (ECPDDH) [PSEC-2] 楕円曲線上の DH 問題 (ECDH) [PSEC-3] 楕円曲線上の GAP DH 問題 (EC-GAP-DH)

【守秘用途の公開鍵暗号ソフトウェア実装結果】

詳細評価の対象となった守秘用途の公開鍵暗号技術のうち、ソフトウェア実装評価を行った応募は、

（１）RSA OAEP

（２）EPOC-1/ EPOC-2/ EPOC-3

（３）HIME-1/ HIME-2

（４）ECAES in SEC1

（５）PSEC-1/ PSEC-2/ PSEC-3

の５方式である。詳細評価対象となった守秘用途の暗号技術のうち ACE Encrypto に関しては、応募された暗号技術ではあったが、応募書類提出後に仕様が変更されこともあり、応募された技術に対応したソフトウェア実装評価をしなかった。

ソフトウェア実装評価の対象とした上記の 5 方式に関しては、実装可能なレベルにあると判断される。HIME-1、HIME-2 に関しては、ソフトウェア実装評価を行う際に、素数判定条件等の実装に必要な情報が提出された。この新たに追加された情報は、「暗号技術仕様書」には記載されていない。

また、今回実装評価の対象となった 5 方式のうち、RSA-OAEP と ECAES in SEC1 の２方式は実装の完成度は製品版ないしそれに近いレベルである。

この 5 方式に関しては、実装パラメーター選択の選択にあたって、1,024 ビットの RSA 暗号相当以上の安全性を有することを目安として、応募者に選択いただいた。

鍵生成、暗号化、復号の実行速度の目安（測定結果）及び目安となるコードサイズは以下の通りである。

〔特徴及び測定パラメータ(1/2)〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
RSA OAEP	素因数分解問題 (IF)	公開鍵モジュロ値長 1024bit	中国人剰余定理(CRT)を使用し、高速化。 公開指数：65537
EPOC-1		EPOC の合成数の長さは 、 384bit × 3 ＝ 1,152bit。	評価用 EPOC のデータサイズは、128 ビット。 素数判定は、Miller-Rabin 法を使用。
EPOC-2			
EPOC-3			
HIME-1		法 N は、$N=p^3q$ (d=3) の形式であり、p,q は各々256bit である。N=1,024bit の RSA 暗号（守秘同等以上の強度）	p,q の選定にあたっては、p±1 法を考慮。 公開鍵は、仕様書上(N,k,d)の 3 種であるが、今回の実装評価では、d=3 に固定した。 べき乗演算においてモンゴメリー乗算及び 4 array 法を使用。
HIME-2		法 N は、$N=p_1p_2p_3p_4$ （d=3）の形式であり、p_1，p_2，p_3，p_4は各々256bit である。N=1,024bit の RSA 暗号（守秘同等以上の強度）	・p_1，p_2，p_3，p_4の選定にあたっては、p±1 法を考慮。 ・暗号技術仕様書では n のビット長が 1023 ビット以上であることおよび p_1,p_2,p_3,p_4 が全て異なることは要求していなかったが、問題が生じる場合があるため今回の実装では条件の追加を行った。 ・ べき乗演算においてモンゴメリー乗算及び 4 array 法を使用。

〔特徴及び測定パラメータ(2/2)〕

測定対象		安全性の根拠	実装パラメータと位置づけ	その他
ECAES in SEC1	160bit 素体	ECDLP	RSA1024bit 相当	・ SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。 ・ 楕円曲線の２つの係数が、(コントロールすることのできない)SHA-1の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。 ・ X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。 ・ 各種特殊攻撃を適用できないことも確かめられている。 ・ 最適化すればさらなる高速化の可能性あり。 ・ 任意のパラメータに適用できる高速手法を使用。 ・ パラメータには依存しない実装を行っている。
	163bit 標数 2 の体		RSA1024bit 相当。	
PSEC-1			PSEC の各パラメータは、160bit。RSA1024bit 相当	評価用の PSEC のデータサイズは、128 ビット。 楕円曲線のパラメータが必要。他の手法で計算された結果あるいは NIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどメインプログラムに与える。 フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。
PSEC-2				
PSEC-3				

〔鍵対生成の速度(目安)〕

測定対象		平均実行時間	備考
RSA-OAEP		2946.5 ms	e=17
		3405.5 ms	e=65537
EPOC-1		73.9 ms	乱数生成は含まない
		417.6 ms	乱数生成、素数生成を含めた測定値
EPOC-2		73.9 ms	乱数生成は含まない
		577.0 ms	乱数生成、素数生成を含めた測定値。
EPOC-3		73.9 ms	乱数生成は含まない
		743.3 ms	乱数生成、素数生成を含めた測定値。
HIME-1		934.0 ms	素数生成を含む。
		785.0 ms	合成数のサイズは、256×4=1,024 ビット。 各素数は、ｐ±１法の検査のみ実装。
HIME-2		2845.0 ms	素数生成を含む。
		1829.0 ms	合成数のサイズは、256×4=1,024 ビット。 各素数は、ｐ±１法の検査のみ実装。
ECAES in SEC1	160bit 素体	1.9 ms	鍵対生成
		5.8 ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
	163bit 標数 2 の体	3.2 ms	鍵対生成
		8.0 ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
PSEC-1		11.3 ms	なし
PSEC-2		11.8 ms	なし
PSEC-3		11.5 ms	なし

［暗号化の速度（目安）］

測定対象		平均実行時間	備考
RSA OAEP		4.2ms	e=17
EPOC-1		13.9ms	なし
EPOC-2		10.9ms	なし
EPOC-3		11.0ms	なし
HIME-1		140.6ms	HIME1 のデータサイズは、256 ビット。
HIME-2		0.4ms	HIME2 のデータサイズは、256 ビット。
ECAES in SEC1	160bit 素体	8.5ms	なし
	163bit 標数2の体	12.5ms	なし
PSEC-1		24.0ms	なし
PSEC-2		24.2ms	なし
PSEC-3		22.9ms	なし

［復号の速度（目安）］

測定対象		平均実行時間	備考
RSA OAEP		84.2ms	e=17
EPOC-1		21.9ms	なし
EPOC-2		18.9ms	なし
EPOC-3		8.3ms	なし
HIME-1		24.7ms	HIME-1 のデータサイズは、256 ビット。
HIME-2		15.3ms	HIME-2 のデータサイズは、256 ビット。
ECAES in SEC1	160bit 素体	5.4ms	なし
	163bit 標数 2 の体	8.7ms	なし
PSEC-1		24.1ms	なし
PSEC-2		24.6ms	なし
PSEC-3		12.0ms	なし

［コードサイズ（参考値）］

測定対象	コードサイズ	備考
RSA OAEP	62,413byte	性能測定用実行形式テストプログラムのサイズ
EPOC-1	177,058byte	C言語（Intel C/ C++）ソースコードサイズ
EPOC-2	187,662byte	
EPOC-3	186,851byte	
HIME-1	2631step	C言語（Intel C/ C++）
HIME-2	2666step	
ECAES in SEC1	356,352byte	性能測定用実行形式テストプログラムのサイズ
PSEC-1	189,334byte	C言語（Intel C/ C++）ソースコードサイズ
PSEC-2	202,333byte	
PSEC-3	196,986byte	

4.2.3 鍵共有

鍵共有のカテゴリーは、DH、ECDHS in SEC1、ECMQVS in SEC1、HDEF-ECDH の 4 暗号が評価の対象である。本章では、鍵共有カテゴリーに属する 4 暗号の概要を記述する。DH は離散対数問題(DLP)、ECDHS in SEC1、ECMQVS in SEC1、HDEF-ECDH は楕円離散対数問題(ECDLP)に安全性の根拠をおく鍵共有法である。下表にこれらの概要をまとめる。

公開鍵暗号（鍵共有）の一覧表

		DH	ECDHS in SEC1	ECMQVS in SEC1	HDEF-ECDH
特徴	使用するパラメータ	乗法群 Z_p^* を使用する。	ランダムに選ばれた楕円曲線または Koblitz 曲線と呼ばれる楕円曲線を使用する。推奨する楕円曲線パラメータが具体的に指定されている。ただし、この鍵共有法は、推奨されたパラメータ以外の楕円曲線でも動作する。	ランダムに選ばれた楕円曲線または Koblitz 曲線と呼ばれる楕円曲線を使用する。推奨する楕円曲線パラメータが具体的に指定されている。ただし、この鍵共有法は、推奨されたパラメータ以外の楕円曲線でも動作する。	素体上トレース 3、かつ discriminant が小さな CM 体を持つ楕円曲線を使用する。
	推奨パラメータサイズ	1024 bit 以上を推奨する。	素体上 112 bit から 512 bit、標数 2 の体上 113 bit から 571 bit までの楕円曲線パラメータが具体的にリストされている。	素体上 112 bit から 512 bit、標数 2 の体上 113 bit から 571 bit までの楕円曲線パラメータが具体的にリストされている。	トレース 3 という条件以外に具体的楕円曲線の指定は無いが、160 bit 以上が推奨される。
安全性	暗号プリミティブ	Diffie Hellman 問題の難しさに安全性の根拠をおく。適切なパラメータを選択すれば、既知攻撃法に対して安全である。	楕円 Diffie Hellman 問題の難しさに安全性の根拠をおく。既知攻撃法に対して安全である。	楕円 Diffie Hellman 問題の難しさに安全性の根拠をおく。既知攻撃法に対して安全である。	楕円 Diffie Hellman 問題の難しさに安全性の根拠をおく。既知攻撃法に対して安全である。

	暗号スキーム	DH スキームには多くのバリエーションが存在する。したがって、これらバリエーション個々の安全性評価が必要である。最も基本的なスキームを使用した場合、受動的攻撃に対して安全である。しかし、能動的攻撃に対しては問題がある。能動的攻撃に対して安全で、かつ、forward secrecy を満足するためには、電子署名と組み合わせるなど、仕様の改定が必要である。	受動的攻撃に対して安全である。しかし、このままの仕様では、能動的攻撃に対しては安全性が証明されていない。また、forward secrecy も満足しない。能動的攻撃に対して安全で、かつ、forward secrecy を満足するためには、電子署名と組み合わせるなど、仕様の改訂が必要である。 Koblitz 曲線は高速性を特徴とする。この曲線を使うことは、現時点では安全性に問題は無い。しかし、この曲線は限定されたクラスの曲線である。したがって、将来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。	受動的攻撃に対して安全である。しかし、このままの仕様では、能動的攻撃に対しては安全性が証明されていない。また、forward secrecy も満足しない。能動的攻撃に対して安全で、かつ、forward secrecy を満足するためには、電子署名と組み合わせるなど、仕様の改訂が必要である。 Koblitz 曲線は高速性を特徴とする。この曲線を使うことは、現時点では安全性に問題は無い。しかし、この曲線は限定されたクラスの曲線である。したがって、将来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。	受動的攻撃に対して安全である。しかし、このままの仕様では、能動的攻撃に対しては問題がある。能動的攻撃に対して安全で、かつ、forward secrecy を満足するためには、電子署名と組み合わせるなど、仕様の改訂が必要である。 トレース 3 かつ discriminant が小さな CM 体を持つという限定されたクラスの楕円曲線を用いることは、現時点では安全性に問題は無い。しかし、この曲線は限定されたクラスの曲線である。したがって、将来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。 使用する楕円曲線を、各ユーザーで同一にするスキームと、ユーザーごとに異なるものにするスキームとの2種類がある。ユーザーごとに異なる曲線を用いるスキームに関しては、安全性に関してさらなる検討が必要である。

共有できる鍵のサイズ	暗号プリミティブのみを用いた場合、使用する群のサイズ(素数 p のサイズ)に依存する。例えば、1024 bit の素数 p を使用すれば、1024 bit までの鍵が共有できる。 使用する群パラメータのサイズを超えた大きさの鍵を共有したい場合、ANSI X9.63 などに記述された鍵導出関数などを、暗号スキームに使用する必要がある。	鍵導出関数として、ANSI X9.63 に記載されている“Key Derivation Function”の使用が指定されている。この鍵導出関数はハッシュ関数を使用しており、ハッシュ関数の出力が hashlen ビットであるとすると、hashlen × $(2^{32}-1)$ bit 未満の鍵を共有できる。推奨されるハッシュ関数として、SHA-1 が挙げられている。SHA-1 を用いた場合は、160 × $(2^{32}-1)$ bit 未満の鍵を共有できる。ただし、他のハッシュ関数を使用しても、この鍵共有スキームは動作する。	鍵導出関数として、ANSI X9.63 に記載されている“Key Derivation Function”の使用が指定されている。この鍵導出関数はハッシュ関数を使用しており、ハッシュ関数の出力が hashlen ビットであるとすると、hashlen × $(2^{32}-1)$ bit 未満の鍵を共有できる。推奨されるハッシュ関数として、SHA-1 が挙げられている。SHA-1 を用いた場合は、160 × $(2^{32}-1)$ bit 未満の鍵を共有できる。ただし、他のハッシュ関数を使用しても、この鍵共有スキームは動作する。	暗号プリミティブのみを用いた場合、使用する楕円曲線パラメータのサイズに依存する。例えば、160 bit の楕円曲線を使用すれば、160 bit までの鍵が共有できる。 使用する楕円曲線パラメータのサイズを超えた大きさの鍵を共有したい場合、ANSI X9.63 などに記述された鍵導出関数などを、暗号スキームに使用する必要がある。

【鍵共有用途の公開鍵暗号ソフトウェア実装結果】

詳細評価の対象となった鍵共有用途の公開鍵暗号技術のうち、ソフトウェア実装評価を行った暗号技術は、

（1）ECDHS in SEC1

（2）ECMQVS in SEC1

（3）HDEF-ECDH

の 3 方式である。詳細評価対象となった鍵共有用途の暗号技術のうち DH に関しては、その他評価が必要と判断した暗号技術であり、多くの実装実績を有しており、今回のソフトウェア実装評価の必要がないと判断した。

ソフトウェア実装評価の対象とした上記の 3 方式に関しては、実装可能なレベルにあると判断される。

この 3 方式に関しては、実装パラメータ選択にあたって、1,024 ビットの RSA 暗号相当以上の安全性を有することを目安として、応募者に選択いただいた。

鍵生成、暗号化、復号の実行速度の目安（測定結果）及び目安となるコードサイズは以下の通りである。

〔特徴及び測定パラメータ〕

測定対象		安全性の根拠	実装パラメータと位置づけ	その他
ECDHS in SEC1	160bit 素体	ECDLP	N = 1024 bit の RSA 暗号（守秘）同等以上の強度。	SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。 任意に選択されたものであることが検証可能 (randomly verifiable)。 X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。 各種特殊攻撃を適用できないことも確かめられている。 パラメータには依存しない高速実装法を行っている。
	163bit 標数 2 の体		N = 1024 bit の RSA 暗号（守秘）同等以上の強度。	
ECMQVS in SEC1	160bit 素体		N = 1024 bit の RSA 暗号（守秘）同等以上の強度。	
	163bit 標数 2 の体		N = 1024 bit の RSA 暗号（守秘）同等以上の強度。	
HDEF-ECDH	パラメータサイズは、160 bit		N = 1024 bit の RSA 暗号（守秘）同等以上の強度。	使用した楕円パラメータは、MOV(FR)帰着攻撃、SSSA 攻撃に対して安全であることを検証している。 評価用コードではベースポイントを固定。 任意点のべき倍演算と同様のべき倍演算アルゴリズム（符号付き 2 進法とウィンドウ法の組合わせ）を使用。 固定点アルゴリズムを利用することで高速化が望める。 定義体上での乗算剰余演算（160×160→160 ビット）についてはアセンブラで記述。 基本形の測定。 任意点べき倍算アルゴリズム使用。 乗算剰余演算アセンブラ

〔鍵対生成〕

測定対象		平均実行時間	備考
ECDHS in SEC1	160 bit 素体	1.9 ms	鍵対生成。
		5.8 ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
	163 bit 標数 2 の体	3.2 ms	鍵対生成。
		8.0 ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
ECMQVS in SEC1	160 bit 素体	1.9 ms	鍵対生成。
		5.8 ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
	163 bit 標数 2 の体	3.2 ms	鍵対生成。
		8.0 ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
HDEF-ECDH		1.5 ms	片側処理。鍵サイズ : 160 bit。

［鍵共有処理（片側)］

測定対象		平均実行時間	備考
ECDHS in SEC1	160 bit 素体	6.6 ms	
	163 bit 標数 2 の体	8.8 ms	
ECMQVS in SEC1	160 bit 素体	13.2 ms	
	163 bit 標数 2 の体	16.9 ms	
HDEF-ECDH		1.8 ms	鍵サイズ : 160 ビット。

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
ECDHS in SEC1	356,352 byte	性能測定用実行形式 TP のサイズ。
ECMQVS in SEC1	356,352 byte	
HDEF-ECDH	73,758 byte	定義体上での乗算剰余演算(160×160→160 ビット)についてはアセンブラで記述。その他はＣ言語。

4.3　安全性の根拠に基づく評価

4.3.1　素因数分解問題

基盤となる代数系に整数nの剰余環を用い、nを素因数分解することの困難さによって暗号プリミティブの安全性を主張する次の公開鍵暗号系を、素因数分解問題という共通の切り口で評価する。

・守　秘: RSA-OAEP、EPOC-1、 EPOC-2、 EPOC-3、HIME-1、HIME-2

・鍵共有: なし

・認　証: なし

・署　名: RSA-PSS、ESIGN-signature、ACE Sign

（1）暗号プリミティブの安全性

本項で評価する暗号系の暗号プリミティブは、基盤となる代数系 $\mathbf{Z}_n := \mathbb{Z}/n\mathbb{Z}$の法$n$が素因数分解されれば安全でなくなるという共通の性質をもつ。各暗号系の法nの形は次のとおりである。

機能	守秘				署名		
暗号系	RSA-OAEP	EPOC-1 EPOC-2 EPOC-3	HIME-1	HIME-2	RSA-PSS	ESIGN-signature	ACE Sign
$n=$	pq	p^2q	p^kq	$\prod_{i=1}^{d} p_i$	pq	p^2q	pq
	($p,q,p_1,\ldots,p_d$は奇素数)						

(a)**問題の複雑さによる評価**

各nを素因数分解することは、確かに一般の素因数分解問題に帰着する。しかし、それぞれの方式のnの形には特徴があり、細かく見れば、nを分解する問題の複雑さに微妙な差異が認められる。

これを示すために、次の6 個の問題(部分関数)を定義する。

IntegerFactoring:

入力$n(>1)$に対して、$n = p_1^{e_1}\cdots p_k^{e_k}$となる　$\{(p_1,e_1),\ldots,(p_k,e_k)\}(p_i:$素数$, e_i \geqq 1)$を出力する。

Divisor:

入力$n(>1)$に対して、nが合成数ならば$n=ab$となる$\{a,b\}(1<a,b<n)$を出力し、素数の場合はnを出力する。

SquarefreePart:

入力$n(>1)$に対して、$n=r^2s$かつsがsquarefreeとなる$\{r,s\}$を出力する。

Factorpq:

入力$n(>1)$に対して、$n=pq$となる素数の組$\{p,q\}$を出力する。

Factorp^{k}q(kは定数>1):

入力$n(>1)$に対して、$n = p^k q$となる素数の組$\{p,q\}$を出力する。

Factorp$_1$...p$_d$(dは定数>2):

入力$n(>1)$に対して、$n = p_1 \cdots p_d$となる素数のd項組$\{p_1,\cdots p_d\}$を出力する。

いま、これらの関数の難しさを比較するために、帰着の概念を導入する。$A \leqq_\alpha^P B$とは、関数Aが関数Bに$\leqq_\alpha^P$帰着性の意味で帰着することを意味する。ここに$\alpha \in \{m, \ell - tt, T\}$($\ell$は正整数の定数)であり、$\alpha$によって帰着の態様に違いはあるものの、直観的には$\alpha$の如何にかかわらず、「$A \leqq_\alpha^P B$とは、関数$B$を計算する力があれば関数$A$も計算できる」と読むことができる。細かく言えば、$A \leqq_m^p B \Rightarrow A \leqq_{\ell-tt}^P B \Rightarrow A \leqq_T^P B$である。また、ある特定の$\alpha$について、$A \leqq_\alpha^P B$かつ$B \leqq_\alpha^P A$のとき、$A \equiv_\alpha^P B$と書く。

このとき、これらの関数の帰着関係が次のようになることは容易に導ける。

Factorpq$\leqq_m^P$Divisor$\equiv_T^P$IntegerFactoring

Factorp$_1$...p$_d$$\leqq_T^P$Divisor$\equiv_T^P$IntegerFactoring

Factorp^{k}q$\leqq_\beta^P$SquarefreePart$\leqq_T^P$Divisor$\equiv_T^P$IntegerFactoring

(βは$k=2$のときm、$k>2$のとき1-tt)

すなわち、IntegerFactoringを多項式時間で計算するアルゴリズムが存在するならば、上記のすべては多項式時間で計算される。しかし、IntegerFactoringやDivisorがSquarefreePart に帰着するかどうか不明である。また、FactorpqやFactorp$_1\cdots$p$_d$がSquarefreePartに帰着するかどうかも知られていない。したがって現状では、SquarefreePart を多項式時間で計算するアルゴリズムが発見された場合は、Factorp^{k}q は多項式時間で計算されるが、IntegerFactoring、DivisorFactorpq、Factorp$_1\cdots$p$_d$などはすべて、難しい問題のまま生き残る可能性が理論上はある。

(b)**素因数分解アルゴリズムによる評価**

数体ふるい法(NFS)の実行時間は入力nのサイズに依存しており、楕円曲線法(ECM)の実行時間は入力nの最小素因数のサイズに依存している。したがって、ここで評価しようとしているすべての方式において、法nのサイズを同一とした場合には、NFS による分解では理論上は差異がないが、ECM を適用すれば、

$n = p^2 q$ や $n = p_1 \cdots p_d$ の法を用いている方式と $n = pq$ を用いている方式では差が現れる。これは、最小素因数サイズの違いによるものであり、少なくとも1個の素因数を求めるためだけならば、ECM のほうが実行時間が速い場合がある。そして一般に、$n = pq$ や $n = p^k q$ (k : 定数>1)の場合、1個の素因数を知れば残りは容易に完全分解される。いま、法 n のサイズを $|n|$ ビット、最小素因数のサイズを $|n|/d$ ビット(d≧2)で表したとき、「ECMによる実行時間≦NFS による実行時間」が成立する場合は、平均的実行時間を使った不等式により、

$$\exp\left((1.414+o(1))\left(\frac{|n|\log_e 2}{d}\right)^{1/2}\left(\log_e\left(\frac{|n|\log_e 2}{d}\right)\right)^{1/2}\right)$$

$$\leqq \exp\left((1.901+o(1))(|n|\log_e 2)^{1/3}(\log_e(|n|\log_e 2))^{2/3}\right)$$

と近似的に書ける。これらの平均的実行時間の評価はもともと厳密ではなく、また上の不等式では $o(1)$ ($n \to \infty$ で0になる項)が残っているため、厳密な評価はさらに難しいが、$|n|$ と d の組合せによっては注意を要する。なお、$n = p^k q$ (k : 定数>1)の形の合成数に対して特化した素因数分解法としてLFM (LatticeFactoring Method)がある。LFM の平均的実行時間は、$k = c\log_e p$ (c : 定数)であれば多項式時間となるが、k が小さいとき(例えば $k = 2$ のとき)は指数関数時間と見積もられており、$n = p^2 q$ の分解にLFM が有効であるという事実はない。

（2）スキームの安全性

(a)**守秘暗号系**

守秘を機能とする暗号系について、スキームの安全性モデルを目標(GOAL)と攻撃方法(ATK)の組を使って定義する。

目標としては、“ 一方向性(one-way:OW) ”、“ 識別不可能性(indistinguishability:IND) ”、“ 頑強性(non-malleability:NM) ”の3 つが挙げられる。一方向性とは、暗号文 y を与えられた攻撃者が y を復号して平文 x の全体を求められないことである。識別不可能性とは、2つの平文 x, x' といずれかの暗号文 y を与えられた攻撃者が、y が x, x' いずれの暗号文であるか識別できないことである。また、強秘匿性とは、暗号文 y を与えられた攻撃者が平文 x の任意の部分情報を知り得ないことである。さらに、頑強性とは、暗号文 y を与えられた攻撃者が y の平文 x に対してある関係を満たす平文 x' の暗号文 y' を求められないことである。適応的選択暗号文攻撃（後述）に対しては、識別不可能性は強秘匿性及び頑強性と等価であることが知られている。

一方、攻撃方法には“ 選択平文攻撃(chosen-plaintext attack:CPA) ”、“ 非適応的選択暗号文攻撃(non-adaptive chosen-ciphertext attack:CCA1) ”、“ 適応的選択暗号文攻撃(adaptive chosen-ciphertext

attack:CCA2)”の3つがある。選択平文攻撃とは、公開鍵のみを与えられた状況での攻撃であり、公開鍵を利用して自分で選んだ任意の平文に対する暗号文を入手しながら行う攻撃である。選択平文攻撃は公開鍵暗号系では原理的に避けられない。非適応的選択暗号文攻撃とは、公開鍵が与えられたうえに、解読を試みる暗号文を入手するまでは復号オラクルへのアクセスが許される攻撃である。適応的選択暗号文攻撃とは、公開鍵が与えられたうえに、解読を試みる暗号文を入手する前後において、復号オラクルへのアクセスが許される攻撃である。ただし、解読を試みる暗号文自体の復号を復号オラクルに要求することは禁止されている。

以上のGOAL∈ {OW,IND,NM}とATK∈ {CPA,CCA1,CCA2}の組合せ(9 通り)によって、GOAL-ATK という記号で暗号スキームの安全性が表現される。これは、そのスキームがATK という攻撃に対してGOAL という目標が達成されることを意味するのであって、それが事実であることは、当然のことながら論理的に証明されなければならない。もし、IND-CCA2 (適応的選択暗号文攻撃に対して強秘匿)という性質を証明できれば、それ以外の(8 通りの)すべての安全性が成り立つことが知られている。その意味で、IND-CCA2 は最も強い意味の安全性と位置付けることができる。ただし、一般には“ある性質をもった乱数が使えるというモデル(ランダム関数仮定)において、ある数論的問題が難しいならば(数論問題仮定)、GOAL-ATLである”という形で安全性を証明するのが普通である。ランダム関数仮定としては、「真にランダムな乱数を使える」などの形で表現される。また、数論問題仮定は、「ある問題が難しいならば」という形で表現される。その具体的問題は、暗号スキームによって異なる。

また、仮定が“強い”とか“弱い”などの表現を使うことがあるが、これは基本的には仮定の相対関係の表現である。例えば、真にランダムな理想的な乱数を使えるモデル(ランダムオラクルモデル)という仮定は、疑似ランダムでよいというモデル(疑似ランダム関数モデル)よりは強い仮定である。また数論問題仮定においては、例えば、問題 A が問題 B に帰着するとき($A \leqq B$ のとき)、 A が難しいという仮定よりは、 B が難しいという仮定のほうが、相対的に弱い仮定である。このほか、仮定の強弱を相対関係でなく一般的な定性的表現として使うこともある。例えば、その数論的問題の難しさが広く認知されているような場合や、暗号理論分野においては誰もが成り立つと信じているような予想(一方向性関数の存在や $P \neq NP$ など)を仮定する場合は、定性的表現として弱い仮定と扱うことがある。

以上を総合すれば、暗号スキームの安全性を証明する場合、できるだけ弱い仮定のもとでIND-CCA2の成立を示すことが最も強い結果となる。本項で扱う暗号系のうち守秘の機能をもつものについて、スキームの安全性に関して証明されている事実を、上記の記号で表現すると次のようになる。(括弧内はランダム関数仮定である。)

RSA-OAEP: RSA 暗号化関数が一方向性置換である(逆関数の計算が難しい)という仮定のもとで IND-CCA2 (ランダムオラクルモデル)

EPOC-1: $p-$ 部分群問題が難しいという仮定のもとでIND-CCA2 (ランダムオラクルモデル)

EPOC-2: $n = p^2 q$ の素因数分解が難しいという仮定のもとでIND-CCA2 (ランダムオラクルモデル)

EPOC-3: $n = p^2 q$ 型のGAP-素因数分解問題が難しいという仮定のもとでIND-CCA2 (ランダムオラクルモデル)

HIME-1:「 $n = p^k q$ 型の合成数($1 < k < \min\{p,q\}$)に対する素因数分解が難しいという仮定のもとで

IND-CCA2 (ランダムオラクルモデル)」という主張が自己評価書にあるが、これは確認できない。(OAEP 変換だけではIND-CCA2 の成立に十分ではないことが判明したため)

HIME-2:「$n = p_1 \cdots p_d$ 型の合成数に対する素因数分解が難しいという仮定のもとでIND-CCA2(ランダムオラクルモデル)」という主張が自己評価書にあるが、これは確認できない。(OAEP変換だけではIND-CCA2 の成立に十分ではないことが判明したため)

なお、上記の数論問題仮定はどれも、IntegerFactoring の困難性よりは強い仮定である。また、p－部分群問題やGAP-素因数分解問題の困難性の仮定は、Factorp^{k}q の困難性よりは強い仮定である。しかしながら、上記の数論的問題はどれも、効率的解法が発見されていない。

(b)**署名暗号系**

署名を機能とする暗号系のスキームの安全性に関しては、GOAL とATK の組は次のとおりである。GOALとしては、“一般偽造不可”、“選択的偽造不可”、“存在的偽造不可”の3つが挙げられる。一般偽造不可とは、署名の偽造ができない文書が存在することである。選択的偽造不可とは、ある文書以外に対しては署名の偽造ができないことである。存在的偽造不可とは、どの文書に対しても署名の偽造ができないことである。

またATK としては、“受動攻撃”、“一般選択文書攻撃”、“適応的選択文書攻撃”がある。受動攻撃とは、公開鍵だけを使って偽造を行う攻撃である。一般選択文書攻撃とは、攻撃者が選んだ文書に対して真正な署名者に署名をさせた後に、そこで得た情報に基づいて別の文書の署名を偽造する攻撃である。適応的選択文書攻撃とは、攻撃者が選んだ文書に対して真正な署名者に署名させ、それをもとに適応的に選んだ文書に対してさらに同じことを繰り返し、その結果得た情報をもとに最終的に別の文書の署名を偽造する攻撃である。明らかに、「適応的選択文書攻撃に対して存在的偽造不可」という性質が最も強い安全性となっている。署名を機能とする暗号系のスキームの安全性に関しては、次の事実が示されている。

RSA-PSS:RSA 暗号化関数が一方向性置換であるという仮定のもとで、適応的選択文書攻撃に対して存在的偽造不可(ランダムオラクルモデル)

ESIGN-signature:$n = p^2 q$ 型の法に対するe乗根近似問題(AER)が難しいという仮定のもとで、適応的選択文書攻撃に対して存在的偽造不可(ランダムオラクルモデル)

ACE Sign:強RSA 仮定のもとで、適応的選択文書攻撃に対して存在的偽造不可(擬似ランダム関数モデル)

ACE Sign では真の乱数を用いない実際的なモデルで証明がなされている点で、仮定が弱い。しかし、数論問題仮定は強RSA 仮定である。強RSA 仮定とは、RSA の法nと$y \in \mathbf{Z}_n^*$が与えられたとき、$y \equiv x^r \pmod n$なる$x \in \mathbf{Z}_n^*$と$r > 1$を発見する問題が難しいという仮定であり、RSA 暗号化関数の一方向性置換を仮定するよりは強い仮定である。この強RSA 仮定を含めて、IntegerFactoringが難しいという仮定よりは強い仮定であるが、どの問題についても効率的な解法は発見されていない。

4.3.2 離散対数問題

本章では、基盤となる代数系に素体 $\mathbf{F}_p$ を用い, $\mathbf{F}_p$ 上の離散対数問題の困難さに基づく暗号スキームを、共通の離散対数問題という切り口で評価する。具体的に評価する暗号は以下の通りである。

・守秘　：　ACE Encrypt
・鍵共有　：　DH
・署名　：　DSA

（1） 各種スキームのベースとなる問題

各スキームは、すべて何らかの仮定において、ある問題との安全性の関係が議論できる。本章で述べる方法は，全て離散対数問題（DLP）ベースなので，DLP の効率的な解法が提案されると、すべてのスキームの安全性は崩れる。しかし、その逆は必ずしも成り立たない。ここでは、スキームの安全性を明確に記述するために，スキームのベースとなる問題について定義する。

DLP p を素数とする時，$g \in \mathbf{F}_p$, $y \in \mathbf{F}_p^*$ が与えられたとき、$y = g^x \pmod p$ となる $x \in Z_{p-1}^*$ を見つける問題。

CDH p を素数とする時、$g \in \mathbf{F}_p$, $y_A, y_B \in \mathbf{F}_p^*$ が与えられたとき、$K = g^{ab} \pmod p$ を求める問題。ここで、 $y_A = g^a \pmod p, \mathrm{y}_B = g^b \pmod p$ である。

DDH p を素数とする時，$A = g^a, B = g^b, C \in \mathbf{F}_p^*$ が与えられたとき、$C = g^{ab} \pmod p$ ならば1 を、そうでなければ、0 を出力する問題。

（2）各種関数間の関係

前述した各種の問題を解く難しさについて説明するため，これらの問題を関数として定義し，関数間の帰着概念を用いて関数の複雑さを比較する。

これは一般的にいえば、関数 G を計算するサブルーチン（オラクル）を用いて、関数 F が計算できるとき、関数 F は関数 G に帰着すると言う。以下で述べる帰着は、本章の内容に沿った形式で述べるが、より一般的で厳密な定義については、例えば[10]などに記述されている。

定義 1 (多項式時間Turing 帰着)関数 F, G に関して、もし任意の x に対し $|x|$ の多項式回数以内（ここでは j 回とする）のサブルーチンおよび多項式で計算可能な関数 $h_i, (1 \leqq i \leqq j-1)$ を用いて $F(x) = h_j\left(\circ_{i=0}^{j-1} (G \circ h_i)\right)(x)$ となるならば、 F は G に多項式時間Turing 帰着するといい、$F \leqq_T^{FP} G$ と表記する。

各種スキームの困難さを表す関数DLP、CDH、DDH について、以下の関係が示せる。
定理 1 ([17]) DDH $\leqq_T^{FP}$ CDH $\leqq_T^{FP}$ DLP

（3）各種スキームの安全性

各種スキームの安全性を，守秘系においては適応的選択暗号文攻撃に対して強秘匿であるか，署名系においては適応的選択文書攻撃に対して存在的偽造不可であるかという観点で議論する。このように本来能動的攻撃に対する安全性まで議論するが、non-interactive DH の場合、能動的攻撃が想定できないという観点から公開鍵が正しいという仮定でのimplicit secrecyを満たすといえる。
定理 2 ([6])ハッシュ関数が汎用一方向性関数であり、CDH が困難であるという仮定のもとで、ACE Encrypt は、IND-CCA2である。
ここで，汎用一方向性ハッシュ関数とは、関数 h とその定義域内にある x が与えられたとき $h(x)=h(z)$ となる z を求めることが困難な関数である。
最後にDSA と離散離散対数問題との帰着関係を示す。
定理 3 ([11])ハッシュ関数がランダム関数であり、DLP が困難であるという仮定のもとで，ハッシュ関数の変更を加えたDSA は、適応的選択文書攻撃に対して存在的偽造不可である。
ここで，ランダム関数[2]とは，入力データに対して真にランダムなデータを出力する理想的なハッシュ関数である。

（4）離散対数問題を解くアルゴリズム

本章では、DLP の解法アルゴリズムについて記載する。前セクションで、各スキームはDLP を利用しているが、スキームによっては、安全性がDLP 以下である場合があることを述べた。しかしながら、現実的な解法としてはDLP の解法しかなく、そういう観点では、DLP の解法に対して強力であるようにパラメータを設定すればよい。

離散対数問題のアルゴリズムは、次の2 種類に大別することができる。

1. $H=\langle g\rangle \subseteq G$ としたとき H の位数 ℓ に依存して $\log_g y$ を求めるアルゴリズム。その計算時間は ℓ の最大素因数のサイズの指数時間オーダ $O(\sqrt{\ell})$ の実行時間となる。

2. $G=\mathbf{F}_q^*$ $(q=p^k)$ として、指数計算法とよばれる手法で $\log_g y$ 求めるアルゴリズム。その計算時間は有限体 $\mathbf{F}_q$ のサイズの準指数時間オーダの実行時間となる。

前者に属するものとしては、Shanks [15]、Pohlig-Hellman [12]、Pollard [13]のアルゴリズムなどが知られている。後者に属するものとしては、Adleman [1]、Coppersmith [3]、ElGamal [5]、Pomerance [14]、Coppersmith-Odlyzko-Schroeppel [4]、Gordon [8][9]のアルゴリズムなどが知られている。

(a)位数に依存するアルゴリズム

Shanks のアルゴリズム:

$G \ni g$ とし $H = \langle g \rangle \subseteq G$ の位数を $\ell, m = \left| \ell^{1/2} \right|$, $0 \leqq r, q \leqq m$ とする。

また、H に対して $f : H \to \{1, \ldots, n\}$ なる単射が存在し、$\log \ell$ の低次多項式時間で計算できるとする。

Step 1.次の集合 L_1, L_2 を計算する。

$$L_1 = \left\{ (i, f(yg^i)) \middle| 0 \leqq i \leqq m \right\},$$

$$L_2 = \left\{ (i, f(g^{mi})) \middle| 0 \leqq i \leqq m \right\}$$

Step 2. L_1, L_2 の各元を第2 成分についてソートする。

Step 3. L_1 の元の第2 成分と L_2 の元の第2 成分が一致しているものを探索する。これを

$(r, f(yg^r)) \in L_1, (q, f(g^{mq})) \in L_2$ とおく。

このとき、$yg^r = g^{mq}$ より, $y = g^{mq-r}$ が成り立つ。

故に r, q を求めることで $\log_g y = mq - r$ が得られる。

このアルゴリズムにおける全体の計算時間は $O(\sqrt{\ell})$ となる。さらに $O(\sqrt{\ell})$ 個の元を格納するための記憶領域も必要となる。

Pohlig-Hellman のアルゴリズム:

$H = \langle g \rangle \subseteq G$ の位数を ℓ とする。

Step 1: ℓ が

$$\ell = \prod_{i=1}^{k} p_i^{e_i}, p_1 < \cdots < p_k \ (p_i : \text{異なる素数})$$

と分解されているとする。

Step 2:各 $\mathbf{Z}_{p_i^{e_i}}$ における $\log_g y$ を求める。

Step 3:中国人剰余定理

$$\mathbf{Z}_\ell \ \cong \ \mathbf{Z}_{p_1^{e_1}} \times \cdots \times \mathbf{Z}_{p_k^{e_k}}$$

よりStep 2 で求めた値を合成することで $\log_g y \in \mathbf{Z}_\ell$ を求める。

このアルゴリズムの実行時間は、Step 2 に依存し、$O\left(\sum_{i=1}^{k} e_i \left(\log \ell + \sqrt{p_i}\right)\right)$である。よって ℓ の最大

素因数 p_k のサイズの指数関数オーダとなる。しかし ℓ が $O(\log \ell)$ -スムーズであるとき、 $O(\sqrt{\log \ell})$ を得る。即ち、 $p-1$ が小さな素数の積になっている場合に有効である。

(b)指数計算法

指数計算法は効率よい因数分解アルゴリズムと多くの類似点をもつ。一般に、指数計算法のアルゴリズムは2部構成となっており、Step 1 では、 $H = \langle g \rangle \subseteq G$ から適切に選んだ部分集合(因子基底)の離散対数を求めてデータベースとして蓄積し、Step 2 では、このデータベースを利用して実際に $\log_g y$ を求める。この一般的アルゴリズムを以下に記述する。

指数計算法の一般的アルゴリズム

Step 1: $H = \langle g \rangle \subseteq G, H$ の位数を ℓ とする。因子基底として $\beta = \{p_1, \ldots . p_m\} \subseteq H$ を定め、

$$g^{b_i} = \prod_{j=1}^{m} p_j^{a_{ij}}$$

と分解される b_i を探す。両辺の対数をとれば、

$$b_i \equiv \sum_{j=1}^{m} a_{ij} \log_g p_j (\mathrm{mod}\, \ell)$$

となり、これは $\log_g p_j$ を未知数とする方程式と見ることができる。このような b_i に関して

$\# \{b_i\} \geqq \# \beta$ となったとする。このとき、$\mathbf{Z}_\ell$ 上の行列 $A = (a_{ij})$ の階数が $m(= \# \beta)$ となれば、

$\log_g p_j (1 \leqq j \leqq m)$ に関する線形方程式は一意の解をもち、 β の各元の離散対数がわかる。

Step 2:ランダムに r を選び、 yg^r が以下のように分解されるまで続ける。

$$yg^r = \prod_{j=1}^{m} p_j^{e_j}$$

このとき両辺の対数をとることによって

$$\log_g y = \sum_{j=1}^{m} e_j \log_g p_j - r$$

となり、$\log_g y$ を求めることができる。

Adleman のアルゴリズム:

連分数法による素因数分解のアイデアを離散対数問題に応用したアルゴリズム。 $G = H = \mathbf{Z}_p^*$ の離散対数問題に対して、因子基底 β として $u = L_p[1/2,0]$ 以下の素数の集合を設定するものである。即ち、上のアルゴリズムにおけるStep 1 では b_i をランダムに選び、 g^{b_i} がスムーズになるような b_i のみをふるいにかけることになる。このように β を選ぶことで、 g^{b_i} がスムーズになる確率の評価に若干の仮定が入るがアルゴリズムの実行時間は $L_p[1/2,c](c \approx 1)$ となる。

Pomerance のアルゴリズム:

Adleman のアルゴリズムを改良して,仮定なしの厳密な実行時間を評価したアルゴリズムである。その速度はAdleman のアルゴリズムとほぼ同等で、 $L_p[1/2,\sqrt{2}]$ である。

Gordon のアルゴリズム:

数体ふるい法による素因数分解の概念を応用したアルゴリズムである。一般的数体ふるい法に基づくものと、特殊数体ふるい法に基づくものの2 種類がある。前者の場合は一般の p に対して適用可能であり、実行時間は $L_p[1/3,3^{2/3}] = L_p[1/3,2.08008]$ である。後者の場合はある性質をもつ p に対して適用可能であり、実行時間は $L_p[2/5,1.00475]$ である。ここでいう p の特別な性質とは、以下のような整係数既約モニック多項式 f の選択を可能にするということである。

1. $f \in \mathbf{Z}_p[X]$ の係数は適当に小さい。
2. ある整数 x, y が存在し、その大きさはどちらも $p^{1/k}$ 程度で、 $y^k f(x/y) \equiv 0 (\mathrm{mod}\, p)$ をみたす。
3. $\mathrm{O}_k = \mathbf{Z}[\alpha]$ は一意分解整域。

(c)離散対数解法アルゴリズムのまとめ

上記に述べた現在知られている有効なアルゴリズムとその計算量、アルゴリズムを有効にする条件についてまとめる。

アルゴリズム	計算量	アルゴリズムを有効にする素数の条件
Shanks のアルゴリズム	$O(\sqrt{\ell})$ + 記憶領域 $O(\sqrt{\ell})$	汎用的(全ての素数)
Pohlig-Hellman のアルゴリズム	$O(\sum_{i=1}^{k} e_i(\log \ell + \sqrt{p_i}))$	$p-1$ が小さな素数の積のとき
Adleman のアルゴリズム	$L_p[1/2,c]$	汎用的(全ての素数)
Pomerance のアルゴリズム	$L_p[1/2,\sqrt{2}]$	汎用的(全ての素数)
Gordon のアルゴリズム	$L_p[1/3,3^{2/3}]$	汎用的(全ての素数)

従って、離散対数問題の難しさに安全性の根拠をおいた暗号を使用する際は、上記のアルゴリズムを困難

にするパラメータ(素数 p)を設定する必要がある。

(d)選択した素数におけるDLP の困難さについて

各種スキームは、一般にDLP 等が困難であるという仮定のもとで安全性について議論する。つまり、このような議論においては、具体的にDLP に利用する群の選択、すなわち有限体$\mathbf{F}_p$ に何を用いるかということは議論の対象としない。しかし、現実的な安全性を議論する際、特にDLP ベースのスキームにおいては、全システムで固定した有限体$\mathbf{F}_p$ を用いることが多いので、有限体の選択は重要な課題となる。本節では、提案されているスキームが推奨している有限体上のDLP の安全性について議論する。

3つのスキームのうち推奨有限体を提示しているのは、DSA 及びACE Encrypt である。それぞれ条件は下記の通り。

	素数の条件	サイズ
DSA	$2^{L-1} < p < 2^L$　L は64の倍数	$512 \leqq L \leqq 1024$
ACE Encrypt	なし	1024 bit 以上

DSA は、処理速度の観点から取るべき素数の大きさを、64 ビット 単位になるようにしている。このような素数を用いたDLP について、現時点では有効となる解法アルゴリズムは提案されていない。しかし、DSAのサイズが1024 ビット 以下という限定については、今後の解法アルゴリズム及びコンピュータの性能改良の状況によっては、鍵のサイズを1024 ビット 以上にする必要がある。ACE Encrypt については、素数の取り方に関して現時点では、特段の問題はないと考えられる。

（6）まとめ

各種スキームについて、ベースとなる問題、その仮定についてまとめる。

表4.2.1　DLP ベーススキームのまとめ

機能	守秘	鍵共有	署名
スキーム名	ACE Encrypt	DH	（修正）DSA
安全性の根拠	DDH	CDH	DLP
仮定	汎用一方向性ハッシュ関数	implicit secrecy	ランダムハッシュ関数

参考文献

[1]L.M.Adleman,“A subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography”,Proc.of FOCS,pp.50-60 (1979).

[2]M.Bellare and P.Rogaway,“Random oracles are practical:A paradigm for designing efficient protocols”,1st ACM Conf. on Computer and Communications Security pp.62-73,1993.

[3]D.Coppersmith, “ Fast Evaluation of Logarithms in Fields of Characteristic Two ” ,IEEE Trans. Inform.Theory,IT-30,pp.587-594 (1984).

[4]D.Coppersmith,A.M.Odlyzko and R.Schroeppel, “ Discrete Logarithms in GF (p) ” ,Algorithmica Vol.1,pp.472-492 (1986).

[5] T.ElGamal, “ A subexponential-Time Algorithm for Computing Discrete Logarithms over GF (p 2) ” , IEEE Trans. Inform. Theory IT-31,pp.473-481 (1985).

[6]R.Cramer and V.Shoup, “ A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack ” ,In Advances in Cryptology Crypto ’ 98 pp.13-25,1998.

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

[8]D.M.Gordon, “ Discrete Logarithm in GF (p)Using the Number Field Sieve ” ,to appear in SIAM Journal on Discrete Math

[9]D.M.Gordon, “ Designing and Detecting Trapdoors for Discrete Log Cryptosystems ” ,Proc.of CRYPTO ’ 92,LNCS 740,pp.66-75 (1992).

[10]H.Sakurai and H.Shizuya, “ A Structual Comparison of the Computational Di .culty of Breaking Discrete Log Cryptosystems ” ,In Journal of Cryptology 11:29-43 Springer-Verlag,1998.

[11]D.Pointcheval and J.Stern, “ Security Proofs for Signature Schemes ” ,In Advances in Cryptology – EUROCRYPT ’ 96 pp.387 – 398,1996.

[12]S.Pohlig and M.Hellman, “ An Improved Algorithm for Computing Logarithms over GF (p)and Its Cryptographic Signi .cance ” ,IEEE Trans. Information Theory 24,pp.106-110 (1978).

[13]J.M.Pollard, “ Monte Carlo Methods for Index Computation mod p ” ,Math. Comp .32,pp.918-924 (1978).

[14]C.Pomerance, “ Fast,Rigorous Factorization and Discrete Logarithm Algorithm,Discrete Algo-rithms and Complexity ” ,Proc.of the Japan-U.S.Joint Seminar,Academic Press,pp.119-143 (1987).

[15]D.Shanks, “ Class Number,a Theory of Factorization ” ,and Genera,Proc.Symposium Pure Math-ematics,AMS (1972).

[16]National Institute of Standards and Technology (NIST),FIPS Publication 186:Digital Signature Standard May 19,1994.

[17]H.Woll, “ Reductions among number theoric problems ” ,In Information and Computation vol.72, pp.167 – 179,1987.

4.3.3 楕円曲線離散対数問題

本節では、有限体上定義された楕円曲線上の離散対数問題(ECDLP)の困難さに基づく暗号スキームを評価する。具体的に評価する方式は以下の通りである。

- 守秘: ECAES in SEC1、PSEC-1、PSEC-2、PSEC-3
- 鍵共有: ECDHS in SEC1、ECMQVS in SEC1、HDEF-ECDH、
- 署名: ECDSA in SEC1、MY-ELLTY ECMR-160/192/OEF-h

（1）ECDLP とその関連問題

上記方式の安全性は、基本的に楕円曲線の離散対数問題とその派生問題に基づいている。

ECDLP : $E/\mathbf{F}_q$ を $\mathbf{F}_q$ 上定義された楕円曲線とする時、$G \in E(\mathbf{F}_p)$、$P \in <G>$ が与えられたとき $P = aG$ となる $a \in \mathbf{Z}$ を見つける問題。

EC-CDH(EC-DH)： $G \in E(\mathbf{F}_q)$、$P = aG, Q = bG \in E(\mathbf{F}_q)$ ($a, b \in \mathbf{Z}$) が与えられたとき、$R = abG$ を求める問題。

EC-DDH： $(G,P,Q,R) \in E(\mathbf{F}_q)^4$、$P = aG, Q = bG$ が与えられたとき $R = abG \in E(\mathbf{F}_q)$ ならば 1 を、そうでなければ、0 を出力する問題。

EC-GAP-DH： $(G,P,Q,R) \in E(\mathbf{F}_q)^4$ が与えられたとき、EC-DDH オラクルを仮定して、EC-CDH を解く問題。

これらの問題の困難性には以下のような関係が知られている。

EC-DDH $\leqq_T^{FP}$ EC-CDH $\leqq_T^{FP}$ EC-DL

また、楕円曲線上の離散対数に関する諸問題は、有限体上の離散対数に関する諸問題の拡張と見なすことができるが、特殊な場合を除けば、その相互関係は現在解明されていない。

（2）楕円暗号に対する攻撃法

ECDLP に対して、特殊な楕円曲線に対して、準指数時間もしくは多項式時間の攻撃法が発見されているが、一般的な楕円曲線の離散対数に基づく暗号系に対して、現在指数時間の攻撃法しか知られていない。

(a)Pollard の rho 法

現在、楕円暗号に対する最も強力な攻撃法は、並列 Pollard の rho 法である。

Pollard の rho 法: $P, Q \in E(\mathbf{F}_q)$、$xP = Q$ に対して、離散対数 x を求める。

Step 1 ランダム関数

$$f :< P >\rightarrow< P >$$

を選ぶ。

・$Q_0 := Q \in < P >$ から出発して、collision $Q_{i+1} = f(Q_i)$ を探す。具体的に

$$Q_0 = [a_0]P + [b_0]R \text{、} \quad a_0, b_0 \in \mathbb{Z}$$

から出発して、$Q_i = Q_j$ となるまで、点列

$$Q_{i+1} = f(Q_i) \text{、} \qquad Q_i = [a_i]P + [b_i]R$$

を計算する。

・もし $b_i \neq b_j$ ならば、

$$[a_i - a_j]P + [b_i - b_j]R = O$$

から、

$$R = (b_i - b_j)^{-1}(a_i - a_j)$$

を計算する。

・離散対数

$$\log_P Q = a_0 - (b_i - b_j)^{-1}(a_i - a_j)b_0$$

を出力する。

Pollard の rho 法による ECDLP の計算量は、鍵長の指数関数 $O(\sqrt{p_{\max}})$ である。ここでは、$p_{\max}$ は、楕円曲線の有理点群の位数 $\# E(\mathbf{F}_q)$ の最大素因数とする。特徴としては、Shanks の BSGS 法に比べてメモリはほとんどいらないことと、並列化が容易であることが挙げられる。特に、n 台の CPU によって並列化されたとき、rho 法による ECDLP の計算量は、$O(\sqrt{p_{\max}/n})$ となる。

従って、素位数あるいは位数に大きな素因数を含む楕円曲線の ECDLP は、準指数時間攻撃の知られている DLP に比べてより困難な問題であると考えられている。

現時点の楕円曲線暗号の解読記録としては、素体上では、鍵長 97 ビット、標数 2 の拡大体上では、109 ビットの Koblitz 曲線が解かれている。必要な群計算は、4×10^5 MIPS・YEARS となって、512 ビットの RSA の解読の 50 倍である。

160 ビットと 210 ビット ECDLP に基づく暗号方式のプリミティブの安全性は、1024 ビットと 2048 ビットの IF と DLP のそれに相当すると考えられている。

(b)Weil 対と Tate 対による DLP への変換攻撃

特殊な構造をもつ楕円曲線に対して、ECDLP を、Weil 対(MOV 還元)と Tate 対を用いて定義体（の拡大体）

の乗法群上の DLP に変換する MOV 攻撃と Frey-Ruck 攻撃(FR 攻撃)が提案されている。

楕円曲線 $E(\mathbf{F}_q)$に対して、p =char $\mathbf{F}_q$ として、Weil 対は、gcd(m, p)=1 のとき、m 等分点群 $E(\mathbf{F}_q)_m$ から、1 の m 乗根群 μ_m を含む定義体の拡大体 $\mathbf{F}_{q^k}$ への写像

$$e_m : E(\mathbf{F}_q)_m \times E(\mathbf{F}_q)_m \to \mu_m(\mathbf{F}_{q^k})$$

として定義される。この写像は、Miller の確率的多項式時間アルゴリズムによって計算できる。

また、Tate 対

$$t_m : E(\mathbf{F}_q)_m \times E(\mathbf{F}_q)/mE(\mathbf{F}_q) \to \mu_m(\mathbf{F}_{q^k})$$

による変換攻撃も提案されている。この写像も、Miller のアルゴリズムを拡張した確率的多項式時間アルゴリズムで計算できる。

一般的には、これらの写像を定義するには、m 等分群 $E(\mathbf{F}_q)_m$ が $E(\mathbf{F}_{q^k})$に入ること, 或は、1 の l 乗根 μ は $\mathbf{F}_{q^k}$ に入ることが必要である。従って、$E(\mathbf{F}_q)$上の ECDLP は、$\mathbf{F}_{q^k}$ 上の DLP と変換される。

トレース $t := q+1-\#E(\mathbf{F}_q) \equiv 0 \bmod p$ となる Supersingular の楕円曲線に対して、定義体の拡大次数 k は 6 以下であるため、ECDLP は準指数時間で攻撃できる。

また、$t \equiv 2 \bmod m$ となるトレース 2 の楕円曲線 E では、$E(\mathbf{F}_q)_m$ 上の ECDLP を確率的多項式時間で $\mathbf{F}_q^{\times}$ 上の DLP へ変換できる。従って、トレース 2 の楕円曲線上の ECDLP は、準指数時間攻撃が可能である。しかし、一般的な楕円曲線の部分群を定義体の拡大体の乗法群へ単射するためには、拡大次数 k が $\log q$ の指数関数となるので、準指数時間攻撃は得られない。

(c)p 等分点群に対する多項式時間攻撃

楕円曲線 E は、位数が定義体 $\mathbf{F}_q$ の標数 p =char $\mathbf{F}_q$ のべきである場合、p-divisible 楕円曲線と呼ばれる。このような楕円曲線の有理点群或はその p 等分点群を対数微分を用いて、多項式時間で、有限体上 $\mathbf{F}_q$ の加法群へ帰着する

$$E(\mathbf{F}_q)_p \to \mathbf{F}_q^+$$

攻撃が、Semaev、Smart、Sato-Araki によって提案されている。(SSSA 攻撃)

特に、素体 $\mathbf{F}_p$ 上の p-divisible 楕円曲線は $p = m \Leftrightarrow t \equiv 1 \bmod p$ となるため、トレース 1 の楕円曲線になる。参考までに、定義体上でのトレースが 1 となる楕円曲線 $E/\mathbf{F}_q$ は、anomalous 楕円曲線と呼ばれ、素体上の楕円曲線の場合は、SSSA 攻撃が対象となる楕円曲線は anomalous 楕円曲線となる。

(d)Weil Descent を用いる攻撃

拡大体上定義された楕円曲線上の離散対数問題を超楕円曲線に変換する攻撃法である。適用範囲や計算量などの詳細の解析はなされていないが、大種数の超楕円楕円暗号に対する Adleman-DeMarrais-Huang による攻撃法が、Gaudry によって高速化されているため、種数 4 以上の超楕円曲線へ変換可能の場合は、Pollard の rho 法より速くなる。

(e)自己同型群による攻撃法

楕円曲線は大きな自己同型群を持つとき、その性質を利用して、Pollard の rho 法を高速化することができる。位数 m の自己同型群を持つ場合、rho 法の計算量は $\sqrt{m}$ 分の一となる。一般的な楕円曲線は、自明な自己同型群しか持たないが、拡大体 $\mathbf{F}_{q^k}$ 上の楕円曲線 E はその方程式が部分体 $\mathbf{F}_q$ 上で定義されたとき、その自己同型群の位数 m は、k の倍数となるため、一般的に、このような曲線を使うときには注意を払う必要がある。

（3）選択した楕円曲線における ECDLP の困難性について

各種スキームは、一般に ECDLP 等が困難であるという仮定のもとで安全性について議論する。つまり、このような議論においては、具体的に ECDLP に利用する群の選択、すなわち楕円曲線 $E/\mathbf{F}_q$ に何を用いるかということは議論の対象としない。しかし、現実的な安全性を議論する際、特に ECDLP ベースのスキームにおいては、全システムで固定された楕円曲線 $E/\mathbf{F}_q$ を用いることが多いので、楕円曲線の選択は重要な議論となる。本説では、提案されているスキームが推奨している楕円曲線上の ECDLP の安全性について議論する。

全ての応募暗号方式では、一般的な曲線に対する最も強力な並列 Pollard の rho 法などの攻撃に対し、160 ビット以上のサイズの素因数を持つ位数の楕円曲線を用いているため、安全であると考えられる。また、楕円曲線上の有理点の群を有限体へ移す MOV、FR、SSSA などの攻撃に弱い特殊な曲線を避けている。楕円曲線の選択にあたっては、広い範囲の曲線のクラスからランダムに選ぶという戦略が安全性の観点から取られることが多い。ただし、特殊な曲線のクラスを選択し、そのクラスの利点を活かす場合もある。ECAES in SEC1, ECDSA in SEC1, ECDHS in SEC1, ECMQVS in SEC1 では推奨されている曲線の中に高速処理に適した Koblitz 曲線を含めている。Koblitz 曲線は，$E/\mathbf{F}_2$ 上のトレース 1 の楕円曲線、すなわち、

anomalous 楕円曲線を拡大体 $\mathbf{F}_{2^r}$ に持ち上げた曲線であり、anomalous binary 楕円曲線とも呼ばれる。また、HDEF-ECDH では、特徴のあるクラスの曲線（トレース 3 の楕円曲線 $E/\mathbf{F}_q$）を使用している。

(a)スキームの安全性

・守秘

ECDLP に基づく守秘方式は、ともに能動的な適応的選択暗号文攻撃に対する安全性対策を講じている。

表 4.2.2　ECDLP に基づく守秘スキームの IND-CCA2 に対する安全性

スキーム名	ECAES in SEC1	PSEC-1	PSEC-2	PESC-3
安全性の根拠	楕円曲線上の適応的ハッシュ DH 問題	楕円曲線上の部分決定 DH 問題	楕円曲線上の DH 問題	楕円曲線上の GAP DH 問題
仮定	共通鍵と MAC	ランダムオラクル	ランダムオラクル	ランダムオラクル

特に、ECAES in SEC1 は、ある楕円曲線上の適応的ハッシュ DH 問題の変形となる仮定の元で、適応的選択暗号文攻撃への安全性の証明を示している。ランダムオラクルモデルによって、楕円曲線上の GAP DH 問題への帰着の可能性も指摘されている。

・鍵共有

鍵共有方式は、受動的な攻撃に対して安全と思われるが、能動的な攻撃に対する検討が必要である。

ECDHS と ECMQVS in SEC1 については、署名と組み合わせることで能動的な攻撃に対する安全性と forward secrecy が得られることが指摘されている。

表 4.2.3　ECDLP に基づく 鍵共有スキームの受動的攻撃に対する安全性

スキーム名	ECDHS in SEC1	ECMQVS in SEC1	HDEF-ECDH
安全性の根拠	楕円曲線上の DH 問題	楕円曲線上の DH 問題	楕円曲線上の DH 問題

・署名

ECDSA in SEC1 については、受動的な攻撃に対して安全と思われるが、能動的な攻撃に対する検討は必要である。

MY-ELLTY ECMR-160/192/OEP-h については、80 ビットハッシュ関数によるランダムオラクルの実現の問題点が指摘されている。

表 4.2.4　ECDLP に基づく署名スキームの受動的攻撃に対する安全性

スキーム名	ECDSA in SEC1	MY-ELLTY ECMR-160/192/OEP-h
安全性の根拠	ECDLP	ECDLP
仮定	ランダムオラクル	ランダムオラクル

4.4 個別暗号評価

4.4.1 ACE Sign

1 暗号技術

1.1 技術概要

ACE Sign は1999 年にRonald Cramer とVictor Shoup により1999 ACM Conference on Computer and Communication Security に提案された署名方式の特殊な変形であり、2000年にIBM Zürich 研究所のThomas Schweinberger とVictor Shoup によりmanuscript として発表され、日本IBMから提案された。ACE Sign は($pq:(p-1)/2,(q-1)/2$ も素数型の) 素因数分解(IF) の困難性に基づき、署名を実現する公開鍵方式である。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

ACE Sign に関する知的所有権は、IBM が所有している。

Ronald Cramer, Victor Shoup,“Practical non-malleable public-key cryptosystem" Filed February 16, 1999。

(ISO に)採用された場合には, 非差別的、かつ、適正な対価条件でライセンス提供することが宣言されている。

1.2 技術仕様

署名生成、署名検証関数の概要を示す。

［秘密鍵］

二つの素数 p,q (但し、 $p'=\frac{(p-1)}{2},q'=\frac{(q-1)}{2}$ も素数)

［公開鍵］

$n(=pq)$

e' :素数 $|e|=160$

$a,b:\in QR_n$

［ハッシュ関数］

$H_1:\{0,1\}^* \rightarrow \{0,1\}^{160}$

$H_{2:}:\{0,1\}^* \rightarrow \{0,1\}^{160}$

【署名生成】

以下のようにして、平文 m に対する署名 s を得る(特に、指示のない限り演算は法 n の元で行なう)。

1. k :乱数
2. $h_1 \leftarrow H_1(k,m)$
3. $y' \in QR_n$
4. $x' \leftarrow (y')^{e'} a^{h_1}$
5. e :証拠付素数、$e \neq e', |e| = 160$
6. $h_2 \leftarrow H_2(k,x')$
7. $d \leftarrow 1/e \bmod p'q'$
8. $y \leftarrow (x/a^r)^d$
9. $s \leftarrow e \| y \| y' \| k$

但し、e はMARS の累積/ カウンタ・モードを利用して生成され、それに用いられたパラメータを証拠として付与される。

【署名検証】

以下のようにして、平文 m に対する署名 s を検証する(特に、指示のない限り演算は法 n の元で行なう) 。

1. $e \| y \| y' \| k \leftarrow s$
2. e が証拠付素数であり、$e \neq e'$ であるかを検証する。条件を満たさない場合は、" reject"を出力して終了する。
3. $x' \leftarrow (y')^{e'} h^{H_1(k,m)}$
4. $b = y^e a^{H_2(k,x')}$ であるかを検証する。条件を満たす場合は、" accept"を出力して終了し、そうでなければ、" reject"を出力して終了する。但し、e はMARS の累積/ カウンタ・モードを利用して生成され、それに用いられたパラメータを証拠として付与されている。

この暗号は、セキュリティと効率の良好なバランスを達成するように設計されている。デジタル署名が使用されて、この暗号方式のパフォーマンスが許容される、高度の安全性を要するアプリケーションに適する。

この暗号方式は、Cramer と Shoup によって最近発見された方式に基づいている。この方式は, 適応的選択文書攻撃に対して、存在的安全(存在的偽造は不可能) であることが、Strong RSA Assumption に基づいて証明できる。この証明ではランダムオラクルモデルに論拠をおかない。しかし、ランダムオラクルモデルに論拠をおくと普通の RSA Assumption に基づいて安全性を証明できる。

2. 評価結果

2.1 安全性評価

(a) プリミティブの安全性

現時点で有効な攻撃法は確認されていない。

プリミティブの安全性は、法nの元でのe乗根求解の困難性に基づく、特殊な素因数を用いているため、これのe乗根求解の困難性は未知である。利用されるハッシュ関数が汎用一方向性(より正確には第二プレイメージ衝突耐性)を満たし、また、共通鍵暗号は累積/ カウンタ・モードで使用した場合に擬似ランダム性を有することが必要である。

署名生成の際に利用される乱数は十分にランダムであることが求められる。

(b) パラメータの安全性

pq ($\frac{(p-1)}{2}, \frac{(q-1)}{2}$ も素数)型の素因数分解の困難性に基づく。特殊な素因数を用いているため、これの素因数分解の困難性は未知である。$p+1$法(素因数分解法の一つ) に対しても耐性を有するように、鍵生成の仕様を変更すべきである。

(c) スキームの安全性

Standard Model (実用に近い状況下) で安全性を証明しており、強 RSA 問題、汎用ハッシュ関数に関する仮定(SHA-1 第二プレイメージ衝突耐性)、共通鍵暗号の擬似ランダム性に関する仮定(MARS 累積/ カウンタ・モード疑似乱数性) に帰着されている(但し，強 RSA 問題は，RSA 問題や素因数分解問題よりも特殊な仮定である点に注意すること)。

本スキームでは、ハッシュ関数 SHA-1 から汎用一方向性ハッシュ関数を構成しており、共通鍵暗号 MARS を累積/ カウンタ・モードで利用することで、生成される素数に対する証拠を与えている。そのため、SHA-1 の第二プレイメージ衝突耐性に関する仮定と MARS の累積/ カウンタ・モード疑似乱数性に関する仮定が必要となる。

本スキームの安全性をランダムオラクルモデル で評価した場合には、強 RSA 問題に依存することになる。

2.2 SW 実装評価

ACE Sign に関しては、応募された暗号技術ではあったが、応募書類提出後に仕様が変更されこともあり、応募された技術に対応したソフトウェア実装評価を行わなかった。

2.3 その他

本スキームは、非常に巧妙に構築されているために、Flexibility がない。例えば、ハッシュ関数を単純な SHA-1 に置き換えた場合には、スキームの安全性に問題が生じる。また、共通鍵暗号プリミティブ (MARS) を変更した場合に、生成される素数に対して証拠を与えることができるかは不明である。

4.4.2 ESIGN-signature

1. 暗号技術

1.1 技術概要

ESIGN-signatureはe乗根近似関数に基づく、(公開鍵暗号を用いた)電子署名方式である。NTTにより提案され、基本ESIGN署名関数[OS][O]に基づき、補助関数としてハッシュ関数を利用した暗号スキーム[IFM]を用いて変換したものである。

[OS] Okamoto,T.and Shiraishi,A.:A Fast Signature Scheme Based on Quadratic Inequalities, Proc.of the ACM Symposium on Security and Privacy,ACM Press (1985).

[O] Okamoto,T.:A Fast Signature Scheme Based on Congruential Polynomial Operations,IEEE Trans.on Inform.Theory,IT-36,1,pp.47-53 (1990).

[OFM] Okamoto,T.,Fujisaki,E.and Morita,H.:TSH-ESIGN:Efficient Digital Signature Scheme Using Trisection Size Hash,submission to P1363a (1998).

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・以下の、2件の日本国内登録特許、および1件のUSP/CANADA/EPがある。

(1) 登録番号 1875643 出願番号 60-42052: 署名文書通信方式

(2) 登録番号 1708995 出願番号 59-052696: 署名文書通信方式

(3) US 4625076 Canada 255784 EP 0157258:Signed Document Transmission System

非排他的かつ妥当な条件で他者に実施許諾するとしている。

(関連特許)

・応募者によると, 関連する他社特許の訴求はないものとしている。

応募暗号技術仕様の公開Webアドレス http://info.isl.ntt.co.jp/

1.2 技術仕様

【鍵生成】

入力:セキュリティパラメータ k

出力:公開鍵 $(n,e,HID,pLEN)$ と秘密鍵 (p,q)

(1)2つの素数 p,q を生成し、$n=p^2q$ を計算する。

(2)整数 $e>4$ を選択する。

(3) $pLEN=k$ を決定する。

なお、使用するハッシュ関数のIDを HID とする。

【署名生成】

入力:メッセージ m 、公開鍵 $(n, e, HID, pLEN)$、秘密鍵 (p, q)

出力:署名 s

(1)ランダムに r を選ぶ。

(2) $z = (0 \| H(m) \| 0), \quad \alpha = (z - r^e) \bmod n$

(3) $w0 = \left\lceil \frac{\alpha}{pq} \right\rceil, w1 = w0 \cdot pq - \alpha$

(4) $t = \frac{w0}{e^{r^{e-1}}} \bmod p, s = (r + tpq) \bmod n$

(5) m の署名として s を出力する。

【署名検証】

入力:署名 s 、メッセージ m 、公開鍵 $(n, e, HID, pLEN)$

出力:検証結果

(1) $s^e \bmod n = 0 \| H(m)$ が満たされたら正当、さもなければ不正を出力する。

1.3 その他

ISO/IEC 14888-3 (Digital Signature Algorithms with Appendix)annex b に記載されている。

IEEE P1363a/D4 (Draft Version 4),May 22,2000 に記載されている。

2. 評価結果

2.1 安全性評価

(a) プリミティブの安全性

＜e 乗根近似問題＞

この問題に対してはあまり多くの研究が行われていないが、今のところ $e \geq 5$ であれば安全性の上で特に問題はないと考えられている。なお、ESIGN として最初に提案された署名方式は e＝２を採用しており、弱点が指摘された後に e の値が大きくされた経緯がある。今後長期にわたって使用するために、どの程度セキュリティマージンをとればよいのかは必ずしも明らかとなっていない。たとえば、1024 ビットの教科書的 RSA 署名方式と同程度の安全性を達成するためのパラメータをどう設定すればよいか、明らかにすることが望ましい。この点で、

・ 提案者の仕様書中の条件: $e \geq 5$

・ 提案者の推奨パラメータ: $e \geq 8, |p| = |q| \geq 320, |n| \geq 960$

・ ソフトウェア実装評価時の提案者による採用パラメータ: $e = 2^{10}, |p| = |q| = 384, |n| = 1152$

・ 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案の条件:

$$e \geq 8, |n| \geq 1024$$

のように各所であげられた条件やパラメータ設定例が異なるため、使用によっては十分な吟味が必要である。

なお、B.Vallee らによる攻撃の拡張に対抗するため、$e > 2^{30}$ のように余裕をもって設定すべきであるという意見もある。

＜素因数分解＞

法である合成数の素因数分解の形が RSA 署名方式（教科書的 RSA 署名方式および RSA-PSS）と異なるため、法が RSA 署名方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある（4.3.1 参照)。すなわち、$n = p^2 q$ の素因数分解が $n = pq$ より簡単かどうかについては分かっていない。

(b) スキームの安全性

e 乗根近似仮定が正しいという条件でランダムオラクルモデルにおいて、適応的選択文書攻撃に対して存在的に偽造不可であるとの、応募者による証明に無理はないものと考えられる。スキームの解読成功確率は、e 乗根近似問題の解読成功確率の q_H 倍（ランダムオラクルへの質問回数分）であり、多少の差があるとの意見もある。

2.2 ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

・ 合成数の長さ 1,024 ビットの RSA 暗号相当の強度を持たせるため､ESIGN の合成数の長さは、384 ビット×３＝1,152 ビット。

・ ESIGN のセキュリティパラメータは、10 ビット、２の１０乗(1,024)である。

（２）パラメータの設定方式

・ 乱数の種をファイルとして読み込む。

（３）使用パラメータの検証状況

・ GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。本関数で用いる素数判定アルゴリズムは Miller-Rabin 法である。第 2 引数 reps の値が大きいほど誤判定の確率を低めることができるが、それだけ多くの時間を要し、現実的には 5 から 10 くらいが適当とされている。本実装では reps の値を 5 とした。

［実装上で使用した手法］

・ フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。

・ 仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別な

ことはしていない。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
ESIGN Signature	素因数分解問題	合成数の長さは、384bit × 3 = 1,152bit	ESIGN のセキュリティパラメータは、10 ビット、２の１０乗(1,024)である仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

［システムパラメータ生成］

- システム生成は、事前に済ませており、今回の評価では実施しなかった。

［鍵対生成］

- 素数判定条件：GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。
- 乱数生成方式：まず GNU MP 3.1.1 の関数 gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムは GNU MP のデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数 gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数 mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
ESIGN Signature	50.8ms	素数生成、乱数生成は含まれていない。合成数の長さは、384 ビット×３＝1,152 ビット。
	452.8ms	乱数生成、素数生成を含めた測定値。

［署名生成］

- パディング：不要
- ハッシュ関数：SHA-1 を使用.
- 測定結果：

［署名生成の速度］

測定対象	平均実行時間	補助関数	備考
ESIGN Signature	9.2ms	SHA-1	署名対象データのサイズ：31KＢ
	49.1ms		署名対象データのサイズ：178KB

［署名検証］

（１）検証確認ルーチンの有無

- 実装されている。

（２） 測定結果

［署名検証の速度］

測定対象	平均実行時間	備考
ESIGN Signature	6.6ms	署名対象データのサイズ：31KＢ
	44.3ms	署名対象データのサイズ：178KB

［コードサイズ］

評価対象	ソースコードサイズ	備考
ESIGN Signature	178,762Bytes	Ｃ言語（Intel C/ C++） 多倍長演算ルーチン GMP の一部を含んでいる。

4.4.3 RSA-PSS

1. 暗号技術

1.1. 技術概要

RSA-PSS は 1977 年に Ronald L. Rivest, Adi Shamir, Leonard M. Adleman により提案された RSA 署名と 1996 年に Mihir Bellare, Phillip Rogaway により提案された *Probabilistic Signature Scheme* (PSS) を組み合わせた素因数分解問題(IF)に基づく署名手法である。

＜暗号の発表年、提案論文＞

(1) RSA 発表年:1977 年

R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2), pp. 120-126, February 1978.

(2)PSS 発表年:1996 年

M. Bellare and P. Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In Advances in Cryptology-Eurocrypt '96, pp. 399-416, Springer-Verlag, 1996.

知的財産権 (応募者資料による)

(提案者特許とその扱い)

米国カリフォルニア州立大学は、PSS 署名に関してペンディングされている特許を所有していると主張している。米国カリフォルニア州立大学は同時に IEEE P1363 に対しレターを提出しており、その中で、「もし PSS 署名が IEEE 標準に採用された場合には添付型署名の実現のための技術として、無償でこの PSS を実現・利用することを認める」と述べている。

商用あるいは非商用の製品での利用に関しては RSA Security 社の文書による許可が必要。

なお、ここで言う利用とは、日本における登録商標の利用を含む。

“RSA”は日本において RSA Security Inc.の登録商標である。

応募暗号技術仕様の公開 Web アドレス http://www.rsasecurity.com/rsalabs/rsa_algorithm/

1.2. 技術仕様

【鍵生成】

[出力] 公開鍵 (e, n)、 秘密鍵 d。

1. 素数 p,q を生成。

2. n=pq, λ(n)=LCM(p-1,q-1)を計算する。

3. $e \in Z_{\lambda(n)}$ (GCD(e,λ(n))=1)を適当に定める。

4. d = 1/e mod λ(n)を計算する。

5. 公開鍵(e, n)、 秘密鍵 d を出力する。

システムで次の 3 種類のランダム関数が定められているとする。

H_1:{0,1}* → {0,1}^{k1}

H_2:{0,1}^{k1} → {0,1}^{k0}

H_3:{0,1}^{k1} → {0,1}$^{k-k0-k1-1}$

【署名生成】

[入力] メッセージ m∈{0,1}^{n} , 秘密鍵 d。

[出力] 署名 s.

1. 乱数 r∈{0,1}^{k0} を生成する。

2. w=H_1(m||r)とする。

3. R=H_2(w) ⊕ r とする。

4. y= 0 || w || R|| H_3(w) とする。

5. s=y^d mod n を出力する。

【署名検証】

[入力] 署名 s、 メッセージ m、 公開鍵 (n, e)。

[出力] 1 (検証成功), 0 (検証失敗)。

1. y=s^e mod n とする。

2. y を b || w || R ||γ に分割する。

3. r=R⊕ H_2(w)とする。

4. H_1(m||r) = w, H_3(w) =γ , b = 0 がすべて成立するか検証する。

5. 成立すれば 1 を出力、そうでない場合 0 を出力。

1.3. その他

IEEE P1363[１]、PKCS #1 V2.1[２]、ISO/IEC 9796-2. などに記載されている。

[1] IEEE P1363a: Standard Specifications for Public Key Cryptography: Additional Techniques. Draft D4, May 22, 2000. Available from http://grouper.ieee.org/groups/1363/.

[2] RSA Laboratories. PKCS #1 v2.1: RSA Cryptography Standard. Draft 1, September 17, 1999. Available from http://www.rsasecurity.com/rsalabs/pkcs/.

[3] ISO/IEC 9796-2. Information Technology - Security Techniques - Digital Signature Schemes Giving Message Recovery - Part 2: Mechanisms Using a Hash Function. Working draft, July 2000.

2. 評価結果

2.1. 安全性評価

RSA 暗号化関数の一方向性およびランダムオラクルモデルを仮定することにより適応的選択文書攻撃に対して潜在的偽造不可能であることが証明されている。 RSA-PSS の安全性は、RSA 自体の安全性、並びに、素因数分解問題の困難性に大きく依存するため、これらの問題を解くアルゴリズム、及び、計算機能力には、今後も十分に注意を払っていく必要がある。 なお、教科書的 RSA 署名に対して、証明可能安全性はないので使用に際しては注意を要する。

2.2. ソフトウェア実装評価

RSA PSS に関しては、その他評価が必要と判断した暗号技術であり、多くの実装実績を有しており、今回のソフトウェア実装評価の必要がないと判断した。

4.4.4 DSA

1. 暗号技術

1.1 技術概要

DSA は離散対数問題が困難であるという仮定に基づく署名方式である。

National Institute of Standards and Technology (NIST). FIPS Publication 186-2:Digital SignatureStandard

知的財産権

(提案者特許とその扱い)

DSA 署名特許（U. S. Patent Number5,231,668（出願日 1991 年 7 月 26 日、発効日 1993 年 7 月 27 日）

1.2 技術仕様

1)公開鍵を生成

p：$2^{L-1} < p < 2^{L}, 512 \le L \le 1024$, Lは64の倍数,となるような素数pを生成する。

q : $p-1$の素因子, $2^{159} < q < 2^{160}$

$g = h^{(p-1)/q} \bmod p$　ただし、$1 < h < p-1$、$h^{(p-1)/q} \bmod p > 1$

(gの$\bmod p$での位数は、qとなる)

x：$0 < x < q$　なる乱数をとり、秘密鍵とする。

$y = g^{x} \bmod p$　：公開鍵とする。

2)署名処理

k　：$0 < k < q$　なる乱数をとる。k　は署名毎に変えなければならない。

$r = (g^{k} \bmod p) \bmod q$

$s = (k^{-1}(SHA-1(m) + xr)) \bmod q$

(r, s)　を署名とする。

3)署名検証処理

$w = (s)^{-1} \bmod q$

$u1 = ((SHA-1(m))w) \bmod q$

$u2 = (rw) \bmod q$

$v = ((g^{u1} y^{u2}) \bmod p) \bmod q$

$v = r$　ならば検証成功。

2. 評価結果

2.1 安全性評価

a)前節で述べたスキームは、安全性の証明はできていないが、$SHA-1(m)$ を $SHA-1(m,r)$ に置き換え、$SHA-1$ がランダムオラクルモデルを実現しており、かつ離散対数問題が困難であると仮定すると、適応的選択文書攻撃に対して存在的偽造不可であることが証明される。

b)パラメータは、離散対数問題が、各種の既存攻撃(Shanks, Pohlig-Hellman, Adleman, Pomerance, Gordon など)に対して、弱くならないよう注意して選択する必要がある。

c)素数 p は、$2^{L-1} < p < 2^L, 512 \le L \le 1024, L$は64の倍数, という条件を満たすものとされている。512 ビット程度では、安全性に問題が生ずる恐れがある。さらに 1025 ビット以上にできないため、達成しうる安全性に上限があることも注意を払う必要がある。

d)乱数生成方法の例として FIPS186-2Appendix3 に記述されている方法は最近その有効性については疑問が提示されており、更なる検討が必要である。

e)DSA は、広く標準として使用され実績もあることから、現時点で、安全性に大きな脅威を与えるような問題はないと思われる。しかしながら、将来的には、上記にあげたような要因を考慮にいれた上で注意して使用する必要がある。

2.2 ソフトウェア実装評価

DSA に関しては、その他評価が必要と判断した暗号技術であり、多くの実装実績を有しており、今回のソフトウェア実装評価の必要がないと判断した。

4.4.5 ECDSA in SEC1

1. 暗号技術

1. 1 技術概説

1999 年に、SECG(Standards for Efficient Cryptography Group)によって策定された公開鍵暗号技術であり、楕円曲線を用いた署名方式である。

参考 http://www.secg.org/

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・ 4,745,568: Computational method and apparatus for finite field multiplication, issued May 17, 1988. This patent includes methods for efficient implementation of finite field arithmetic using a normal basis representation.

・ 5,761,305: Key Agreement and Transport Protocol with Implicit Signatures, issued June 2, 1998. This patent includes versions of the MQV protocols.

・ 5,787,028: Multiple Bit Multiplier, issued July 28, 1998.

・ 5,889,865: Key Agreement and Transport Protocol with Implicit Signatures, issued March 30, 1999. This patent includes versions of the MQV protocols.

・ 5,896,455: Key Agreement and Transport Protocol with Implicit Signatures, issued April 20, 1999. This patent includes versions of the MQV protocols.

上記提案者所有の特許および著作権は、提案者により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される。

なお、詳細は、特許所有者、及び SECG のウェブサイト（www.secg.org）を参照のこと。

http://www.secg.org/patent_policy.htm

http://www.secg.org/collateral/certicom_secg_patent.pdf

特許所有者が、特許技術の使用許諾を求める申請者に対し、妥当かつ差別待遇のない取引条件で、使用を許諾することに同意する文書を、SECG に提出済みかどうかを確認すること。

応募暗号技術仕様の公開 Web アドレス http://www.labs.fujitsu.com/theme/crypto/public_key.html
http://www.secg.org/drafts.htm

1.2 技術仕様

署名生成、署名検証関数の概要を示す。

楕円曲線パラメータ (p, a, b, G, l)

素体 F_p 上の楕円曲線 $y^2 = x^3+ax+b$ に素数 l を位数としてもつ有理点 G があることを示す。

楕円曲線パラメータ (k, f, a, b, G, l)

既約式 f で定義される標数 2 の k 次拡大体 F_2^k 上の楕円曲線 $y^2+xy = x^3+ax^2+b$ に素数 l を位数としてもつ有理点 G があることを示す。

公開鍵

楕円曲線パラメータ(p,a,b,G,l)または(k,f,a,b,G,l), 点 G の整数倍である点 U

秘密鍵

U=d・G となる整数 d

【署名生成】

入力: 平文 m

出力: 署名文 (s_1, s_2)

1. r ← 1 以上 l-1 以下の乱数, R ← r・G
2. s_1 ← x(R) mod n
3. e ← Hash(m)
4. s_2 ← r^{-1} (e + s_1 d) mod n

【署名検証】

入力: 平文 m、署名文 (s_1, s_2)

出力:“valid”または“invalid”

1. s_1 および s_2が 1 以上 l-1 以下であることを確認。
2. e ← Hash(m)
3. u_1 ← e s_2^{-1}, u_2 ← s_1 s_2^{-1}
4. R ← u_1 G + u_2 U, v ← x(R)
5. v=s_1 ならば“valid”をそうでないなら“invalid”を出力。

ただし、上で、楕円曲線上の点 P に対して、x(P)は P のｘ座標を表す。

1.3 その他

ECDSA in SEC1 は IEEE P1363 や American National Standard X9.62 に記載されている。

2. 評価結果

2.1 安全性評価

2.1.1 暗号プリミティブの安全性

ECDSA in SEC1 はその基本的な安全性を楕円曲線上の離散対数問題によっている。

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、ECDSA in SEC1 においては、いろいろなビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC2 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からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC2 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。

2.2　スキームの安全性

署名スキームに対しては、何らかの妥当な仮定のもとで能動的な攻撃者による適応的選択文書攻撃に対して存在的偽造不可能性が証明されていることが望ましい。ECDSA in SEC1 に対しては、これまでのところ通常の仮定に基づく、そのような証明は与えられていない。ただし、攻撃者が群構造を全く用いないという仮定をおけば、存在的偽造不可能性が証明されるとの主張がある論文によりなされている。しかし、この主張の妥当性と証明の方法の現実的効果とについて、本評価では結論に得るにいたっていない。

ECDSA in SEC1 は American National Standard X9.62 などに記載されるなど多くの実績があり、署名スキームとしての安全性の上で特段の問題はないと考えられる。

注

ECDSA in SEC1 が DSKS(Duplicate-Signature Key Selection)特性（ある平文と署名文の組が他の公開鍵と秘密鍵の組に対しても正当な平文と署名文の組になってしまう性質）をもたないようにするには、楕円曲線パラメータは全ユーザー間で固定したほうがよい。

2.2　ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

- 以下の 2 種類の曲線を使用。
  - secp160r1 ： 160 ビット素体楕円曲線パラメータ
  - sect163r2 ： 163 ビット標数 2 の体楕円曲線パラメータ
- どちらも SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。

（２）パラメータの設定方式

- 楕円曲線パラメータをファイルから読み込んで使用。
- パラメータには依存しない実装を行っている。

（３）使用パラメータの検証状況

- 楕円曲線の２つの係数が、(コントロールすることのできない)SHA-1 の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。
- X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。
- 各種特殊攻撃を適用できないことも確かめられている。

（4）パラメータサイズのバリエーション

・ 固定パラメータだけではなく、任意の楕円曲線パラメータの利用可能。

・ 素体楕円曲線パラメータで扱えるビット長は以下の通り。

112,128,160,192,224,256,384,521

・ 標数 2 の体楕円曲線パラメータで扱えるビット長は以下の通り。

113,131,163,193,233,239,283,409,571

［実装上で使用した手法］

・ ライブラリは、a=0，a=-3 などの特殊パラメータに限定した高速手法は使用していない

・ 任意のパラメータに適用できる高速手法を使用。

・ パラメータには依存しない実装を行っている。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
ECDSA in SEC1 160bit 素体	ECDLP	RSA1024bit 相当	・楕円曲線の２つの係数が、(コントロールすることのできない)SHA-1 の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。 ・X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。 ・各種特殊攻撃を適用できないことも確かめられている。
ECDSA in SEC1 163bit 標数 2 の体			

［システムパラメータ生成］

・ 楕円曲線パラメータ生成処理は、確率的アルゴリズムであるため、正確な予想処理時間を出すことは困難。

測定結果

測定対象	平均実行時間	備考
ECDSA in SEC1 160bit 素体	6.0 min	18 分間に 3 つの楕円曲線パラメータを生成

［初期化/終了処理］

測定結果

測定対象		平均実行時間	備考
ECDSA in SEC1 160bit 素体	PC/PF	0.004 ms	なし
	PI	13.525 ms	なし
	PV	60.438 ms	なし
	WC/WF	0.032 ms	なし
ECDSA in SEC1 163bit 標数 2 の体	PC/PF	0.004 ms	なし
	PI	15.283 ms	なし
	PV	43.917 ms	なし
	WC/WF	0.032 ms	なし

PC/PF ：楕円曲線パラメータ領域の獲得/解放処理。

PI ：ファイルから読んだパラメータを領域に設定。

PV　：パラメータ検証処理。パラメータが与えられた最初の時にのみ必要。

生成元が曲線上の点か、点を位数倍すると無限遠点となるかなど、正当性をチェック。

WC/WF ：作業領域獲得/解放処理。

［鍵対生成］

（1）素数判定条件

・　方式上、不要。

（2）乱数生成方式

富士通オリジナル擬似乱数生成アルゴリズム。DES/SHA-1 などを使用。

（3）測定結果〔鍵対生成の速度〕

測定対象	平均実行時間	備考
ECDSA in SEC1 160bit 素体	1.9ms	鍵対生成
	5.8ms	鍵対の検証。但し、鍵対の検証処理は、自分で鍵を生成したときには不必要。
ECDSA in SEC1 163bit 標数 2 の体	3.2ms	鍵対生成
	8.0ms	鍵対の検証。但し、鍵対の検証処理は、自分で鍵を生成したときには不必要。

・　鍵対の検証処理は、自分で鍵を生成したときには不必要。

［署名生成］

（1）パディング

・　使用せず。

（2）ハッシュ関数

・　SHA-1 を使用。

（3）測定結果［署名生成の速度］

測定対象	平均実行時間	補助関数	備考
ECDSA in SEC1 160bit 素体	3.7ms	SHA-1	署名対象データのサイズ：31 KB
	11.1ms		署名対象データのサイズ：178 KB
ECDSA in SEC1 163bit　標数 2 の体	5.0ms		署名対象データのサイズ：31 KB
	13.1ms		署名対象データのサイズ：178 KB

［署名検証］

（1）検証確認ルーチンの有無

・　検証に失敗したときは、エラーメッセージを出力。

（2）測定結果［署名検証の速度］

測定対象	平均実行時間	備考
ECDSA in SEC1 160bit 素体	9.7ms	署名対象データのサイズ：31 KB
	17.2ms	署名対象データのサイズ：178 KB
ECDSA in SEC1 163bit　標数 2 の体	13.6ms	署名対象データのサイズ：31 KB
	21.4ms	署名対象データのサイズ：178 KB

［コードサイズ］

測定対象	コードサイズ	備考
ECDSA in SEC1	356,352byte	性能測定用実行形式テストプログラム

4.4.6 MY-ELLTY ECMR-160/192/OEF-h

１．暗号技術

1.1 技術概要

MY-ELLTY ECMR-160/192/OEF-h は、1996 年に SCIS ’96 において松下電器産業株式会社の宮地充子(当時)により提案された暗号技術であり、ECDLP を安全性根拠とする署名スキームである。

MY-ELLTY ECMR-160/192/OEF-h は、各々異なる 3 件の応募であったが、使用する体が異なるだけで、署名スキームとしては同一のものと見なされるため、3 件まとめて記述する。本応募の暗合技術に関して取得あるいは出願中の特許、著作権などの知的所有権に関する状況を説明する。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

[1]特開 H06-110386(特願 H04-257800)、“ 楕円曲線を用いた書名、認証及び秘密通信方式 ”、ベースポイントの絶対値を小さいものをとる

[2]特開 H06-295154(特願 H05-082978)、“ 楕円曲線を用いた署名、認証及び秘密通信方式 ”、$p=2^n-\alpha$なる素数とするときの GF(p)上定義される楕円曲線を用いる

[3]特開 H09-034357(特願 H07-178483)、 “ 署名方式 ”、本提案方法の楕円曲線メッセージ回復型署名

[4]特開 H09-160492(特願 H07-324908)、“ 署名方式 ”、本提案方法の楕円曲線メッセージ回復型署名

[5]特開 H11-316542(特願 H11-056592)、“ 楕円曲線変換装置 ”、a=-3 の楕円曲線

[6]特開 H11-102158(特願 H10-200725)、“ 楕円曲線演算装置 ”、楕円曲線上の点のスカラ倍算の方法

(関連特許)

「本応募技術」に直接関わる他社の特許権および著作権は認められない。

応募暗号技術仕様の公開 Web アドレス http://www.panasonic.co.jp/mdc/crypt/

1.2 技術仕様

ここでは、MY-ELLTY ECMR-160-h の技術仕様を記述する。MY-ELLTY ECMR-192/OEF-h については、使用する体が異なるだけで、署名スキームとしては同一のものと見なされるため、省略する。

メッセージ回復型署名生成スキーム

1. 80 ビット以上の任意長のメッセージ m を 80 ビットの m_{re} とそれ以降の m_{nr} に分割する。
2. 署名者秘密鍵 x を選択する。
3. m をハッシュ関数に入力し、80 ビットのハッシュ値 h を得る。
4. m_{re}, h を結合し、160 ビット値の d を得る。($d = m_{re} \| h$)
5. 署名生成プリミティブに従いメッセージ d に対応する署名 (r, s) を計算する。
6. 計算した署名 (r, s)、及びメッセージの一部 m_{nr} を結合し、署名文 $r \| s \| m_{nr}$ を出力する。

署名生成プリミティブ:

1. 0＜k＜q を満たす乱数 k を生成する。
2. 楕円曲線上の点（affine 座標）$(x_1, y_1) = kG$ を計算する。
3. $r = d \oplus x_1$ を計算する。
4. r’ = r (mod q) を計算し、r’ = 0 の時乱数 k の生成からやり直す。
5. s = (r’k-r’-1)/(x+1) (mod q) を計算し、s = 0 の時乱数 k の生成からやり直す。
6. 署名として (r, s) を出力する。

メッセージ回復型署名検証スキーム

1. 署名者の秘密鍵に対応した署名者公開鍵 Y を取得する。
2. 署名文を 320 ビットの署名 (r, s) とそれ以降の m_{nr} に分割する。
3. (r, s)と公開鍵 Y を署名検証プリミティブに入力し、メッセージ d を回復する。もしくは “検証失敗” を取得する。”検証失敗” の場合、検証に失敗した旨を出力し処理を終了する。
4. d の前半 80 ビットを m_{re}, 後半を h とする。
5. m_{re} と m_{nr} を結合して m とする.（$m = m_{re} \| m_{nr}$）
6. m をハッシュ関数に入力し、80 ビットのハッシュ値 h’ を得る。
7. h と h’ が一致しない場合、検証に失敗した旨を出力する。一致した時 m を回復したメッセージとして出力する。

署名検証プリミティブ:

1. r’ = r (mod q) を計算する。
2. r’≠0, 0＜s＜q を検証する。この条件を満たさない時 ‘検証失敗’ を出力する。
3. 楕円曲線上の点（affine 座標）$(x_2, y_2) = ((1+r'+s)/r')G + (s/r')Y$ を計算する。
4. $d = r \oplus x_2$ を計算する。
5. 回復したメッセージとして d を出力する。

推奨パラメータ:

各スキーム(160/192/0EF)について各々１つの推奨パラメータのみが記載されている。

◆ 160 ビット素体上の楕円パラメータ

$p = 2^{160} - 33689$

= 1461 50163 73309 02918 20368 48327 16283 01965 59325 09287

$= [\ ffffffff\ ffffffff\ ffffffff\ ffffffff\ ffff7c67\ ]_{16}$

楕円曲線 $E(GF(p)) : y^2 = x^3 + ax + b$

$a = -3$

$= [\ ffffffff\ ffffffff\ ffffffff\ ffffffff\ ffff7c64\ ]_{16}$

b = 221 26390 04360 85830 53354 00546 97282 39214 54692 45783

$= [\ 26c1d102\ 82415e10\ a4995e19\ 80b59224\ d7120957]_{16}$

ベース点 $G = (g_x, g_y)$, 位数 q

g_x = 1

g_y = 199 11984 96906 58063 76419 67878 55655 25945 56232 98470

= [22e0d7c6 1eb0627b 334456c7 a50b77fd a9007da6]$_{16}$

q = 2^{160} − 2678 34839 33601 50259 78183

= 1461 50163 73309 02918 20368 45648 81443 68364 09065 64793

= [ffffffff ffffffff ffffc748 a4eea1b0 dc8744b9]$_{16}$

◆ 192 ビット素体上の楕円パラメータ

p = 2^{192} − 34757

= 627 71017 35386 68076 38357 89423 20766 64161 02355 44446 40344 78139

= [ffffffff ffffffff ffffffff ffffffff ffffffff ffff783b]$_{16}$

楕円曲線 E(GF(p)) : $y^2 = x^3 + ax + b$

a = -3

= [ffffffff ffffffff ffffffff ffffffff ffffffff ffff7838]$_{16}$

b = 570 48185 89025 54558 85102 23460 41752 70002 25150 94686 47895 98136

= [e8a915cd 5ea5560c dca0ee33 8c0b6377 7101fc2b 25a22fb8]$_{16}$

ベース点 G = (g_x, g_y), 位数 q

g_x = 2

g_y = 121 87592 69632 40567 43864 47414 41820 27280 76148 35976 18105 14187

= [31b470c4 16b0d2cc 78fbd092 c0e2c5ea b94a91ce f85d610b]$_{16}$

q = 2^{192} − 1005 28889 60154 04021 58223 05019

= 627 71017 35386 68076 38357 89423 19761 35271 42201 40424 82122 07877

= [ffffffff ffffffff ffffffff df8471f5 7b763287 b0c97905]$_{16}$

◆ 32 ビット×5 次拡大体上の楕円パラメータ

p = 2^{32} − 185 = 42949 67111 = [ffffff47]$_{16}$

f(ξ) = ξ^5 - 2

楕円曲線 E(GF(p^5)) : $y^2 = x^3 + ax + b$

a = -3 = (0, 0, 0, 0, -3)=(0, 0, 0, 0, [ffffff44]$_{16}$)

b = 969768922×α^4+4095377333×α^3+1216762277×α^2+3814912639×α+3024742656

= (969768922, 4095377333, 1216762277, 3814912639, 3024742656)

= ([39cd7fda]$_{16}$, [f41a7fb5]$_{16}$, [488651a5]$_{16}$, [e362f27f]$_{16}$, [b449e900]$_{16}$)

ベース点 G = (g_x, g_y), 位数 q

g_x = (0, 0, 0, 0, 2)

g_y = (2319840967, 2197013598, 3799084265, 643252031, 3306181529)

= ([8a45f6c7]$_{16}$, [82f3c45e]$_{16}$, [e2716ce9]$_{16}$, [26573f3f]$_{16}$, [c5105399]$_{16}$)

q = 1461 50132 25697 40632 17306 09208 36062 44677 38509 59223

= [fffffc63 000538e9 fc3bbe32 da01dc69 c2516d77]$_{16}$

2. 評価結果

2.1 安全性評価

MY-ELLTY ECMR-160/192/OEF-h は、各々異なる 3 件の応募であったが、使用する体が異なるだけで、署名スキームとしては同一のものと見なされるため、3 件まとめて安全性評価を行った。

a)選択文書攻撃による署名の存在的偽造が実現可能であり、安全性に問題があると思われる。使用しているハッシュ関数の出力が、80 ビット(160/OEF)または 96 ビット(192)と短いため、Birthday 攻撃により、2^{40} (ECMR -160/OEF)または2^{48}(ECMR -192) のオーダーの計算量でハッシュ値の衝突が期待され、署名の存在的偽造が可能となる。

b)安全性のためにハッシュ値のビット長を長くすると、回復されるメッセージ長はその分短くなり、応募者の訴求するメリット（署名文長を短くできること）が失われていく。

c) no message attack に対してランダムオラクルモデルでは、安全性は ECDLP に帰着されるとの提案者による証明があるが、証明に誤りがある。証明を正しく行うためには、メッセージ m だけではなく、kG もハッシュ関数に入力するようにスキームを変更する必要がある。したがって提案スキームのままでは、上記安全性は証明されていない。

d)使用している楕円曲線パラメータの選択方法についての言及が不十分であり、なぜその曲線を選んだのかという根拠が不明確である。

e)以下の 2 点に関して、仕様が不明確である。

-SHA-1 のどの 80/96 ビットをハッシュ値として使用するか。

-メッセージ長が 80/96 ビット未満の場合、それを 80/96 ビットに拡張する方法。

ただし、これらの点が明確になってもならなくても、上記安全性評価に影響はない。

2.2 ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

・N=1,024 ビットの RSA 暗号（署名同等以上の強度）

（２）パラメータの設定方式

・事前に発生し、固定。バイナリーコードに埋め込んである。

・楕円パラメータは固定という前提で実装している。そのため、システムパラメータの生成や、システムパラメータの読み込みを行うコードを保持しない

（3）使用パラメータの検証状況

・使用した楕円パラメータは、擬似ランダムに生成したものであり、MOV, FR 帰着攻撃、SSSA 攻撃について安全性を検証(判定条件をクリア)。

（4）パラメータ・サイズのバリエーション

・実測したパラメータのサイズは、160 ビット、OEF32 ビット×５拡大体版、192 ビットの 3 種類。安全性は､160 ビット､OEF32 ビット×５拡大体版については､RSA1024 ビット相当､192 ビットについては､RSA1536 ビット相当。

（5）その他

・鍵対の生成や、署名生成時に乱数を必要とする。

・実システムへの適用においては、安全性の高い疑似乱数ルーチンを用いるべきである。

［実装上で使用した手法］

・　予備計算は事前に実施し、バイナリーに組み込んである。

・　テーブル参照方式で楕円演算を高速化。テーブル初期化等のセットアップ処理を要しない。

［システムパラメータ生成］

・　事前に実施されており、今回は測定しなかった。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠		実装パラメータと位置づけ	その他
MY-ELLITY ECMR-192-h	ECDLP	192	RSA1536bit 相当	・ MOV, FR 帰着攻撃、SSSA 攻撃について安全性を検証(判定条件をクリア)。 ・ パラメータ固定。バイナリ組み込み。 ・ 予備計算結果をバイナリ組み込み。 ・ テーブル参照。
MY-ELLITY ECMR-OEF-h		OEF	RSA1024bit 相当	
MY-ELLITY ECMR-160-h		160	RSA1024bit 相当	

［鍵対生成］

（1）判定条件

・関数内部でベース点の位数のビットサイズと等しいビットサイズの乱数を生成し、その乱数に対してベース点の位数を法としてモジュロを取り、秘密鍵とする。

・前記秘密鍵を用いてベース点をスカラ倍して公開鍵を生成する。

（2）乱数生成方式

・　Ｃ言語の rand 関数を使用。

・　実使用時には、より安全な乱数生成手段が必要とコメントあり。

（3）測定結果［鍵対生成の速度］

測定対象	平均実行時間	備考
MY-ELLITY ECMR-192-h	0.8ms	乱数生成を含む。 但し、実使用時には、乱数生成部の入れ替えが必要。当然時間も変化する。
MY-ELLITY ECMR-OEF-h	0.7ms	
MY-ELLITY ECMR-160-h	0.7ms	

［署名生成］

（1）パディング

・ SHA-1 の出力（80 ビット）を使用。

（2）ハッシュ関数

・ SHA-1 を使用。

・ ハッシュサイズは、80 ビット。

・ ハッシュの結果は、中間処理ファイルには出力されていない。

（3）測定結果［署名生成の速度］

測定対象	平均実行時間	補助関数	備考	
MY-ELLITY ECMR-192-h	2.0ms	SHA1	署名対象データのサイズ：31KＢ	乱数生成を含む。SHA-1 を補助関数として使用。ファイルの長さに比例して、SHA-1 の処理時間が増加する。
	9.2ms		署名対象データのサイズ：178KB	
MY-ELLITY ECMR-0EF-h	1.9ms		署名対象データのサイズ：31KＢ	
	9.6ms		署名対象データのサイズ：178KB	
MY-ELLITY ECMR-160-h	1.9ms		署名対象データのサイズ：31KＢ	
	9.7ms		署名対象データのサイズ：178KB	

［署名検証］

（1）検証確認ルーチンの有無

・検証結果判定機構は組み込み済みであるが、確認手段（ファイルへの出力など）はない。

（2）測定結果［署名検証の速度］

測定対象	平均実行時間	備考	
MY-ELLITY ECMR-192-h	5.4ms	署名対象データのサイズ：31KＢ	検証の正当性は、チェックするようになっているが、今回の実装では、その結果は出力していない。
	12.8ms	署名対象データのサイズ：178KB	
MY-ELLITY ECMR-0EF-h	4.4ms	署名対象データのサイズ：31KＢ	
	11.9ms	署名対象データのサイズ：178KB	
MY-ELLITY ECMR-160-h	4.3ms	署名対象データのサイズ：31KＢ	
	11.9ms	署名対象データのサイズ：178KB	

［コードサイズ］

評価対象	オブジェクトサイズ（申告値）	備考
MY-ELLTY ECMR-160-h	226,306 Bytes	Ｃ言語 多倍長演算ルーチン GMP の一部を含んでいる。
MY-ELLTY ECMR-192-h	230,626 Bytes	
MY-ELLTY ECMR-0EF-h	256,640 Bytes	

参考文献

[1]宮地　充子、“強化されたメッセージ復元型署名”、SCIS’96, 2C, (1996)

[2]A.Miyaji, “Another countermeasure to forgeries over message recovery signature”, Trans. IEICE, Fundamentals. Vol. E80-A, No. 11(1997)

4.4.7 EPOC-1

1. 暗号技術

1.1 技術概要

EPOC-1は、岡本龍明、内山成憲、藤崎英一郎(NTT)により提案された素因数分解問題(IF)に基づく守秘目的の公開鍵暗号で、OU(Okamoto-Uchiyama)暗号関数（基本暗号プリミティブ）を FO-1 法により変換したものである。

＜暗号の発表年、提案論文＞

(1) EPOC暗号基本方式　発表年:1998年

Okamoto,T. and Uchiyama,S.:A New Public-Key Cryptosystem as Secure as Factoring,
Proc.of Eurocrypt'98,LNCS 1403,Springer-Verlag,pp.308--318(1998).

(2)変換方式　発表年:1999年

Fujisaki,E. and Okamoto,T.:How to Enhance the Security of Public-Key Encryption
at Minimum Cost,Proc.of PKC'99,Springer-Verlag,LNCS 1560,pp.53--68 (1999).

知的財産権　(応募者資料による)

(提案者特許とその扱い)

出願番号(公開番号) 10-320172、

発明の名称　ランダム関数利用公開鍵暗号の暗号装置、復号装置

出願番号(公開番号) 2000-32461

発明の名称　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

出願番号(公開番号) EP 98123917.1(EP 924895)(Canada 2256179)

発明の名称 ENCRYPTION AND DECRYPTION DEVICE FOR PUBLIC-KEY CRYPTOSYSTEMS AND RECORDING MEDIUM WITH THEIR PROCESSING PROGRAMS RECORDED THEREON

・排他的かつ妥当な条件で他者に実施許諾する。

・関連する他社特許の訴求はなし。

応募暗号技術仕様の公開Webアドレス　http://info.isl.ntt.co.jp/

1.2 技術仕様

【鍵生成】

[入力] 正整数であるセキュリティパラメータ k

[出力] 公開鍵 (n, g, h, HID, pLen, mLen, hLen, rLen)

秘密鍵 (p, g_p)

1. 2つの素数 p, q ($2^{k-1} \leqq p,q \leqq 2^k - 1$) を選択し、$n=p^2q$ を計算する。
2. $g \in Z_n^*$ をランダムに選択する。ここで $g_p = g^{p-1} \bmod p^2$ の位数が p となるようにする。

3. $h_0 \in Z_n^*$ をランダムかつ g と独立に選択する。 $h = h_0^n \bmod n$ を計算する。
4. pLen = k とし、mLen と rLen を mLen + rLen ≦ pLen − 1 を満たすように設定する。
5. ハッシュ関数 $H: \{0,1\}^{mLen + rLen} \rightarrow \{0,1\}^{hLen}$ を定め、その識別番号を HID とする。

【暗号化】

[入力] 平文 $m \in \{0,1\}^{mLen}$, 公開鍵 (n, g, h, HID, pLen, mLen, hLen, rLen)

[出力] 暗号文 C

1. $R \in \{0,1\}^{rLen}$ を一様にランダムに選択し、r = H(m || R)を計算する。
2. $C = g^{(m||R)} h^r$ を計算し暗号文とする。

【復号】

[入力] 暗号文 C

公開鍵 (n, g, h, HID, pLen, mLen, hLen, rLen)

秘密鍵 (p, g_p)

[出力] 平文 $m \in \{0,1\}^{mLen}$ または出力なし。

L(x) := (x-1)/p とする。

1. $C_p = C^{p-1} \bmod p^2$ を計算する。
2. $X = L(C_p)/L(g_p) \bmod p$ を計算する。
3. 以下の 2 つの式が成立するかどうかを検証する。

$X \leqq 2^{mLen + rLen} - 1$,

$C = g^X h^{H(X)} \bmod n$.

4. もし成立すれば、X の上位 mLen ビットを平文として出力する。 成立しない場合は、何も出力しない。

＜推奨パラメータ＞

(1) 応募暗号技術仕様によるパラメータの推奨値

・k: 320 ビット以上 (n のサイズを 960 ビット以上)

・hLen: 128 ビット以上

(2) 応募暗号技術仕様による「性能評価」での代表的なパラメータ

ケース 1（強い安全性仮定の下）

n のサイズ 1152 ビット, mLen=128, rLen=80, hLen=208

ケース 2（弱い安全性仮定の下）

n のサイズ 1152 ビット, mLen=128, rLen=80, hLen=832

1.3 その他

IEEE P1363a/D4 (Draft Version 4),May 22,2000 に記載されている。

2. 評価結果

2.1 安全性評価

p 部分群問題の困難性と、ランダム関数が存在するという仮定のもとで、適応的選択暗号文攻撃に対して強秘匿であるという証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 方式と異なることなどにより、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。また証明で十分な安全性を導くためには、hの位数は大きくなければならないとの指摘がある。ただし、これらの指摘の妥当性について本評価では結論を得るには至っていないため、パラメータ選択について更なる検討が必要である。

2.2 ソフトウェア実装評価

［実装仕様］

（1）使用したパラメータの位置づけ

- 合成数の長さ 1,024 ビットの RSA 暗号相当の強度を持たせるため、EPOC-1 の合成数の長さは、384 ビット×３＝1,152 ビット。
- EPOC-1 のデータサイズは、128 ビット。

（2）パラメータの設定方式

- 乱数の種をファイルとして読み込む。

（3）使用パラメータの検証状況

- GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。本関数で用いる素数判定アルゴリズムは Miller-Rabin 法である。第 2 引数 reps の値が大きいほど誤判定の確率を低めることができるが、それだけ多くの時間を要し、現実的には 5 から 10 くらいが適当とされている。本実装では reps の値を 5 とした。

（4）パラメータ・サイズのバリエーション

- 共通鍵暗号の鍵（高々 256 ビット）の配送。
- 短いデータ（高々 256 ビット）の秘匿通信。

［実装上で使用した手法］

- フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。
- 仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
EPOC-1	素因数分解問題	EPOC-1 の合成数の長さは、384 ビット×３＝1,152 ビット。	評価用 EPOC-1 のデータサイズは、128 ビット。 素数判定は、Miller-Rabin 法を使用。

［システムパラメータ生成］

・ システム生成は、事前に済ませており、今回の評価では実施しなかった。

［鍵対生成］

（1）素数判定条件

・ GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。

（2）乱数生成方式

・ まず GNU MP 3.1.1 の関数 gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムは GNU MP のデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数 gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数 mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

（3）測定結果

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
EPOC-1	73.9ms	乱数生成は含まない
	417.6ms	乱数生成、素数生成を含めた測定値

［暗号化］

（1）パディング

・ 仕様書に記載されている方法でハッシュを作成するため、パディングは不要。

（2）ハッシュ関数

・ ハッシュ関数として SHA-1 を使用。

（3）測定結果

［暗号化の速度］

測定対象	平均実行時間	備考
EPOC-1	13.9ms	なし

［復号］

（1）測定結果［復号の速度］

測定対象	平均実行時間	備考
EPOC-1	21.9ms	なし

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
EPOC-1	177,058byte	C言語（Intel C/ C++）ソースコードサイズ

4.4.8 EPOC-2

1. 暗号技術

1.1 技術概要

EPOC-2 は岡本龍明、内山成憲、藤崎英一郎(NTT)により提案された素因数分解問題(IF)に基づく守秘目的の公開鍵暗号で、OU(Okamoto-Uchiyama)暗号関数（基本暗号プリミティブ）を FO-2 法により変換したものである。

＜暗号の発表年、提案論文＞

(1) EPOC 暗号基本方式　発表年:1998 年

Okamoto,T. and Uchiyama,S.:A New Public-Key Cryptosystem as Secure as Factoring, Proc.of Eurocrypt'98,LNCS 1403,Springer-Verlag,pp.308--318(1998).

(2)変換方式　発表年:1999年

Fujisaki,E. and Okamoto,T.:How to Enhance the Security of Public-Key Encryption at Minimum Cost,Proc.of PKC'99,Springer-Verlag,LNCS 1560,pp.53--68 (1999).

知的財産権　(応募者資料による)

(提案者特許とその扱い)

EPOC-1 を参照のこと。

応募暗号技術仕様の公開 Web アドレス　http://info.isl.ntt.co.jp/

1.2 技術仕様

【鍵生成】

[入力] 正整数であるセキュリティパラメータ k。

[出力] 公開鍵 (n, g, h, H1ID,H2ID,SEID pLen, hLen, gLen, rLen) $\in Z^{10}$ 、
秘密鍵 (p, g_p)のペア。

1. 2 つの素数 p, q ($2^{k-1} \leqq p,q \leqq 2^k - 1$) を選択し、$n=p^2q$ を計算する。
2. $g \in Z_n^*$ をランダムに選択する．ここで $g_p = g^{p-1} \bmod p^2$ の位数が p となるようにする。
3. $h_0 \in Z_n^*$ をランダムかつ g と独立に選択する．$h= h_0^n \bmod n$ を計算する。
4. pLen = k とし、mLen と rLen を $rLen \leqq pLen - 1$ を満たすように設定する。
5. 2 つのハッシュ関数 $H_1 := \{0, 1\}^{mLen + rLen} \rightarrow \{0, 1\}^{hLen}$ と $H_2 := \{0, 1\}^{rLen} \rightarrow \{0, 1\}^{gLen}$ を定め、その識別番号を H1ID および H2ID とする。
6. 共通鍵暗号 SymE を定め、その識別番号を SEID とする．ここで SymE=(SymEnc,SymDec)は gLnen ビットの共通鍵 K を持つ共通鍵暗号・復号アルゴリズムの対である。暗号アルゴリズム SymEnc は鍵 K と平文 m を入力として暗号文 SymEnc(K, m)を出力する。復号アルゴリズム SymDec は鍵 K と暗号文 c を入力と

して平文 SymDec(K,c)を出力する。

【暗号化】

[入力] 平文 $m \in \{0,1\}^{mLen}$

公開鍵 (n, g, h, H1ID,H2ID,SEID, pLen, hLen, gLen, rLen)

[出力] 暗号文 $C=(C_1, C_2)$

1. $R \in \{0,1\}^{rLen}$ を一様にランダムに選択し、$H_2(R)$および $r = H_1(m \,||\, R)$を計算する。
2. $C_1 = g^R h^r$ を計算する。
3. $C_2 = SymEnc(H_2(R), m)$を計算する。
4. $C=(C_1, C_2)$を暗号文として出力する。

【復号】

[入力] 暗号文 $C=(C_1, C_2)$

公開鍵 (n, g, h, H1ID,H2ID,SEID, pLen, hLen, gLen, rLen)

秘密鍵 (p, g_p)

[出力] 平文 $m \in \{0,1\}^{mLen}$ または出力なし。

$L(x) := (x-1)/p$ とする。

1. $C_p = C_1^{\,p-1} \bmod p^2$ を計算する。
2. $R' = L(C_p)/L(g_p) \bmod p$ を計算する。
3. 以下の式が成立するかどうかを検証する。

$R' \leqq 2^{rLen}-1$.

もし成立すれば、$m' = SymDec(H_2(R), C_2)$を計算する。もし成立しなければ復号結果として何も出力しない。

4. $r' = H_1(m'||R')$を計算する。
5. $C_1 = g^{R'} h^{r'} \bmod n$ が成立するかどうか検証する。

もし成立すれば、m'を平文として出力する。成立しない場合は、何も出力しない。

＜推奨パラメータ＞

(1) 応募暗号技術仕様によるパラメータの推奨値

・k: 320 ビット以上 (n のサイズを 960 ビット以上)

・hLen: 128 ビット以上

(2) 応募暗号技術仕様による「性能評価」での代表的なパラメータ(バーナム暗号を使用した場合)

ケース１（強い安全性仮定の下）

n のサイズ 1152 ビット，rLen=128，gLen=128，hLen=128

ケース 2（弱い安全性仮定の下）

n のサイズ 1152 ビット，rLen=128，gLen=128，hLen=832

1.3 その他

EPOC-1 を参照のこと。

2. 評価結果

2.1 安全性評価

$n=p^2q$ 型の素因数分解問題の困難性と、ランダム関数が存在するという仮定のもとで、適応的選択暗号文攻撃に対して強秘匿であるという証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 方式と異なることなどにより、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。また提案者の推奨パラメータ：

k = 384 ビット、rLen=128 ビット　に対して、証明で十分な安全性を導くためには、
rLen は k-1 に十分近く、ｈの位数は大きくなければならない

との指摘がある。ただし、これらの指摘の妥当性について本評価では結論を得るには至っていないため、パラメータ選択について更なる検討が必要である。

2.2 ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

- 合成数の長さ 1,024 ビットの RSA 暗号相当の強度を持たせるため、EPOC-2 の合成数の長さは、384 ビット×３＝1,152 ビット。
- EPOC-2 のデータサイズは、128 ビット。

（２）パラメータの設定方式

・乱数の種をファイルとして読み込む。

（３）使用パラメータの検証状況

- GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。本関数で用いる素数判定アルゴリズムは Miller-Rabin 法である。第 2 引数 reps の値が大きいほど誤判定の確率を低めることができるが、それだけ多くの時間を要し、現実的には 5 から 10 くらいが適当とされている。本実装では reps の値を 5 とした。

（４）パラメータ・サイズのバリエーション

- 任意長の共通鍵暗号鍵の配送。
- 適当な共通鍵暗号と併用することによる長い平文の秘匿通信、特にカプセル的な利用方法（つまり、鍵配送とデータ配送が同期しているような利用形態）

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
EPOC-2	素因数分解問題	EPOC-2 の合成数の長さは、384 ビット×３＝1,152 ビット。	評価用 EPOC-2 のデータサイズは、128 ビット。 素数判定は、Miller-Rabin 法を使用。

[実装上で使用した手法]

・フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。

・仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

[システムパラメータ生成]

システム生成は、事前に済ませており、今回の評価では実施しなかった。

[鍵対生成]

（１）素数判定条件

GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。

（２）乱数生成方式

・まず GNU MP 3.1.1 の関数 gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムは GNU MP のデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数 gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数 mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

（３）測定結果

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
EPOC-2	73.9ms	乱数生成は含まない
	577.0ms	乱数生成、素数生成を含めた測定値。

[暗号化]

（１）パディング

・　仕様書に記載されている方法でハッシュを作成するため、パディングは不要。

（２）ハッシュ関数

・　ハッシュ関数として SHA-1 を使用。

（３）測定結果

[暗号化の速度]

測定対象	平均実行時間	備考
EPOC-2	10.9ms	なし

［復号］

（１）測定結果

［復号の速度］

測定対象	平均実行時間	備考
EPOC-2	18.9ms	なし

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
EPOC-2	187,662byte	Ｃ言語（Intel C/ C++）ソースコードサイズ

4.4.9 EPOC-3

1. 暗号技術

1.1 技術概要

EPOC-3 は岡本龍明、内山成憲(NTT), Pointcheval,D.(ENS)により提案された素因数分解問題(IF)に基づく守秘目的の公開鍵暗号で、OU(Okamoto-Uchiyama)暗号関数（基本暗号プリミティブ）を OP 法により変換したものである。

＜暗号の発表年、提案論文＞

(1) EPOC 暗号基本方式　発表年:1998 年

Okamoto,T. and Uchiyama,S.:A New Public-Key Cryptosystem as Secure as Factoring,
Proc.of Eurocrypt'98,LNCS 1403,Springer-Verlag,pp.308--318(1998).

(2)変換方式　発表年:2000年

Okamoto,T. and Pointcheval,D.:OCAC:an Optimal Conversion for Asymmetric Cryptosystems, manuscript (2000).

知的財産権　(応募者資料による)

(提案者特許とその扱い)

EPOC-1 を参照のこと。

応募暗号技術仕様の公開 Web アドレス　http://info.isl.ntt.co.jp/

1.2. 技術仕様

【鍵生成】

[入力] 正整数であるセキュリティパラメータ k

[出力] 公開鍵 (n, g, h, H1ID,H2ID,SEID, pLen, hLen, gLen, Rlen, rLen)

秘密鍵 (p, g_p)

1. 2 つの素数 p, q ($2^{k-1} \leqq p,q \leqq 2^k - 1$) を選択し、$n=p^2q$ を計算する。
2. $g \in Z_n^*$ をランダムに選択する。ここで $g_p = g^{p-1} \bmod p^2$ の位数が p となるようにする。
3. $h_0 \in Z_n^*$ をランダムかつ g と独立に選択する。 $h= h_0^n \bmod n$ を計算する。
4. pLen = k とし、mLen と RLen を RLen $\leqq$ pLen − 1 を満たすように設定する。
5. 2つのハッシュ関数 $H_1 := \{0, 1\}^{3k+kLen+Rlen+mLen} \rightarrow \{0, 1\}^{hLen}$ と $H_2 := \{0, 1\}^{RLen} \rightarrow \{0, 1\}^{gLen}$ を定め、その識別番号を H1ID および H2ID とする。
6. 共通鍵暗号 SymE を定め、その識別番号を SEID とする. ここで SymE=(SymEnc,SymDec)は gLnen ビットの共通鍵 K を持つ共通鍵暗号・復号アルゴリズムの対である.暗号アルゴリズム SymEnc は鍵 K と平文 m を入力として暗号文 SymEnc(K, m)を出力する。復号アルゴリズム SymDec は鍵 K と暗号文 c を入力と

して平文 SymDec(K,c)を出力する。

【暗号化】

[入力] 平文 $m \in \{0,1\}^{mLen}$

公開鍵 (n, g, h, H1ID, H2ID, SEID, pLen, hLen, gLen, RLen, rLen)

[出力] 暗号文 C=(C_1 , C_2 , C_3)

1. $r \in \{0,1\}^{rLen}$ および $R \in \{0,1\}^{RLen}$ を一様にランダムに選択し、H_2 (R)を計算する。
2. $C_1 = g^R h^r$ を計算する。
3. C_2 = SymEnc(H_2 (R), m)を計算する。
4. $C_3 = H_1(C_1 || C_2 || C_3 || R || m)$を計算する。
5. C = (C_1 , C_2 , C_3)を暗号文として出力する。

【復号】

[入力] 暗号文 C=(C_1 , C_2 , C_3)

公開鍵 (n, g, h, H1ID, H2ID, SEID, pLen, hLen, gLen, RLen, rLen)

秘密鍵 (p, g_p)

[出力] 平文 $m \in \{0,1\}^{mLen}$ または出力なし。

L(x) := (x-1) / p とする。

1. $C_p = C_1^{\,p-1} \bmod p^2$ を計算する。
2. R' = L(C_p) / L(g_p) mod p を計算する。
3. 以下の式が成立するかどうかを検証する。

$$R' \leqq 2^{rLen}-1.$$

もし成立すれば、m' = SymDec(H_2 (R'), C_2)を計算する。もし成立しなければ復号結果として何も出力しない。

4. r' = H_1(m' || R')を計算する。
5. $C_3 = H_1(C_1 || C_2 || C_3 || R || m)$ が成立するかどうか検証する.

もし成立すれば、m'を平文として出力する。成立しない場合は、何も出力しない。

＜推奨パラメータ＞

(1) 応募暗号技術仕様によるパラメータの推奨値

・k: 320 ビット以上 (n のサイズを 960 ビット以上)

・hLen: 128 ビット以上

(2) 応募暗号技術仕様による「性能評価」での代表的なパラメータ(バーナム暗号を使用した場合)

ケース１（強い安全性仮定の下）

n のサイズ 1152 ビット，RLen=128，gLen=128，rLen=hLen=128

ケース 2（弱い安全性仮定の下）

n のサイズ 1152 ビット，RLen=128，gLen=128，rLen=832，hLen=128

1.3 その他

EPOC-1 を参照のこと。

2. 評価結果

2.1 安全性評価

$n=p^2q$ 型の GAP-素因数分解問題の困難性と、ランダム関数が存在するという仮定のもとで、適応的選択暗号文攻撃に対して強秘匿であるという証明可能安全性を有する。この仮定を満たすため適切なパラメータを選択することに注意を払う必要がある。法である合成数の素因数分解の形が RSA 方式と異なることなどにより、RSA 方式と同じサイズでも安全性は同等とは限らないことに注意を払う必要がある。また、提案者の推奨パラメータ：

k=384、RLen = 128

に対して、証明で十分な安全性を導くためには、

RLen は k-1 に十分近く、

ｈの位数は大きくなければならない

との指摘がある。ただし、これらの指摘の妥当性について本評価では結論を得るには至っていないため、パラメータ選択について更なる検討が必要である。

2.2 ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

- 合成数の長さ 1,024 ビットの RSA 暗号相当の強度を持たせるため、EPOC-3 の合成数の長さは、384 ビット×３＝1,152 ビット。
- EPOC-3 のデータサイズは、128 ビット。

（２）パラメータの設定方式

・乱数の種をファイルとして読み込む。

（３）使用パラメータの検証状況

- GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。本関数で用いる素数判定アルゴリズムは Miller-Rabin 法である。第 2 引数 reps の値が大きいほど誤判定の確率を低めることができるが、それだけ多くの時間を要し、現実的には 5 から 10 くらいが適当とされている。本実装では reps の値を 5 とした。

（４）パラメータ・サイズのバリエーション

- 任意長の共通鍵暗号鍵の配送。
- 適当な共通鍵暗号と併用することによる長い平文の秘匿通信、特にカプセル的な利用方法（つまり、鍵配送とデータ配送が同期しているような利用形態）。
- 適当な共通鍵暗号と併用することによる長い平文の秘匿通信、特にセッション的利用方法（つまり、セッション開設時における鍵配送とそれ以降の該セッション開設中での複数回の共通鍵暗号によるデータ暗号化）。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
EPOC-3	素因数分解問題(IF)	EPOC-3 の合成数の長さは、384 ビット×3＝1,152 ビット。	評価用 EPOC-3 のデータサイズは、128 ビット。 素数判定は、Miller-Rabin 法を使用。

［実装上で使用した手法］

・フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。

・仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

［システムパラメータ生成］

システム生成は、事前に済ませており、今回の評価では実施しなかった。

［鍵対生成］

（1）素数判定条件

- GNU MP 3.1.1 の関数 mpz_probab_prime_p(mpz_t n, int reps)を用いて素数判定を行う。

（2）乱数生成方式

・まず GNU MP 3.1.1 の関数 gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムは GNU MP のデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数 gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数 mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

（3）測定結果

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
EPOC-3	73.9ms	乱数生成は含まない
	743.3ms	乱数生成、素数生成を含めた測定値。

［暗号化］

（1）パディング

- 仕様書に記載されている方法でハッシュを作成するため、パディングは不要。

（2）ハッシュ関数

・　ハッシュ関数として SHA-1 を使用。

（3）測定結果

［暗号化の速度］

測定対象	平均実行時間	備考
EPOC-3	11.0ms	なし

［復号］

（1）測定結果

［復号の速度］

測定対象	平均実行時間	備考
EPOC-3	8.3ms	なし

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
EPOC-3	186,851byte	C言語（Intel C/ C++）ソースコードサイズ

4.4.10 HIME-1

1 暗号技術

1.1 技術概要

HIME-1は、2000年に西岡玄次と瀬戸洋一により考案された、モジュラー平方関数をベースとした基本方式と1994年にMihir Bellare とPhillip Rogaway により提案されたOAEP方式を組み合わせたものであり、2000年に日立製作所の西岡玄次と佐藤尚宜、瀬戸洋一により発表された。

HIME-1 は($p^d q, d$:奇数)型の 素因数分解(IF) の困難性を前提に基づき、守秘目的の公開鍵暗号方式である。(注:応募者は鍵共有の分類に応募していた)

知的財産権 (応募者資料による)

HIME-1 に関する知的所有権は、(株) 日立製作所が所有している。

米国特許第5,103,479 号“ ENCIPHER METHOD AND DECIPHER METHOD ”

特願2000-208237 「公開鍵暗号方法および公開鍵暗号を用いた通信システム」採用された場合には、非差別的、かつ、適正な対価条件でライセンス提供することが宣言されている。

応募暗号技術仕様の公開Webアドレス http://www.sdl.hitachi.co.jp/crypto/

1.2 技術仕様

暗号化、復号関数の概要を示す。

［秘密鍵］

二つの素数 p, q(但し, $p \equiv 3 \pmod 4, q \equiv 3 \pmod 4$)

［公開鍵］

n ($= p^d q, |pq| = k, d \geq 3$ (但し, d は奇素数))

［ハッシュ関数］

$$H_1 : \{0,1\}^{k_0} \rightarrow \{0,1\}^{k-k_0-2}$$

$$H_2 : \{0,1\}^{k-k_0-2} \rightarrow \{0,1\}^{k_0}$$

【暗号化】

以下のようにして、平文 $m (m \in \{0,1\}^{k-k_0-k_1-2})$ を暗号化して、暗号文 c を得る。

1. r :乱数 $(r \in \{0,1\}^{k_0})$

2. $z \leftarrow (((m \| 0^{k_1}) \oplus H_1(r)) \| (r \oplus H_2((m \| 0^{k_1}) \oplus H_1(r))))$

3. $c \leftarrow z^{2n} \bmod n$

4. $a \leftarrow \left(\frac{z}{n}\right)$

【復号】

以下のようにして、暗号文cを復号して平文mを得る。

1. $z_p \leftarrow c\frac{(p+1)q^{-1}}{4} \bmod p$

2. $z_q \leftarrow c\frac{(q+1)p^{-d}}{4} \bmod q$

3. $z_{p,} z_q$ から、中国人の剰余定理を用いて、$c \equiv z^{2n} \bmod n$を満たすz'_jを計算し、

$a = \left(\frac{z'_j}{n}\right)$かつ$z'_j < 2^{k-2}$を満足するものを求める。

4. $z'_j = a_j \| b_j (a_j \in \{0,1\}^{k-k_0-2}, b_j \in \{0,1\}^{k_0})$に対して

$$m = \begin{cases} [z'_j]^{k-k_0-k_1-2} & \text{if } [H_1(H_2(a_j) \oplus b_j) \oplus a_j]_{k_1} = 0^{k_1} \\ \text{“reject”} & \text{otherwise} \end{cases}$$

としてmを出力する。

但し$[a]^k, [a]_k$は各々aの上位，および，下位kビットを表す。

2 評価結果

2.1 安全性評価

仕様が明確性に欠け非常に曖昧なため、スキームを一意に特定できず，その安全性を評価できない。詳細評価では仕様が曖昧な部分を適宜修正して行なっているが、その修正法の正当性は保証できていない点に注意が必要である。

また、以下は、本スキームを守秘として分類した場合の評価である。

(a) プリミティブの安全性

現時点で有効な攻撃法は確認されていない。

プリミティブの安全性は、法nの元での平方根(正確には、$2n$乗根）求解の困難性に基づく。

OAEP に用いられる乱数は十分にランダムであり、ハッシュ関数の出力もまたランダムであることが求め

られる。但し、利用されるハッシュ関数などが特定されていない。また、H_2(仕様書ではH_1)の出力長は128 ビットと規定されているため、近い将来、これのコリジョンは比較的容易に計算可能となり、注意が必要である。

(b) パラメータの安全性

$p^d q$ (d : 奇数) 型の素因数分解の困難性に基づく。特殊な素因数を用いているため、これの素因数分解の困難性は未知である。

(c) パラメータサイズの安全性

仕様では鍵サイズパラメータの推奨値がないが、1024 ビット鍵を使用した場合，現時点では十分な耐性を持つと考えられるが、15 ～20 年後の安全性は保証できないため注意が必要である。パラメータを可変とできるよう、仕様を変更すべきである。

また、各種パラメータの指定理由が明確でないため、その妥当性を検証することができない。

(d) スキームの安全性

仕様ではランダムオラクルモデルの基で、OAEP を利用して、その安全性(適応的選択暗号文攻撃に対する強秘匿性: IND-CCA2) を主張している。

まず、OAEP を適用可能とするには、プリミティブが一方向置換性を満たすことが求められる。一方向性については満足するように思われるが、法として特殊な素因数を用いているためにその点注意が必要である。また、置換性については特殊な形(平文空間を制限したり、Jacobi 記号との併用により実現) であるが満足している。

さらに、最近、OAEP だけではIND-CCA2 を証明できないという欠陥が明らかになったため、現時点の証明では本スキームの安全性を保証できていない(証明可能になるとの指摘もある)。

2.2 SW 実装評価

本仕様は曖昧性を有している。具体的には、鍵生成部の実装方法においては数体ふるい法を考慮し合成数nのサイズを1024ビット程度とし、nの素因数p,qの選択にあたってはp±1法を考慮ことが応募者により示されている。鍵生成において256ビットの素数p,qを生成する必要があるが、(p±1)法による素因数分解および周期性による攻撃を避けるためにp.qは以下の条件を満たすことが望ましい。

1） p+1 と p-1 はそれぞれ十分に大きな素数 p1,p2 を因数に持つ(q も同様)。

2） p1+1 と p1-1 はそれぞれ十分に大きな素数 s1,r1 を因数に持ち、p2+1 と p2-1 はそれぞれ大きな素因数 s2,r2 を因数に持つ(q も同様)。

この仕様を追加することによりソフトウェア実装評価が可能となった。

［実装仕様］

（1） 使用したパラメータの位置づけ

(ア) N=1,024 ビットの RSA 暗号（守秘同等以上の強度）

(イ) 具体的には、法 N は、$N=p^3q$（d=3）の形式であり、p,q は各々256 ビットである。

(ウ) p,q の選定にあたっては、p±1 法を考慮。

(エ) 公開鍵は、仕様書上(N,k,d)の３種であるが、今回の実装評価では、d=3 に固定した。

（2） パラメータの設定方式

・実装上、公開鍵のうち、d のみを組み込んでいる。

（3） 使用パラメータの検証状況

・検証の必要なし

（4） パラメータ・サイズのバリエーション

・今回の実装では d は ’3’ に固定。仕様書上は可変である。

（5） コンパイラの設定

① －O2 実行速度を優先した速度最適化を実行した。

② －QxK 命令コードを実行するプロセッサを PentiumⅢ(SIMD サポート)のみに限定する。(-G6 オプションと併せて対象となるプロセッサを PentiumⅢに限定した。)

③ －Zp16 構造体の境界調整を 16 バイト境界に設定した。

［実装上で使用した手法］

(ア) べき乗演算においてモンゴメリー乗算及び 4ary 法を使用。

［システムパラメータ生成］

(イ) 今回は測定しなかった。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
HIME-1	素因数分解問題	法 N は、$N=p^3q$ (d=3) の形式であり、p,q は各々256 ビットである。 N=1,024 ビットの RSA 暗号(守秘同等以上の強度)	・p_1, p_2, p_3, p_4 の選定にあたっては、p±1 法を考慮。 ・暗号技術仕様書では n のビット長が 1023 ビット以上であることおよび p_1,p_2,p_3,p_4 が全て異なることは要求していなかったが、問題が生じる場合があるため今回の実装では条件の追加を行った。 ・ べき乗演算においてモンゴメリー乗算及び 4 array 法を使用。

［鍵対生成］

【アルゴリズム 1: 鍵生成】

出力：公開鍵(n,k),秘密鍵(p,q,α,β,z)

Step1: $p \equiv 3 \pmod 4$, |p|=256 を満たす素数 p を選択する。

Step2: $q \equiv 3 \pmod 4$ |q|=256,p≠q を満たす素数 q を選択する。

Step3: k=|pq|を計算する。

Step4: $n = p^3q$ を計算する。

Step5: k=|pq|を計算する。

Step6: $\alpha = q^{-1} \pmod{p-1}$を計算する。

Step7: $\beta = p^{-1} \pmod{q-1}$を計算する。

Step8: $z = q^{-1} \pmod{p}$を計算する。

Step9: 公開鍵(n,k),秘密鍵(p,q,α,β,z)を出力する。

（1） 素数判定条件

鍵生成において 256 ビットの素数 p,q を生成する必要があるが、(p±1)法による素因数分解および周期性による攻撃を避けるために p,q は以下の条件を満たすことが望ましい。

1） p+1 と p-1 はそれぞれ十分に大きな素数 p1,p2 を因数に持つ(q も同様)。

2） p1+1 と p1-1 はそれぞれ十分に大きな素数 s1,r1 を因数に持ち、p2+1 と p2-1 はそれぞれ大きな素因数 s2,r2 を因数に持つ(q も同様)。

HIME-1 の実装では上記条件を満たす素数を次に示す方法で実現した。

p1+1 が素数 s1 を因数に持つ ⇔ $p1 \equiv -1 \pmod{s1}$ —①

p2+1 が素数 r1 を因数に持つ ⇔ $p1 \equiv 1 \pmod{r1}$ —②

s1,r1 は素数であるから

$s1 \cdot g \equiv 1 \pmod{r1}$ となる g が存在する。 $g = s1^{-1} \pmod{r1}$ —③

このとき

$p\_1 = -1+2 \cdot s1 \cdot g \pmod{s1 \cdot r1}$ は①,②を満たす。 —④

p_1 が偶数となる場合はさらに、p_1=p_1+s1・r1 とする。 —④’

そこでまず素数 s1,r1 を生成し③,④,④’に適用し素数候補 $A = p\_1+2 \cdot k \cdot r1 \cdot s1$ とし—⑤

(k:正の整数) 次の候補を $A = A+2 \cdot k \cdot r1 \cdot s1$ とする。

以上をアルゴリズムの形にまとめると

【アルゴリズム 2：p±1 法に対して強い素数生成】

入力：素数 s1,r1

出力：素数 p1

Step1: ③,④,④’,⑤に従い、素数 s1,r1 より素数の候補 A を生成する。

Step2: 3000 番目までの素数からなるテーブル{2,3,5,7,・・・prime[3000]}を作成する。なお今回の実装では予め素数テーブルを用意した。

Step3: k=1 から k=3000 までの剰余テーブルを m[k],m’[k]を作成する。

$m[k] \leftarrow A \pmod{prime[k]}$

$m'[k] \leftarrow 2 \cdot r1 \cdot s1 \pmod{prime[k]}$

Step4: $A \leftarrow A+2 \cdot r1 \cdot s1$

Step5: k=1 から k=3000 までの剰余テーブル m[k]を更新する。

$m[k] \leftarrow (m[k] + m'[k]) \pmod{prime[k]}$

Step6: 全ての m[k]のうち 0 があれば Step4 にもどり、なければ Step7 に進む。

Step7: アルゴリズム 5 を用いて A が素数でないと判断されたら Step4 にもどり、素数と判定されれば Step8 に進む。
Step8: p1 ← A

素数 p(および q)の生成は上記アルゴリズムを用いて素数 s1,r1 より p1 を,素数 s2,r2 より素数 p2 を生成し,さらに素数 p1,p2 に対してアルゴリズム 2 を用いることにより行う。

【アルゴリズム 3：p±1 法に対して強い 256 ビット素数の生成】
出力：256 ビット素数 p
Step1: 55 ビット素数 s1 を生成する。
Step2: 55 ビット素数 r1 を生成する。
Step3: アルゴリズム 2 を用いて s1,r1 から 120 ビット素数 p1 を生成する。
Step4: 55 ビット素数 s2 を生成する。
Step5: 55 ビット素数 r2 を生成する。
Step6: アルゴリズム 2 を用いて s2,r2 から 120 ビット素数 p2 を生成する。
Step7: アルゴリズム 2 を用いて p1,p2 から 256 ビット素数 p を生成し p を出力する。

アルゴリズム 3 では 55 ビット素数を生成する必要があるが、その生成には次のアルゴリズム 4 を用いた。

【アルゴリズム 4：55 ビット素数生成】
出力：55 ビット素数 r
Step1: ランダムに 55 ビット整数 r を生成する。
Step2: r を 3000 番目までの素数(テーブル参照する)で試し割し r が素数を因数として持てば Step1 に戻り、そうでなければ Step3 に進む。
Step3: アルゴリズム 5 を用いて r が素数でないと判断されたら Step1 にもどり、素数と判定されれば Step4 に進む。
Step4: 素数 r を出力する。
素数判定については暗号技術仕様書では例として Miller-Rabin 法をあげているが、今回の実装においては Solovay-Strassen 法を用いた。

【アルゴリズム 5：素数判定(Solovay-Strassen 法)】
入力：素数の候補 n
出力： n が素数であれば 1 そうでなければ 0 を出力
Step1: a[1]=2,a[2]=3,a[3]=5,a[4]=7 とおく。
Step2: i が 1 から 4 まで次を実行する。

Step2.1: r=a[i] $^{(n-1)/2}$ mod n を計算する。

Step2.1: r==2 であれば 0 を返し終了。

Step2.2: ヤコビ記号 s=J(a,n)を計算する。

Step2.3: r≠s mod n であれば 0 を返し終了。

Step3: 1 を返し終了。

（2） 乱数生成方式

・C 言語の rand 関数を用いて生成。

（3） 測定結果

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
HIME-1	934.0ms	素数生成を含む。 合成数のサイズは、256×4=1,024 ビット。 各素数は、ｐ±１法の検査のみ実装。
	785.0ms	

［暗号化処理］

（1） パディング

・なし

（2） ハッシュ関数

・SHA-1 を使用。

（3） 測定結果

［暗号化の速度］

測定対象	平均実行時間	備考
HIME-1	140.6ms	HIME-1 のデータサイズは、256 ビット。

（4） その他

・共有すべき鍵情報に相当する情報は、事務局で準備したデータファイルの先頭 256 ビットを利用している。

［復号側処理］

（3） 使用したパラメータ

・なし

（4） 測定結果

［復号の速度］

測定対象	平均実行時間	備考
HIME-1	24.7ms	HIME-1 のデータサイズは、256 ビット。

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
HIME-1	2631step	C言語（Intel C/ C++)

2.3 その他

仕様は数多くの不備、誤記、暗黙の仮定を有しており、スキームとして曖昧な点が多々ある。

(以下、記号は仕様書に従うものとする)

(a) 仕様詳細が記述されていない

1) ハッシュ関数(G_1, H_1：**仕様書3.2.6 節**)

これらのハッシュ関数(G_1, H_1)は，定数$C, C_1, \ldots, C_3$を用いて構成されているが、この定数C, C_iの選択法が規定されておらず、不適切に設定された場合、関数の出力に偏りが生じてしまう。

ハッシュ関数の定義域は$\{0,1\}^{\infty}$ではなく、$\{0,1\}^{*}$とすべきである。

H_1の定義域は$\{0,1\}^{896}$ではなく、$\{0,1\}^{382}$であろう。

(b) 仕様の表記に不整合が見受けられる

1) 鍵生成(**仕様書3.2.1 節**)

nのビット長を入力しておきながら、p, qを256 ビット固定で生成している。

(c) 記述されるアルゴリズム等に(数学的に) 誤った記述が見受けられる。

1) 素数生成(**仕様書3.3.3 節**)

MILLER-RABIN 関数では素数生成は行なえない。

入力値tの仕様が不明である。

2) Jacobi 記号(**仕様書3.3.5 節**)

ステップ3 で、JACOBI$(a/2, n)\cdot(-1)^{n^2-1/8}$はJACOBI$(a/2, n)\cdot(-1)^{(n^2-1)/8}$の誤りである。

(d) その他

1) 256 ビットの素数の積$(p^d q)$は、かならずしも1024 ビットを保証しない。

2) 復号におけるp^{-1}, q^{-1}の意味が不明である。後の記述から$p^{-1} \bmod q, q^{-1} \bmod p$と思われる。

3) 実装の記述において、$0 < m < 2^{k-2}$としているが、$0 < m < 2^{k-k_0-k_1-2}$の間違いであろう。

4) 実装の記述において$m' \geq 2^{k-2}$の場合に$m' \leftarrow pq - m'$としているが、この条件だけでは、正しくm'からmを復号できないため、注意が必要である。

5) 図1 の相関関係は間違いである。

6) 主張されている性能は、楕円曲線に基づく方式やRabin 法と比較して疑問符が残る。

参考文献

電子情報通信学会技術研究報告 ISEC2000-65(2000-09)

4.4.11 HIME-2

1 暗号技術

1.1 技術概要

HIME-2 は、1979 年に Michael O. Rabin により提案された Rabin 暗号と 1994 年に Mihir Bellare と Phillip Rogaway により提案された OAEP 方式を組み合わせた方式の変形で、2000 年に ISEC 研究会において、日立製作所の西岡玄次、佐藤尚宜、瀬戸洋一により提案された暗号技術である。

HIME-2 は($p_1 \cdots p_d : d \geq 4$ 型の)素因数分解(IF)の困難性に前提に基づき設計した守秘目的の公開鍵方式である。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

HIME-2 に関する知的所有権は、(株)日立製作所が所有している。

米国特許第 5,103,479 号“ ENCIPHER METHOD AND DECIPHER METHOD ”

特願 2000-208237 「公開鍵暗号方法および公開鍵暗号を用いた通信システム」

採用された場合には、非差別的、かつ、適正な対価条件でライセンス提供することが宣言されている。

応募暗号技術仕様の公開 Web アドレス http://www.sdl.hitachi.co.jp/crypto/

1.2 技術仕様

暗号化、復号関数の概要を示す.

［秘密鍵］

d 個の素数 p_i

［公開鍵］

$n\left(=\prod_{i=1}^{d} p_i, |n| = k\right), |n| \geq 1024, |n| = 1024$ のとき $d = 4$〜6 が推奨されている。

以下は $|n| = 1024$ でのパラメータサイズである。

［ハッシュ関数］

$$H_1 : \{0,1\}^{k_0} \rightarrow \{0,1\}^{k-k_0}, k_0 = 128$$

$$H_2 : \{0,1\}^{k-k_0} \rightarrow \{0,1\}^{k_0}$$

【暗号化】

以下のようにして、平文 $m\left(m \in \{0,1\}^{k-k_0-k_1}, k_1 = 128\right)$ を暗号化して、暗号文 c を得る。

1. r:乱数$\left(r \in \{0,1\}^{k_0}\right)$

2. $z \leftarrow (((m\|0^{k_1}) \oplus H_1(r))\|(r \oplus H_2((m\|0^{k_1}) \oplus H_1(r))))$

3. $c \leftarrow z^2 \bmod n$

【復号】

以下のようにして、暗号文cを復号して平文mを得る。

1. $z_i \leftarrow c^{\frac{p_i+1}{4}} \bmod p_i (i = 1,\cdots,d)$

2. $z_i (i = 1,\ldots,d)$から中国人の剰余定理を用いて、$c \equiv z^2 \bmod n$を満たすz'_jをすべて

(2^d個)求める。

3. $z'_j = a_j \| b_j \left(a_j \in \{0,1\}^{k-k_0}, b_j \in \{0,1\}^{k_0}, j = 1,\ldots,2^d\right)$に対して

$$m = \begin{cases} [z'_j]^{k-k_0-k_1} \text{ if } [H_1(H_2(a_j) \oplus b_j) \oplus a_j]_{k1} = 0^{k_1}, \\ \text{“reject”} \quad \text{otherwise} \end{cases}$$

としてmを出力する。

但し$[a]^k, [a]_k$は各々aの上位，および，下位kビットを表す。

2 評価結果

仕様に曖昧さがあり、スキームを一意に特定できず、第三者が実装できない。提案者は証明可能安全性を主張しているが、その正当性は確認できない。

2.1 安全性評価

仕様に曖昧さがあり、スキームを一意に特定できない。詳細評価では仕様が曖昧な部分を適宜修正して行なっているが、厳密にはその修正法の正当性は保証できていない点に注意が必要である。

(a)プリミティブの安全性

現時点で有効な攻撃法は確認されていない。

プリミティブの安全性は、法nの元での平方根求解の困難性に基づく。

OAEP に用いられる乱数は十分にランダムであり、ハッシュ関数の出力もまたランダムであることが求められる。但し、利用されるハッシュ関数などが特定されていない。またH_2の出力長は 128 ビットと規定されているため、近い将来、これのコリジョンは比較的容易に計算可能となり、注意が必要である。

(b)パラメータの安全性

$p_1 \cdots p_d (d \geq 4)$型の素因数分解の困難性に基づく。

HIME-2 の法は RSA と形式が異なるため、最小素因数のサイズに依存した実行時間となる楕円曲線法による素因数分解(ECM)も考慮して法のサイズを決定する必要がある。

秘密鍵の数を増やすことになるが、個々の素数のサイズを小さくして、処理速度を速くすることの必然性には疑問が残るとの指摘もある。

(c)パラメータサイズの安全性

仕様では鍵である法サイズが 1024 ビットの場合以外は、他のパラメータ(d, k_0, k_1)の明確な推奨値がない。1024 ビット鍵を使用した場合、現時点では十分な耐性を持つと考えられるが、15 ～20 年後の安全性は疑問であり、注意が必要である。パラメータを可変にできるよう、仕様を変更すべきである。

また、1024 ビット鍵での各種パラメータサイズの指定理由が明確でないため、その妥当性を検証することができない。

小さな素数の積を利用することは、安全面で問題があるとの指摘もある。

(d)スキームの安全性

仕様ではランダムオラクルモデルの基で、OAEP を利用して、その安全性(適応的選択暗号文攻撃に対する強秘匿性: IND-CCA2)を主張している。

まず、OAEP を適用可能とするには、プリミティブが一方向置換性を満たすことが求められる。一方向性については満足するものと思われるが、仕様には明確に記述されていない。また、置換性については明らかに満足しない。補助情報を用いて復号結果の絞り込みは可能であるが、一意に復号するまでは不可能であり、また、OAEP 部分でチェックするとしても、低い確率にせよ異なる平文が得られる可能性が存在する。

さらに、最近、OAEP だけでは IND-CCA2 を証明できないという欠陥が明らかになったため、現時点の証明では本スキームの安全性を保証できていない(証明可能になるとの指摘もある)。

2.2 SW 実装評価

鍵生成部の実装方法においては数体ふるい法を考慮し合成数nのサイズを1024ビット程度とし、nの素因数p,qの選択にあたってはp±1法を考慮ことが応募者により示されている。鍵生成において256ビットの素数p,qを生成する必要があるが、(p±1)法による素因数分解および周期性による攻撃を避けるためにp.qは以下の条件を満たすことが望ましい。

1） p+1 と p-1 はそれぞれ十分に大きな素数 p1.p2 を因数に持つ(q も同様)。

2） p1+1 と p1-1 はそれぞれ十分に大きな素数 s1,r1 を因数に持ち、p2+1 と p2-1 はそれぞれ大きな素因数 s2,r2 を因数に持つ(q も同様)。

この仕様を追加することによりソフトウェア実装評価が可能となった。

［実装仕様］

（1） 使用したパラメータの位置づけ

(ア) N=1,024 ビットの RSA 暗号（守秘同等以上の強度）

(イ) 具体的には、法 N は、$N=p_1p_2p_3p_4$（d=3）の形式であり、p_1，p_2，p_3，p_4は各々256 ビットである。

(ウ) p_1，p_2，p_3，p_4の選定にあたっては、p±1 法を考慮。

(エ) 暗号技術仕様書ではnのビット長が1023ビット以上であることおよびp_1,p_2,p_3,p_4が全て異なることは要求していなかったが、問題が生じる場合があるため今回の実装では条件の追加を行った。

（2） パラメータの設定方式

・なし

（3） 使用パラメータの検証状況

・検証の必要なし

（4） パラメータ・サイズのバリエーション

・今回の実装では、nのサイズを1023ビットまたは1024ビットとした。

（5） コンパイラの設定

① －O2 実行速度を優先した速度最適化を実行した。

② －QxK 命令コードを実行するプロセッサーをPentiumⅢ(SIMDサポート)のみに限定する。(-G6オプションと併せて対象となるプロセッサをPentiumⅢに限定した。)

③ －Zp16 構造体の境界調整を16バイト境界に設定した。

［実装上で使用した手法］

(ア) べき乗演算においてモンゴメリー乗算及び4array法を使用。

〔特徴及び測定パラメータ(1/2)〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
HIME-2	素因数分解問題	法Nは、$N=p_1p_2p_3p_4$ (d=3)の形式であり、p_1, p_2, p_3, p_4 は各々256 ビットである。 N=1,024 ビットの RSA 暗号（守秘同等以上の強度）	・p_1, p_2, p_3, p_4の選定にあたっては、p±1法を考慮。 ・暗号技術仕様書ではnのビット長が1023 ビット以上であることおよびp_1,p_2,p_3,p_4が全て異なることは要求していなかったが、問題が生じる場合があるため今回の実装では条件の追加を行った。 ・ べき乗演算においてモンゴメリー乗算及び4 array法を使用。

［システムパラメータ生成］

・今回は測定しなかった。

［鍵対生成］

【アルゴリズム1: 鍵生成】

出力：公開鍵(n,k),秘密鍵(p,q,α,β,z)

Step1: $p \equiv 3 \pmod 4$, |p|=256を満たす素数pを選択する。

Step2: $q \equiv 3 \pmod 4$ |q|=256,p≠qを満たす素数qを選択する。

Step3: k=|pq|を計算する。

Step4: $n = p^3q$ を計算する。

Step5: k=|pq|を計算する。

Step6: $\alpha = q^{-1} \pmod{p-1}$を計算する。

Step7: $\beta = p^{-1} \pmod{q-1}$を計算する。

Step8: $z = q^{-1} \pmod{p}$を計算する。

Step9: 公開鍵(n,k),秘密鍵(p,q,α,β,z)を出力する。

（1） 素数判定条件

鍵生成において 256 ビットの素数 p,q を生成する必要があるが、(p±1)法による素因数分解および周期性による攻撃を避けるために p.q は以下の条件を満たすことが望ましい。

1） p+1 と p-1 はそれぞれ十分に大きな素数 p1.p2 を因数に持つ(q も同様)。

2） p1+1 と p1-1 はそれぞれ十分に大きな素数 s1,r1 を因数に持ち、p2+1 と p2-1 はそれぞれ大きな素因数 s2,r2 を因数に持つ(q も同様)。

HIME-1 の実装では上記条件を満たす素数を次に示す方法で実現した。

p1+1 が素数 s1 を因数に持つ ⇔ $p1 \equiv -1 \pmod{s1}$ —①

p2+1 が素数 r1 を因数に持つ ⇔ $p1 \equiv 1 \pmod{r1}$ —②

s1,r1 は素数であるから

$s1 \cdot g \equiv 1 \pmod{r1}$ となる g が存在する。 $g=s1^{-1} \pmod{r1}$ —③

このとき

$p\_1=-1+2 \cdot s1 \cdot g \pmod{s1 \cdot r1}$ は①,②を満たす。 —④

p_1 が偶数となる場合はさらに、p_1=p_1+s1・r1 とする。 —④’

そこでまず素数 s1,r1 を生成し③,④,④’に適用し素数候補 A=p_1+2・k・r1・s1 とし—⑤

(k:正の整数) 次の候補を A=A+2・k・r1・s1 とする。

以上をアルゴリズムの形にまとめると

【アルゴリズム 2：p±1 法に対して強い素数生成】

入力：素数 s1,r1

出力：素数 p1

Step1: ③,④,④’,⑤に従い、素数 s1,r1 より素数の候補 A を生成する。

Step2: 3000 番目までの素数からなるテーブル{2,3,5,7,・・・prime[3000]}を作成する。なお今回の実装では予め素数テーブルを用意した。

Step3: k=1 から k=3000 までの剰余テーブルを m[k],m’[k]を作成する。

$m[k] \leftarrow A \pmod{prime[k]}$

$m'[k] \leftarrow 2 \cdot r1 \cdot s1 \pmod{prime[k]}$

Step4: $A \leftarrow A+2 \cdot r1 \cdot s1$

Step5: k=1 から k=3000 までの剰余テーブル m[k]を更新する。

m[k] ← (m[k] + m'[k]) (mod prime[k])

Step6: 全ての m[k]のうち 0 があれば Step4 にもどり、なければ Step7 に進む。

Step7: アルゴリズム 5 を用いて A が素数でないと判断されたら Step4 にもどり、素数と判定されれば Step8 に進む。

Step8: p1 ← A

素数 p(および q)の生成は上記アルゴリズムを用いて素数 s1,r1 より p1 を,素数 s2,r2 より素数 p2 を生成し,さらに素数 p1,p2 に対してアルゴリズム 2 を用いることにより行う。

【アルゴリズム 3：p±1 法に対して強い 256 ビット素数の生成】

出力：256 ビット素数 p

Step1: 55 ビット素数 s1 を生成する。

Step2: 55 ビット素数 r1 を生成する。

Step3: アルゴリズム 2 を用いて s1,r1 から 120 ビット素数 p1 を生成する。

Step4: 55 ビット素数 s2 を生成する。

Step5: 55 ビット素数 r2 を生成する。

Step6: アルゴリズム 2 を用いて s2,r2 から 120 ビット素数 p2 を生成する。

Step7: アルゴリズム 2 を用いて p1,p2 から 256 ビット素数 p を生成し p を出力する。

アルゴリズム 3 では 55 ビット素数を生成する必要があるが、その生成には次のアルゴリズム 4 を用いた。

【アルゴリズム 4：55 ビット素数生成】

出力：55 ビット素数 r

Step1: ランダムに 55 ビット整数 r を生成する。

Step2: r を 3000 番目までの素数(テーブル参照する)で試し割し r が素数を因数として持てば Step1 に戻り、そうでなければ Step3 に進む。

Step3: アルゴリズム 5 を用いて r が素数でないと判断されたら Step1 にもどり、素数と判定されれば Step4 に進む。

Step4: 素数 r を出力する。

素数判定については暗号技術仕様書では例として Miller-Rabin 法をあげているが、今回の実装においては Solovay-Strassen 法を用いた。

【アルゴリズム 5：素数判定(Solovay-Strassen 法)】

入力：素数の候補 n

出力： n が素数であれば 1 そうでなければ 0 を出力

Step1:　a[1]=2,a[2]=3,a[3]=5,a[4]=7 とおく。

Step2:　i が 1 から 4 まで次を実行する。

Step2.1: $r=a[i]^{(n-1)/2} \bmod n$ を計算する。

Step2.1: r=2 であれば 0 を返し終了。

Step2.2: ヤコビ記号 s=J(a,n)を計算する。

Step2.3: $r \neq s \bmod n$ であれば 0 を返し終了。

Step3: 1 を返し終了。

（2）　乱数生成方式

・C 言語の rand 関数を用いて生成。

（3）　測定結果

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
HIME-2	2,845.0ms 1,829.0ms	素数生成を含む。 合成数のサイズは、256×4=1,024 ビット。 各素数は、ｐ±１法の検査のみ実装。

［暗号化］

（1）　パディング

・なし

（2）　ハッシュ関数

・SHA-1 を使用。

（3）　測定結果

［暗号化の速度］

測定対象	平均実行時間	備考
HIME-2	0.4ms	HIME-2 のデータサイズは、256 ビット。

（4）　その他

・共有すべき鍵情報に相当する情報は、事務局で準備したデータファイルの先頭 256 ビットを利用している。

［復号］

（1）　使用したパラメータ

・なし

（2）　測定結果

［復号の速度］

測定対象	平均実行時間	備考
HIME-2	15.3ms	HIME-2 のデータサイズは、256 ビット。

［使用したパラメータ］

・p,q は鍵生成時に、素数を作成。

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
HIME-2	2666step	C言語（Intel C/ C++）

2.3 その他

暗号スキームの仕様としては、様々な問題が存在している。実際の利用にあたっては修正が必要であろう。

(以下、記号は仕様書に従うものとする)

(a)仕様詳細が記述されていない。

1)ハッシュ関数(G_2, H_2 :**仕様書 3.2.6 節**)

これらのハッシュ関数 (G_2, H_2) は、定数 $C, C_1, \ldots, C_7$ を用いて構成されているが、この定数 C, C_i の選択法が規定されておらず、不適切に設定された場合、関数の出力に偏りが生じてしまう。

ハッシュ関数の定義域は $\{0,1\}^{\infty}$ ではなく、$\{0,1\}^{*}$ とすべきである。

H_2 での変数 $x_1, \ldots, x_7$ のビット長が指定されていない。

2) convert**(仕様書 3.2.4 節)**

m のビット表現方法が不明確である。

(b)仕様詳細が記述されているが、意味をなしていない

1)復号アルゴリズム**(仕様書 2.2.3 節)**

φ は d 入力なのに対し、2 入力として記述されており，意味をなしていない。

$e_d x_i$ は $e_d x_d$ の誤植と考えられる。

ループの継続条件が不明である。

複数の m が得られた場合の処理が不明である。

2)復号アルゴリズム詳細**(仕様書 3.2.3 節)**

“ reject ”が出力されない。

無限ループに陥る可能性がある。

(c)仕様の表記に不整合が見受けられる

1)記号の説明**(仕様書 3.1 節)**

φ に関する記述が一貫していない。

w に関する制限 $(w \equiv x \bmod n, w \equiv y \bmod m)$ が誤っている（$w \equiv x \bmod m, w \equiv y \bmod n$ が正しいものと思われる)。

2)アルゴリズム詳細**(仕様書 3.2 節)**

φ に関する記述が一貫していない。

3)鍵生成**(仕様書 3.2.1 節)**

n のビット長を入力しておきながら、 p_i を 256 ビット固定で生成している。

random prime という言葉の使い方が適当でない。

4)暗号化**(仕様書 3.2.2 節)**

convert への入力値 R が規定されていない。

convert が 3 入力関数となっている(仕様では，2 入力関数)。

(d)記述されるアルゴリズム等に(数学的に)誤った記述が見受けられる。

1)素数生成**(仕様書 3.3.3 節)**

MILLER-RABIN 関数では素数生成は行なえない。

入力値 t の仕様が不明である。

2) Jacobi 記号**(仕様書 3.3.5 節)**

ステップ 3 で，JACOBI $(a/2,n)\cdot(-1)^{n^{2-1/8}}$ は JACOBI $(a/2,n)\cdot(-1)^{(n^2-1)/8}$ の誤りである。

3) 中国人剰余定理による環同型写像**(仕様書 3.3.6 節)**

$z = m^{-1} \bmod n$ は $z = n^{-1} \bmod m$ の誤りである。

(e)記述意図が不明

1)備考**(仕様書 2.2.4 節)**

n のサイズや d の値に関する仕様が規定されていないため、アルゴリズムとの依存関係が不明である。

補助情報 w や a を用いた探索の方法が不明である。

(f)誤植

1)最大公約数および逆元**(仕様書 3.3.4 節)**

BINARY-EUCLID (a,n) は BINARY-EUCLID (x,y) と思われる。

2) HIME-2 の安全性**(仕様書 1.1 節)**

$D_{s,k}$ の仕様が不明である。

(g)その他

1)素数に対する制約 $(p_i \equiv 3 \pmod 4)$ が方式にとって本質的なのかどうかの記述が無い。実装に関する記述などでは制約があり、また、この制約を仮定して記述されている部分もある。

2) 256 ビットの四つの素数の積は，かならずしも 1024 ビットを保証しない。

3)中間変数 z が n 以上となることがある。この場合には、復号結果はさらに複雑となる。

4) k_0, k_1 が公開鍵としてユーザー依存なのは問題である。

5) RSA 法や Rabin 法に対して、有用性が明確でない。

参考文献

電子情報通信学会技術研究報告　ISEC2000-65 (2000-09)

4.4.12 RSA-OAEP

1. 暗号技術

1.1 技術概要

＜暗号の種類＞

RSA-OAEP は 1977 年 Ronald L. Rivest, Adi Shamir, Leonard M. Adleman により提案された RSA 暗号と 1994 年 Mihir Bellare, Phillip Rogaway により提案された Optimal Asymmetric Encryption Padding(OAEP)を組み合わせた素因数分解問題(IF)に基づく守秘目的の公開鍵暗号方式である。

＜暗号の発表年、発表論文＞

(1) RSA 発表年:1977 年

R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2), pp. 120-126, February 1978.

(2)OAEP 発表年:1994 年

M. Bellare and P. Rogaway. Optimal Asymmetric Encryption - How to Encrypt with RSA. In Advances in Cryptology-Eurocrypt '94, pp. 92-111, Springer-Verlag, 1994.

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・ 関連特許なし。

応募暗号技術仕様の公開 Web アドレス http://www.rsasecurity.com/rsalabs/rsa_algorithm/

1.2. 技術仕様

【鍵生成】

[出力] 公開鍵 (e, n)、 秘密鍵 d。

1. 素数 p,q を生成する。
2. n=pq, λ(n)=LCM(p-1,q-1)を計算する。
3. $e \in Z_{\lambda(n)}$, (GCD(e,λ(n))=1)を適当に定める。
4. d = 1/e mod λ(n)を計算する。
5. 公開鍵(e, n)、 秘密鍵 d を出力する。

システムで次の 2 種類のランダム関数を定める。

$H_1:\{0,1\}^{k0} \rightarrow \{0,1\}^{n+k1}$

$H_2:\{0,1\}^{n+k1} \rightarrow \{0,1\}^{k0}$

【暗号化】

[入力] 平文 $m \in \{0,1\}^n$, 公開鍵 (e, n).

[出力] 暗号文 c.

1. 乱数 $r \in \{0,1\}^{k0}$ を生成する。

2. $s=(m||0^{k1}) \oplus H_1(r)$ とする。

3. $t=r \oplus H_2(s)$ とする。

4. $w=s||t$ とする。

5. 暗号文 $c=w^e$ を出力する。

【復号】

[入力] 暗号文 c、 秘密鍵 d。

[出力] 平文 m

1. $w=c^d$ を計算する。

2. s を w の上位 n+k1 ビットとする。

3. t を w の下位 k0 ビットとする。

4. $r= t \oplus H_2(s)$ とする。

5. $z= H_1(r) \oplus s$ とする。

6. z の下位 k1 ビットが 0^{k1} ならば、z の上位 n ビットを平文 m として出力そうでなければ、“decryption error”を出力する。

1.3. その他

IEEE P1363[1]、PKCS #1 V2.0[2]、あるいはまもなく制定されるANSI X9.44[3]に記載されている。
また、コンパチブルでは無いが、関連するメカニズムが Secure Electronic Transactions (SET) Protocol に記述されている。

[1] IEEE Std 1363-2000: Standard Specifications for Public Key Cryptography. IEEE, to appear.

[2] RSA Laboratories. PKCS #1: RSA Cryptography Standard. Version 2.0, October 1, 1998. Available from http://www.rsasecurity.com/rsalabs/pkcs/. (Republished as IETF RFC 2437, available at は ftp://ftp.isi.edu/in-notes/rfc2437.txt.)

[3] ANSI X9.44: Key Establishment Using Factoring-Based Public Key Cryptography for the Financial Services Industry. Working draft, June 2000.

2．評価結果

2.1 安全性評価

RSA 暗号化関数の一方向性、およびランダムオラクルモデルを仮定することにより適応的選択暗号文攻撃に対して強秘匿であることが示される。ただし、RSA 暗号化関数が一方向性を満たすためにはパラメータを適切に設定する必要がある。なお一般的な OAEP 変換は、適応的選択文書攻撃に対して強秘匿であるという性質を示すために十分でなかったが、RSA-OAEP については、RSA 暗号化関数の部分一方向性(RSA 暗号化関数の場合一方向性と同値)を仮定することにより、適応的選択文書攻撃に対して強秘匿であることが最近、確認された。

2.2．ソフトウェア実装評価

［実装仕様］

（1）使用したパラメータの位置づけ

- 現在、主流となっている法の長さが 1,024 ビット。
- 公開鍵モジュロ値長　1024 ビット
- 公開指数　F4（65537）

（2）パラメータの設定方式

- 法（モジュロ）Ｎの値（＝公開鍵）及び秘密鍵を素数から生成。
- モジュロ値の長さの変更は 1024 ビットから 2048 ビットの範囲内で指定可能である。評価プログラムでは、鍵生成を行う関数をコールする直前のソース上で直接長さを指定している。

（3）使用パラメータの検証状況

- 評価プログラムでは鍵生成に用いられる素数の判定に Miller-Rabin の確率的素数判定法を使っており、素数の“強度”つまり生成された数が確率の高さの観点から本当に強いかどうかのチェックは行っていない。

（4）パラメータ・サイズのバリエーション

- モジュロ値 N の長さは 1024 ビットから 2048 ビットまでの範囲で指定可能である。
- 公開指数 e は 17 か 65537 を指定可能である。
- いずれもソース上に直接記述する。(公開指数用パラメータは 1(e=65537)あるいは 0(e=17)を指定)

［実装上で使用した手法］

- 中国人剰余定理(CRT)を使用しモジュロ演算の高速化を図った。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置付け	その他
RSA OAEP	素因数分解問題	公開鍵モジュロ値長　1024 ビット	中国人剰余定理(CRT)を使用し、高速化。 公開指数：65537

［システムパラメータ生成］

（1）測定結果

測定対象	測定結果	備考
RSA-OAEP	1.104ms	e=17
	1.107ms	e=65537

［鍵対生成］

（1）素数判定条件

評価プログラムでは鍵生成に用いられる素数の判定に Miller-Rabin の確率的素数判定法を使っており、素数の“強度”つまり生成された数が確率の高さの観点から本当に強いかどうかのチェックは行っていない。

（2）乱数生成方式

乱数の種は、測定用メインルーチンで直接指定している。素数生成のもととなる乱数生成には SHA-1 を使用している。他のアルゴリズムはサポートしていない。

（3）測定結果〔鍵対生成の速度〕

測定対象	平均実行時間	備考
RSA OAEP	2,946.5ms	e=17
	3,405.5ms	e=65537

（4）その他

実用において、ソフトウェアで乱数の種を生成する場合には、セキュリティ上はある程度の時間と複数の判定処理を合わせて、種の生成を行う必要がある。

［暗号化］

（1）パディング

EME-OAEP パディング

（2）ハッシュ関数

・ SHA-1

（3）測定結果［暗号化の速度］

測定対象	平均実行時間	備考
RSA OAEP	4.2ms	e=17
	4.2ms	e=65537

［復号］

（1）測定結果［復号の速度］

測定対象	平均実行時間	備考
RSA OAEP	84.2ms	e=17
	84.2ms	e=65537

［使用したパラメータ］

・ 公開指数 F4（65537）

［コードサイズ（参考値）］

測定対象	コードサイズ	備考
RSA OAEP	62,413byte	性能測定用実行形式テストプログラムのサイズ

4.4.13 ACE Encrypt

1 暗号技術

1.1 技術概要

ACE Encrypt は1998 年に提案されたRonald Cramer とVictor Shoup によりCrypto'98 に提案された暗号方式の特殊な変形であり、2000年にIBM Zürich 研究所のThomas Schweinberger とVictor Shoup により manuscript として発表され、日本IBM から提案された。ACE Encrypt は(素体上の) 離散対数問題(DLP) の困難性に基づき、守秘を実現する公開鍵方式である。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

ACE Encrypt に関する知的所有権は、IBM が所有している。

Ronald Cramer, Victor Shoup,“Practical non-malleable public-key cryptosystem”Filed February 16, 1999.

(ISO に) 採用された場合には、非差別的、かつ、適正な対価条件でライセンス提供することが宣言されている。

1.2 技術仕様

暗号化、復号関数の概要を示す。

［秘密鍵］

$x, x_1, x_2 x_3, x_4, \in Z_q$

［公開鍵］

$p, q, g_1, g_{2,} y_1, y_2, y_3, y_4$

q : 素数 ($|q| = 256$)

p : 素数 ($p \equiv 1 \pmod{q}$)

$g_1 : (g_1^q \equiv 1 \pmod{p}, g_1 \in Z_p^*, g_1 \neq 1)$

$g_2 : (g_2 = g_1^x \bmod p)$

$y_{1:} : (y_1 = g_1^{x_1} \bmod p)$

$y_2 : (y_2 = g_1^{x_2} \bmod p)$

$y_3 : (y_3 = g_1^{x_3} \bmod p)$

$y_4 : (y_4 = g_1^{x_4} \bmod p)$

［ハッシュ関数］

$H_1 : \{0,1\}^* \rightarrow \{0,1\}^{160}$

$H_2 : \{0,1\}^* \rightarrow \{0,1\}^{256}$

【暗号化】

以下のようにして、平文mを暗号化して、暗号文cを得る(特に、指示のない限り演算は法pの元で行なう)。

1. r':乱数$(r \in \{0,1\}^{128}), r$:乱数$(r \in Z_q)$

2. $u_1 \leftarrow g_1^r, u_2 \leftarrow g_2^r$

3. $h \leftarrow H_1(r', u_1, u_2)$

4. $v \leftarrow y_1^r y_2^{hr}$

5. $\tilde{y}_3 \leftarrow y_3^r, \tilde{y}_4 \leftarrow y_4^r$

6. $k \leftarrow H_2(r', u_1, \tilde{y}_3, \tilde{y}_4)$

7. $z \leftarrow E(k, m)$

8. $c \leftarrow r' \| u_1 \| u_2 \| v \| z$

但し、$E(k,m)$はMARS の累積/ カウンタ・モードを利用したメッセージmに対する鍵kによる暗号化、および、MAC 生成を表す。

【復号】

以下のようにして、暗号文cを復号して平文mを得る。

1. $r' \| u_1 \| u_2 \| v \| z \leftarrow c$

2. $h \leftarrow H_1(r', u_1, u_2)$

3. $v \neq u_1^{x_1 + hx_2}$ の場合は、“ reject ”を出力して終了する。

4. $\tilde{y}_3 \leftarrow u_1^{x_3}, \tilde{y}_4 \leftarrow u_2^{x_4}$

5. $k \leftarrow H_2(r', u_1, \tilde{y}_3, \tilde{y}_4)$

6. $m \leftarrow D(k, z)$

但し、$D(k,z)$はMARS の累積/ カウンタ・モードを利用した暗号文zに対する鍵kによる復号、および、MAC 検証を表す。

2 評価結果

2.1 安全性評価

(a) プリミティブの安全性

現時点で有効な攻撃法は確認されていない。

プリミティブの安全性は、法 p の元での離散対数問題の困難性に基づく。

利用されるハッシュ関数が汎用一方向性(より正確には第二プレイメージ衝突耐性)を満たし、また、共通鍵暗号は累積/ カウンタ・モードで使用した場合に疑似ランダム性を有することが必要である。暗号化の際に利用される乱数は十分にランダムであることが求められる。

(b) パラメータの安全性

素体上の離散対数問題の困難性に基づく。仕様に記述がない特殊なタイプ(Gordon 法が適用可能となるような素数) を鍵として利用することは避けるべきである。

(c) スキームの安全性

Standard Model (実用に近い状況下) で安全性を証明しており、決定DH 問題(DDH)、汎用一方向性ハッシュ関数に関する仮定(SHA-1 第二プレイメージ衝突耐性)、共通鍵暗号の擬似ランダム性に関する仮定(MARS 累積/ カウンタ・モード疑似乱数性) に帰着されている(但し、決定DH 問題(DDH)は、計算DH 問題(CDH)や離散対数問題よりも特殊な仮定である点に注意すること)。

本スキームでは、ハッシュ関数SHA-1 から汎用一方向性ハッシュ関数を構成しており、SHA-1 と共通鍵暗号MARS を累積/ カウンタ・モードで利用することで、共通鍵暗号プリミティブを実現している。そのため、SHA-1 の第二プレイメージ衝突耐性に関する仮定とMARS の累積/ カウンタ・モード擬似乱数性に関する仮定が必要となる。

本スキームの安全性をランダムオラクルモデルで評価した場合には、計算DH 問題に依存することになる。

2.2 SW 実装評価

ACE Encrypt に関しては、応募された暗号技術ではあったが、応募書類提出後に仕様が変更されこともあり、応募された技術に対応したソフトウェア実装評価を行わなかった。

2.3 その他

本スキームは、非常に巧妙に構築されているために、柔軟性がない。例えば、ハッシュ関数を単純なSHA-1 に置き換えた場合には、スキームの安全性に問題が生じる。また、共通鍵暗号プリミティブ(MARS) の変更は可能(但し、安全性は不明)だが、メッセージ認証子部分を削除した場合は安全性が損なわれる。

仕様書では、 m が p ビット長、共通鍵暗号のブロック長(バイト数) の二通りの意味で利用されており注意が必要である。

共通鍵暗号のための鍵を生成するためにハッシュ関数 H_2 に入力されるパラメータが、仕様書(h_1, h_2 の二変数) と元となった論文(h だけの一変数) と異なっているが、この変更は鍵のランダム性のために重要であり、一変数とすると安全性の証明に不備が生じる。

鍵サイズや計算量の面で不利な点があるが、本スキームを楕円曲線上で実現することも可能であろう。

4.4.14 ECAES in SEC1

1. 暗号技術

1. 1 技術概説

1999 年に、SECG(Standards for Efficient Cryptography 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。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・ 4,745,568: Computational method and apparatus for finite field multiplication, issued May 17, 1988. This patent includes methods for efficient implementation of finite field arithmetic using a normal basis representation.

・ 5,761,305: Key Agreement and Transport Protocol with Implicit Signatures, issued June 2, 1998. This patent includes versions of the MQV protocols.

・ 5,787,028: Multiple Bit Multiplier, issued July 28, 1998.

・ 5,889,865: Key Agreement and Transport Protocol with Implicit Signatures, issued March 30, 1999. This patent includes versions of the MQV protocols.

・ 5,896,455: Key Agreement and Transport Protocol with Implicit Signatures, issued April 20, 1999. This patent includes versions of the MQV protocols.

上記提案者所有の特許および著作権は、提案者により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される。

なお、詳細は、特許所有者、及び SECG のウェブサイト（www.secg.org）を参照のこと。

http://www.secg.org/patent_policy.htm

http://www.secg.org/collateral/certicom_secg_patent.pdf

特許所有者が、特許技術の使用許諾を求める申請者に対し、妥当かつ差別待遇のない取引条件で、使用を許諾することに同意する文書を、SECG に提出済みかどうかを確認すること。

応募暗号技術仕様の公開 Web アドレス http://www.labs.fujitsu.com/theme/crypto/public_key.html

http://www.secg.org/drafts.htm

1.2 技術仕様

楕円曲線パラメータ (p, a, b, G, l)

素体 F_p 上の楕円曲線 $y^2 = x^3+ax+b$ に素数 l を位数としてもつ有理点 G があることを示す。

楕円曲線パラメータ (k, f, a, b, G, l)

既約式 f で定義される標数 2 の k 次拡大体 F_2^k 上の楕円曲線 $y^2+xy = x^3+ax^2+b$ に素数 l を位数としてもつ有理点 G があることを示す。

公開鍵

楕円曲線パラメータ(p,a,b,G,l)または(k,f,a,b,G,l)、点 G の整数倍である点 U

秘密鍵

U=d・G となる整数 d

【暗号化】

以下のようにして、平文 m を暗号化して暗号文 c を得る。

1. r ← 1 以上 l-1 以下の乱数, R ← r・G
2. z ← x(r・U)
3. K=EK||MK ← Hash(z)
4. 平文 m を共通鍵暗号 E によって鍵 EK で暗号化:
   em ← E(EK, m)
5. 認証子生成関数 MAC.Gen により鍵 MK で暗号文 em の認証子 D を生成:
   D ← MAC.Gen(MK, em)
6. c ← R||em||D

【復号】

以下のようにして、暗号文 c を復号して平文 m を得る。

1. R||em||D ← c
2. z ← x(d・R)
3. K=EK||MK ← Hash(z)
4. 鍵 MK を用いて認証子検証関数 MAC.Ver により認証子 D を検証:
   v ← MAC.Ver(MK, em, D)
5. v が ' invalid ' なら ' invalid ' を出力して停止。v が ' valid ' ならば共通鍵暗号 E によって鍵 EK を用いて em を復号:
   m ← E(EK, em)

ただし、上で、楕円曲線上の点 P に対して、x(P)は P のｘ座標を表す。

1.3 その他

ECAES in SEC1 は IEEE P1363 に記載されている。

2. 評価結果

2.1 安全性評価

2.1.1 暗号プリミティブの安全性

ECAES in SEC1 は楕円曲線上の DH 問題の困難性に基づく公開鍵暗号である。DH 問題は離散対数問題を解かなければ解けないと一般に強く信じられている。よって、ECAES in SEC1 の暗号プリミティブとしての安全性は、楕円曲線上の離散対数問題の安全性といってよい。

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、ECAES in SEC1 においては、実用上有効な各種のビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC2 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からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC2 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。

2.1.2 スキームの安全性

ECAES in SEC1 は、以下の仮定１のもとで、能動的な攻撃者による適応的選択暗号文攻撃に対する安全性が証明されている。

仮定１ 技術概要と同じ記号を用いる。 DH(U,R)で U と R に対する DH 問題の解を表す。(G,U,R’)に対して x(DH(U,R’))を返すオラクルを DH オラクルと呼ぶ。このとき、どのような攻撃者も分布{(G,U,R,z) | z = Hash(DH(U,R)の x 座標)}と分布{(G,U,R,r) | r:乱数}を, DH オラクルへの問い合わせを許しても、事実上区別できない。

仮定１はつまり、「かくかくの攻撃者は存在しない」という仮定であり、安全性証明の仮定として安易である。実際、多くの暗号専門家が仮定１の正当性を問題にしており、なかには、仮定１のもとでの安全性証明は、公開鍵暗号の安全性証明として何もしていないのと同じであると主張する意見もある。

しかし、詳細評価者によって、ランダムオラクルモデルのもとで ECAES in SEC1 の安全性が実は、楕円曲線上の GAP DH 問題に帰着することが示された（GAP DH 問題とは、DDH 問題が解けるアルゴリズムをもっていると仮定した上で DH 問題を解く、という問題である。）よって、ECAES in SEC1 の暗号スキームとしての安全性には問題がないと考えられる。ただし、GAP DH 問題は比較的新しい問題なので、継続的な安全性評価が必要である。

2.2 ソフトウェア実装評価

［実装仕様］

（1）使用したパラメータの位置づけ

・ 以下の 2 種類の曲線を使用。

secp160r1 ： 160 ビット素体楕円曲線パラメータ

sect163r2 ： 163 ビット標数 2 の体楕円曲線パラメータ

・ どちらも SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。

（2）パラメータの設定方式

・ 楕円曲線パラメータをファイルから読み込んで使用。

・ パラメータには依存しない実装を行っている。

（3）使用パラメータの検証状況

· 楕円曲線の 2 つの係数が、(コントロールすることのできない)SHA-1 の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。

· X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。

· 各種特殊攻撃を適用できないことも確かめられている。

（4）パラメータサイズのバリエーション

・ 固定パラメータだけではなく、任意の楕円曲線パラメータの利用可能。

・ 素体楕円曲線パラメータで扱えるビット長は以下の通り。

112,128,160,192,224,256,384,521

・ 標数 2 の体楕円曲線パラメータで扱えるビット長は以下の通り。

113,131,163,193,233,239,283,409,571

［実装上で使用した手法］

・ ライブラリは、a=0，a=-3 などの特殊パラメータに限定した高速手法は使用していない。

・ 任意のパラメータに適用できる高速手法を使用。

・ パラメータには依存しない実装を行っている。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
ECAES in SEC1 160bit 素体	楕円曲線離散対数問題	RSA1024bit 相当	・ SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。 ・ 楕円曲線の２つの係数が、(コントロールすることのできない)SHA-1の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。 ・ X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。 ・ 各種特殊攻撃を適用できないことも確かめられている。 ・ 任意のパラメータに適用できる高速手法を使用。 ・ パラメータには依存しない実装を行っている。
ECAES in SEC1 163bit 標数 2 の体		RSA1024bit 相当	

［システムパラメータ生成］

・ 楕円曲線パラメータ生成処理は、確率的アルゴリズムであるため、正確な予想処理時間を出すことは困難。

・ 応募者の実験では、160 ビット楕円曲線の場合、PentiumIII 700MHz にて、平均 10 分以内。

測定結果

測定対象	平均実行時間	備考
ECAES 160bit 素体	6.0 min	18 分間に 3 本の楕円曲線パラメータを生成

［初期化/終了処理］

測定結果

測定対象		平均実行時間	備考
ECAES 160bit 素体	PC/PF	0.004 ms	なし
	PI	13.525 ms	なし
	PV	60.438 ms	なし
	WC/WF	0.032 ms	なし
ECAES 163bit 標数 2 の体	PC/PF	0.004 ms	なし
	PI	15.283 ms	なし
	PV	43.917 ms	なし
	WC/WF	0.032 ms	なし

PC/PF :楕円曲線パラメータ領域の獲得/解放処理。

PI :ファイルから読んだパラメータを領域に設定。

PV :パラメータ検証処理。パラメータが与えられた最初の時にのみ必要。

生成元が曲線上の点か、点を位数倍すると無限遠点となるかなど、正当性をチェック。

WC/WF :作業領域獲得/解放処理。

［鍵対生成］

（1）素数判定条件

・ 方式上、不要。

（2）乱数生成方式

・ 富士通オリジナル擬似乱数生成アルゴリズム。DES/SHA-1 などを使用。

（3）測定結果

測定対象	平均実行時間	備考
ECAES 160bit 素体	1.932 ms	鍵対生成
	5.820 ms	鍵対の検証
ECAES163bit 標数 2 の体	3.150 ms	鍵対生成
	8.010 ms	鍵対の検証

・ 鍵対の検証処理は、自分で鍵を生成したときには不必要。

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
ECAES in SEC1 160bit 素体	1.9ms	鍵対生成
	5.8ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
ECAES in SEC1 163bit 標数 2 の体	3.2ms	鍵対生成
	8.0ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。

［暗号化］

（1）パディング

・ 使用せず。

（2）ハッシュ関数

・ 使用せず。

（3）測定結果

［暗号化の速度］

測定対象	平均実行時間	備考
ECAES in SEC1 160bit 素体	8.5ms	なし
ECAES in SEC1 163bit 標数 2 の体	12.5ms	なし

（4）その他

KDF : ANSI-X9.63-KDF with SHA-1

MAC : HMAC-SHA-1-160 with 20 octet or 160 bit keys

ENC : XOR encryption scheme

鍵共有プリミティブ : standard elliptic curve Diffie-Hellman primitive

［復号］

（1） 測定結果

［復号の速度］

測定対象	平均実行時間	備考
ECAES in SEC1 160bit 素体	5.4ms	なし
ECAES in SEC1 163bit 標数 2 の体	8.7ms	なし

［コードサイズ（参考値）］

測定対象	コードサイズ	備考
ECAES in SEC1	356,352byte	性能測定用実行形式テストプログラムのサイズ

4.4.15 PSEC-1

1. 暗号技術

1.1 技術概要

PSEC-1は楕円離散対数問題（ECDLP）に基づく守秘目的の公開鍵暗号方式である。NTTにより提案され、楕円ElGamal暗号をプリミティブとして文献[FO1]の手法により変換された暗号方式である。

文献[FO1] Fujisaki,E. and Okamoto,T.:How to Enhance the Security of Public-Key Encryption at Minimum Cost,Proc.of PKC'99,Springer-Verlag,LNCS 1560,pp.53--68 (1999).

知的財産権 (応募者資料による)

(提案者特許とその扱い)

PSEC-1 , PSEC-2 , PSEC-3に関し，以下の2件の特許が，応募者により出願中である.

(1) 出願番号:10-320172 ランダム関数利用公開鍵暗号の暗号装置、復号装置

(2) 出願番号:2000-32461 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

非排他的かつ妥当な条件で他者に実施許諾としている。

その他、応募者によると、関連する他社特許の訴求はないものとしている。

応募暗号技術仕様の公開Webアドレス http://info.isl.ntt.co.jp/

1.2 技術仕様

PSEC-1 の技術仕様概要を述べる。

なお、PSEC-1 は拡大体の上でも構成できるが、ここでは素体上で説明する。

【鍵生成】

入力：セキュリティパラメータ k

出力：公開鍵 $\{F_p, a, b, l, G, U, HID, lLEN, mLEN, rLEN, pLEN\}$と秘密鍵$d$の対

楕円曲線を決定する。なお、ベースポイントGの位数lのビット数をセキュリティパラメータとする。

次のように秘密鍵と公開鍵を求める。$U = dG$

なお、用いるハッシュ関数のＩＤをHIDとする。

【暗号】
入力：$mLEN$の長さのメッセージm,公開鍵 $\{F_p, a, b, l, G, U, HID, lLEN, mLEN, rLEN, pLEN\}$
出力：暗号文$c1, c2$
(1)$rLEN$の長さの乱数rを発生
(2)$h = H(m \| r)$
(3)$Q = hU, c1 = hG$
(4)$c2 = (m \| r) \oplus B[xQ]$
ただし、は*IEEE*1363に述べているパディングを施したもの。

【復号】
入力：暗号文$c1, c2$, 公開鍵 $\{F_p, a, b, l, G, U, HID, lLEN, mLEN, rLEN, pLEN\}$および,
秘密鍵d
出力：$mLEN$の長さのメッセージm
(1)$Q' = dc1$
(2)$u = c2 \oplus B[xQ']$として、u'をuの下位$mLEN + rLEN$ビットとする。
(3)$h' = H(u')$として、$c1 = h'G$が成り立つかを確認する。
(4)成り立てば、u'のうち上位$mLEN$ビットを出力のメッセージmとする。

2. 評価結果

2.1 安全性評価

＜プリミティブについて＞

プリミティブの安全性は ECDLP に帰着する。楕円曲線の決め方や具体的なパラメータについては、IEEE1363 を引用しているのみであり詳細な記載はない。
しかし、IEEE1363 の楕円曲線生成法の条件判定は現時点で不十分である。

IEEE1363 の楕円曲線生成の条件判定は、Pollard ρアルゴリズム、Pohlig-Hellman アルゴリズムに対して、安全になるための「楕円曲線の位数が almost prime であること」のみである。そのため、その他の MOV 帰着、 FR 帰着攻撃に対しての条件や、 SSSA 攻撃に対しての条件、および標数 2 などの拡大体上の楕円曲線の場合の Weil Decent 攻撃に対する判定などの判定は行っていない。

＜スキームについて＞

PSEC-1 はプリミティブ暗号化関数の問題で、証明可能安全性は有していないとの指摘がある。また応募暗号技術の推奨パラメータ:

k = | p | = 160、mLen=128、rLen=32

に対して、証明可能安全性が成り立つためには、条件

rLen ≧ 140

が必要との指摘がある。この指摘が正しい場合、

$$mLen + rLen \leqq k$$

であるため、

$$mLen \leqq 20$$

となり、1回の暗号処理で安全に扱える平文の長さが著しく制限されることになる。ただし、これらの指摘の妥当性について本評価では結論を得るには至っていないため、パラメータ選択について更なる検討が必要である。

その理由を以下に示す。

PSEC-1が破れる確率εは、楕円Diffie-Hellman部分決定問題が破れる確率をε'とすると、次のように近似できる。

$$\varepsilon \approx \varepsilon' + \frac{2 \cdot q_H}{2^{rLEN}}$$

160ビットの楕円曲線を用いながら、$\varepsilon \approx \varepsilon' \approx 2^{-80}$程度になるためには、ハッシュ関数を用いる回数$q_H$が多項式時間のオーダであるとしても、rLENを80ビット以上にとる必要がある。

なお、rLENを140程度とる必要があると指摘しているが、この真偽についてはさらに検討が必要である。

2.2 SW実装評価

［実装仕様］

（1）使用したパラメータの位置づけ

- 合成数の長さ1,024ビットのRSA暗号相当の強度を持たせるため、PSEC-1の各パラメータは、160ビット。
- PSEC-1のデータサイズは、128ビット。
- 暗号化処理の対象となる1ブロック分のデータを128ビット（16バイト）とし、あらかじめファイルとして準備しておくことにする。
- 準備したファイルのサイズが16バイトより大きい場合は、ファイルの先頭から1ブロック分を切り出して暗号化処理を行うこととする。

（2）パラメータの設定方式

- 乱数の種をファイルとして読み込む。

（3）使用パラメータの検証状況

- 本暗号方式にはシステムパラメータ（スキーム処理に必要な、鍵以外のパラメータ）として楕円曲線のパラメータが必要であるが、これはすでに他の手法で計算された結果あるいはNIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどを後述するファイル形式でメインプログラムに与えるものとし、生成は行わない。

［実装上で使用した手法］

- フリーの多倍長演算ライブラリGNU MP 3.1.1を使用。

・ 仕様に記述された通りにC言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
PSEC-1	楕円曲線離散対数問題	PSEC-1 の各パラメータは、160 ビット。 RSA1024bit 相当	評価用の PSEC-1 のデータサイズは、128 ビット。 楕円曲線のパラメータが必要。他の手法で計算された結果あるいは NIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどメインプログラムに与える。 フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。

［システムパラメータ生成］

・ システム生成は、事前に済ませており、今回の評価では実施しなかった。

［鍵対生成］

・ 乱数生成方式:まず GNU MP 3.1.1 の関数 gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムは GNU MP のデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数 mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
PSEC-1	11.3ms	なし

［暗号化］

・ パディング:不要

・ ハッシュ関数:SHA-1 を使用

［暗号化の速度］

測定対象	平均実行時間	備考
PSEC-1	24.0ms	なし

［復号の速度］

測定対象	平均実行時間	備考
PSEC-1	24.1ms	なし

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
PSEC-1	189,334byte	C言語（Intel C/ C++）ソースコードサイズ

4.4.16 PSEC-2

1. 暗号技術

1.1 技術概要

PSEC-2 は楕円離散対数問題（ECDLP）に基づく守秘目的の公開鍵暗号方式である。NTTにより提案され、楕円ElGamal暗号をプリミティブとして文献[FO2]の手法により変換された暗号方式である。

文献[FO2] Fujisaki,E. and Okamoto,T.:Secure Integration of Asymmetric and Symmetric Encryption Schemes,Proc.of Crypto'99,Springer-Verlag,LNCS 1666,pp.535--554 (1999).

知的財産権 （応募者資料による）

(提案者特許とその扱い)

PSEC-1 , PSEC-2 , PSEC-3に関し、以下の2件の特許が、応募者により出願中である。非排他的かつ妥当な条件で他者に実施許諾としている。

(1) 出願番号:10-320172 ランダム関数利用公開鍵暗号の暗号装置、復号装置

(2) 出願番号:2000-32461 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

応募者によると、関連する他社特許の訴求はないものとしている。

応募暗号技術仕様の公開 Web アドレス http://info.isl.ntt.co.jp/

1.2 技術仕様

PSEC-2 の技術仕様概要を述べる。

なお、PSEC-2 は拡大体の上でも構成できるが，ここでは素体上で説明する。

【暗号】

入力：$mLEN$の長さのメッセージm,公開鍵 $\{F_p, a, b, l, G, U, HID, lLEN, mLEN, rLEN, pLEN\}$

出力：暗号文$c1, c2, c3$

(1)$rLEN$の長さの乱数rを発生

(2)$h1 = H1(r \| m), h2 = H2(r)$

(3)$Q = h1U, c1 = h1G$

(4)$c2 = r \oplus B[xQ]$

(5)$c3 = SymEnc(h2, m)$

【復号】
入力：暗号文$c1,c2,c3$, 公開鍵 $\{F_p,a,b,l,G,U,HID,lLEN,mLEN,rLEN,pLEN\}$および、
秘密鍵d
出力：$mLEN$の長さのメッセージm
(1)$Q' = dc1$
(2)$u = c2 \oplus B[xQ']$として、r'をuの下位$rLEN$ビットとする。
(3)$h1' = H1(r' \| m)$、$h2' = H2(r')$として、$c1 = h1'G$が成り立つかを確認する。
(4)成り立てば、$SymDec(h2',c3)$を復号されたメッセージとして出力する。

2. 評価結果

2.1 安全性評価

＜プリミティブについて＞

プリミティブは PSEC-1 と共通である。安全性も同様。

＜スキームについて＞

PSEC-2 は、楕円 Diffie-Hellman 問題の計算困難性を仮定し、受動的攻撃に対して安全な共通鍵暗号を用いると、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強秘匿であるという証明可能安全性を有している。ただし、その証明で十分な安全性を導くためには、条件：

hLen << k-1 であってはいけない

rLen は qLen に十分近い

を満たさなければならないとの指摘がある。提案者の仕様では、

hLen ≦ k、rLen ≦ qLen

を規定しているだけである。ただし、これらの指摘の妥当性について本評価では結論を得るには至っていないため、パラメータ選択について更なる検討が必要である。

2.2 SW 実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

- 合成数の長さ 1,024 ビットの RSA 暗号相当の強度を持たせるため、PSEC-2 の各パラメータは、160 ビット。
- PSEC-2 のデータサイズは、128 ビット。
- 暗号化処理の対象となる 1 ブロック分のデータを 128 ビット（16 バイト）とし、あらかじめファイルとして準備しておくことにする。
- 準備したファイルのサイズが 16 バイトより大きい場合は、ファイルの先頭から 1 ブロック分を切り出して暗号化処理を行うこととする。

（２）パラメータの設定方式

・ 乱数の種をファイルとして読み込む。

（3）使用パラメータの検証状況

・ 本暗号方式にはシステムパラメータ（スキーム処理に必要な、鍵以外のパラメータ）として楕円曲線のパラメータが必要であるが、これはすでに他の手法で計算された結果あるいはNIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどを後述するファイル形式でメインプログラムに与えるものとし、生成は行わない。

［実装上で使用した手法］

・ フリーの多倍長演算ライブラリGNU MP 3.1.1を使用。

・ 仕様に記述された通りにC言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
PSEC-2	楕円曲線離散対数問題	PSEC-2 の各パラメータは、160 ビット。RSA1024bit 相当	評価用の PSEC-2 のデータサイズは、128 ビット。楕円曲線のパラメータが必要。他の手法で計算された結果あるいは NIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどメインプログラムに与える。 フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。

［システムパラメータ生成］

・ システム生成は、事前に済ませており、今回の評価では実施しなかった。

［鍵対生成］

・ 乱数生成方式:まずGNU MP 3.1.1の関数gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムはGNU MPのデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
PSEC-2	11.8ms	なし

［暗号化］

・ パディング:不要

・ ハッシュ関数:SHA-1を使用

［暗号化の速度］

測定対象	平均実行時間	備考
PSEC-2	24.2ms	なし

［復号の速度］

測定対象	平均実行時間	備考
PSEC-2	24.6ms	なし

［コードサイズ（参考値）］

測定対象	コードサイズ	備考
PSEC-2	202,333byte	C言語（Intel C/ C++）ソースコードサイズ

4.4.17 PSEC-3

1. 暗号技術

1.1 技術概要

PSEC-3 は楕円離散対数問題（ECDLP）に基づく守秘目的の公開鍵暗号方式である。NTTにより提案され、楕円ElGamal暗号をプリミティブとして文献[OP]の手法により変換された暗号方式である。

文献[OP] Okamoto,T. and Pointcheval,D.:OCAC:an Optimal Conversion for Asymmetric Cryptosystems, manuscript (2000)

知的財産権 (応募者資料による)

(提案者特許とその扱い)

PSEC-1 , PSEC-2 , PSEC-3に関し、以下の2件の特許が、応募者により出願中である。非排他的かつ妥当な条件で他者に実施許諾としている。

(1) 出願番号:10-320172 ランダム関数利用公開鍵暗号の暗号装置、復号装置

(2) 出願番号:2000-32461 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

その他、応募者によると、関連する他社特許の訴求はないものとしている。

応募暗号技術仕様の公開 Web アドレス http://info.isl.ntt.co.jp/

1.2 技術仕様

PSEC-3 の技術仕様概要を述べる。

なお、PSEC-3 は拡大体の上でも構成できるが、ここでは素体上で説明する。

【鍵生成】

入力：セキュリティパラメータ k

出力：公開鍵 $\{F_p, a, b, l, G, U, H1ID, H2ID, SEID, lLEN, H1LEN, H2LEN, rLEN, pLEN\}$と秘密鍵$d$の対

楕円曲線を決定する。なお、ベースポイントGの位数lのビット数をセキュリティパラメータとする。

次のように秘密鍵と公開鍵を求める。

$U = dG$

なおハッシュ関数$H1$のIDを$H1ID$、ハッシュ関数$H2$のIDを$H2ID$、秘密鍵暗号$SymEnc$のIDを$SEID$とする。

【暗号】
入力：$mLEN$の長さのメッセージm,公開鍵 $\{F_p, a, b, l, G, U, H1ID, H2ID, SEID, lLEN, H1LEN, H2LEN, rLEN, pLEN\}$
出力：暗号文$c1, c2, c3, c4$
(1)$pLEN$ビットの乱数uと乱数rを発生
(2)$Q = rU, c1 = rG$
(3)$c2 = u \oplus B[xQ]$
(4)$c3 = SymEnc(H2(u), m)$
(5)$c4 = H1(xc1 \| yc1 \| c2 \| c3 \| u \| m)$

【復号】
入力：暗号文$c1, c2, c3, c4$, 公開鍵 $\{F_p, a, b, l, G, U, H1ID, H2ID, SEID, lLEN, H1LEN, H2LEN, rLEN, pLEN\}$ および、秘密鍵d
出力：$mLEN$の長さのメッセージm
(1)$Q' = dc1$
(2)$u' = c2 \oplus B[xQ']$とする
(3)$m' = SymDec(H2(u'), c3)$
(4)$c4 = H1(xc1 \| yc1 \| c2 \| c3 \| u' \| m')$が成立するかを確認する。
(5)成り立てば、m'を復号されたメッセージとして出力する。

2. 評価結果

2.1 安全性評価

＜プリミティブについて＞

プリミティブは PSEC-1 と共通である。安全性も同様。

＜スキームについて＞

PSEC-3 は、楕円 Gap-Diffie-Hellman 問題の計算困難性を仮定し、受動的攻撃に対して安全な共通鍵暗号を用いると、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強秘匿をもつという証明可能安全性を有している。ただし、楕円 Gap-Diffie-Hellman 問題自身は、提案されて間もないこともあり、この安全性については今後検討が必要である。
また、その証明で十分な安全性を導くためには、パラメータ、u や r のビットサイズを明確に指定しなければならないとの指摘がある。ただし、これらの指摘の妥当性について本評価では結論を得るには至っていないため、パラメータ選択について更なる検討が必要である。

2.2 SW 実装評価

［実装仕様］

（1）使用したパラメータの位置づけ

- 合成数の長さ 1,024 ビットの RSA 暗号相当の強度を持たせるため、PSEC-3 の各パラメータは、160 ビット。

- PSEC-3 のデータサイズは、128 ビット。
- 暗号化処理の対象となる 1 ブロック分のデータを 128 ビット（16 バイト）とし、あらかじめファイルとして準備しておくことにする。
- 準備したファイルのサイズが 16 バイトより大きい場合は、ファイルの先頭から 1 ブロック分を切り出して暗号化処理を行うこととする。

（2）パラメータの設定方式

- 乱数の種をファイルとして読み込む。

（3）使用パラメータの検証状況

- 本暗号方式にはシステムパラメータ（スキーム処理に必要な、鍵以外のパラメータ）として楕円曲線のパラメータが必要であるが、これはすでに他の手法で計算された結果あるいは NIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどを後述するファイル形式でメインプログラムに与えるものとし、生成は行わない。

［実装上で使用した手法］

- フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。
- 仕様に記述された通りに C 言語でプログラムしている。高速化や省メモリ化に関する特別なことはしていない。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
PSEC-3	楕円曲線離散対数問題	PSEC-3 の各パラメータは、160 ビット。RSA1024bit 相当	評価用の PSEC-3 のデータサイズは、128 ビット。楕円曲線のパラメータが必要。他の手法で計算された結果あるいは NIST（米国国立標準技術局）で公開されている楕円曲線のパラメータなどメインプログラムに与える。 フリーの多倍長演算ライブラリ GNU MP 3.1.1 を使用。

［システムパラメータ生成］

- システム生成は、事前に済ませており、今回の評価では実施しなかった。

［鍵対生成］

- 乱数生成方式:まず GNU MP 3.1.1 の関数 gmp_randinit(gmp_randstate_t state, gmp_randalg_t alg, …)でランダム状態変数を初期化する。乱数生成アルゴリズムは GNU MP のデフォルトのアルゴリズム(GMP_RAND_ALG_DEFAULT)を指定した。次に、関数 gmp_randseed(gmp_randstate_t state, mpz_t seed)で外部ファイルから読み込んだ種をランダム状態変数にセットする。次に、関数 mpz_urandomb(mpz_t rop, gmp_randstate_t state, unsigned long int n)で乱数を生成する。

〔鍵対生成の速度〕

測定対象	平均実行時間	備考
PSEC-3	11.5ms	なし

［暗号化］

・ パディング:不要

・ ハッシュ関数: SHA-1 を使用。

[暗号化の速度]

測定対象	平均実行時間	備考
PSEC-3	22.9ms	なし

[復号の速度]

測定対象	平均実行時間	備考
PSEC-3	12.0ms	なし

[コードサイズ（参考値)]

測定対象	コードサイズ	備考
PSEC-3	196,986byte	C言語（Intel C/ C++）ソースコードサイズ

4.4.18 DH

1. 暗号技術

1.1 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。

論文誌:1976 年、“New directions in cryptography”, IEEE Trans. Information Theory, vol. IT-22, pp.644-654, 1976

知的財産権

(提案者特許とその扱い)

Diffie-Hellman 特許（U.S.Patent Number4,200,770）

1.2 技術仕様

1)システム共通のパラメータを設定

p：素数を生成する

$g \in Z_p^*$ ：原始元を求める

gの位数をlとする

(p,l,g)をシステム共通のパラメータとする

2)ユーザーA の初期設定

$x_A (0 < x_A < l)$ ： ランダムに選択し秘密鍵とする

$y_A = g^{x_A} \pmod p$ を計算しy_Aを公開鍵とする。

3)ユーザーB の初期設定

同様にして、

$x_B (0 < x_B < l)$ ： ランダムに選択し秘密鍵とする

$y_B = g^{x_B} \pmod p$ を計算しy_Bを公開鍵とする。

4)鍵共有の処理

A の処理: $K = y_B{}^{x_A} \pmod p = g^{x_B x_A} \pmod p$

B の処理: $K = y_A{}^{x_B} \pmod p = g^{x_A x_B} \pmod p$

により、K を共有する。

2. 評価結果

2.1 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/IS1170-3, Oakley, PGP)。今年度の評価対象は、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 decision Diffie-Hellman 問題に帰着される。本来は、ランダムなビット列と見分けがつかなくなるような鍵導出関数(key derivation function)を用いるべきである。

c)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は、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倍することによち、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)s ： 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。

2.2　ソフトウェア実装評価

DH に関しては、その他評価が必要と判断した暗号技術であり、多くの実装実績を有しており、今回のソフトウェア実装評価の必要がないと判断した。

4.4.19 ECDHS in SEC1

1. 暗号技術

1.1 技術概説

1999 年に、SECG(Standards for Efficient Cryptography 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。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・ 4,745,568: Computational method and apparatus for finite field multiplication, issued May 17, 1988. This patent includes methods for efficient implementation of finite field arithmetic using a normal basis representation.

・ 5,761,305: Key Agreement and Transport Protocol with Implicit Signatures, issued June 2, 1998. This patent includes versions of the MQV protocols.

・ 5,787,028: Multiple Bit Multiplier, issued July 28, 1998.

・ 5,889,865: Key Agreement and Transport Protocol with Implicit Signatures, issued March 30, 1999. This patent includes versions of the MQV protocols.

・ 5,896,455: Key Agreement and Transport Protocol with Implicit Signatures, issued April 20, 1999. This patent includes versions of the MQV protocols.

上記提案者所有の特許および著作権は、提案者により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される。

なお、詳細は、特許所有者、及び SECG のウェブサイト（www.secg.org）を参照のこと。

http://www.secg.org/patent_policy.htm

http://www.secg.org/collateral/certicom_secg_patent.pdf

特許所有者が、特許技術の使用許諾を求める申請者に対し、妥当かつ差別待遇のない取引条件で、使用を許諾することに同意する文書を、SECG に提出済みかどうかを確認すること。

応募暗号技術仕様の公開 Web アドレス http://www.labs.fujitsu.com/theme/crypto/public_key.html

http://www.secg.org/drafts.htm

1.2 技術仕様

鍵共有スキームの概要を示す。

楕円曲線パラメータ (p, a, b, G, l)

素体 F_p 上の楕円曲線 $y^2 = x^3+ax+b$ に素数 l を位数としてもつ有理点 G があることを示す。

楕円曲線パラメータ (k, f, a, b, G, l)

既約式 f で定義される標数 2 の k 次拡大体 F_{2^k} 上の楕円曲線 $y^2+xy = x^3+ax^2+b$ に素数 l を位数としてもつ有理点 G があることを示す。

【初期設定】

1. 利用者 U および V は楕円曲線パラメータ(p, a, b, G, l)または(k, f, a, b, G, l)を生成する。
2. 利用者 U および V は上記楕円曲線パラメータに属する秘密鍵 d , 公開鍵 Q の組を生成する:

利用者 U: $d_U \leftarrow$ 1 以上 l-1 以下の整数、$Q_U \leftarrow d_U \cdot G$

利用者 V: $d_V \leftarrow$ 1 以上 l-1 以下の整数、$Q_V \leftarrow d_V \cdot G$

【鍵共有】

利用者 U と V は以下のようにして、秘密情報 K を共有する:

利用者 U: $z \leftarrow x(d_U \cdot Q_V)$, K ← Hash(z || SharedInfo)

利用者 V: $z \leftarrow x(d_V \cdot Q_U)$, K ← Hash(z || SharedInfo)

ただし、上で、楕円曲線上の点 Q に対して、x(Q)は Q のｘ座標を表す。また、SharedInfo はオプションである。

1. 3 その他

ECDHS in SEC1 は IEEE P1363 でも記載されている。

2 評価結果

2.1 安全性評価

2.1.1 暗号プリミティブの安全性

ECDHS in SEC1 はその基本的な安全性を楕円曲線上の離散対数問題によっている。

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、ECDHS in SEC1 においては、実用上有効な各種のビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC2 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からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC2 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。

2.1.2 スキームの安全性

ECDHS in SEC1 は楕円曲線を利用した Diffie-Hellman 鍵共有スキームであり、鍵共有スキーム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、また forward-secrecy も満足しない。特に、秘密鍵 d－公開鍵 Q の組として固定鍵を用いる場合に注意が必要である。

鍵共有スキームが実際に運用される場合には、能動的な攻撃者を想定する必要があるので、電子署名との組み合わせなどを検討すべきである。ちなみに、ECDHS in SEC1 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 を実現する鍵共有スキームが複数の研究者によって提案されている。

2.2　ソフトウェア実装評価

［実装仕様］

（1）使用したパラメータの位置づけ

・　以下の 2 種類の曲線を使用。

secp160r1 ： 160 ビット素体楕円曲線パラメータ

sect163r2 ： 163 ビット標数 2 の体楕円曲線パラメータ

・　どちらも SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。

（2）パラメータの設定方式

・　楕円曲線パラメータをファイルから読み込んで使用。

・　パラメータには依存しない実装を行っている。

（3）使用パラメータの検証状況

・　楕円曲線の 2 つの係数が、(コントロールすることのできない)SHA-1 の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。

・　X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。

・　各種特殊攻撃を適用できないことも確かめられている。

（4）パラメータサイズのバリエーション

・　固定パラメータだけではなく、任意の楕円曲線パラメータの利用可能。

・　素体楕円曲線パラメータで扱えるビット長は以下の通り。

112,128,160,192,224,256,384,521

・　標数 2 の体楕円曲線パラメータで扱えるビット長は以下の通り。

113,131,163,193,233,239,283,409,571

［実装上で使用した手法］

・　ライブラリは、a=0, a=-3 などの特殊パラメータに限定した高速手法は使用していない。

・　任意のパラメータに適用できる高速手法を使用。

・　パラメータには依存しない実装を行っている。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
ECDHS in SEC1 160bit 素体	ECDLP	RSA1024bit 相当以上	SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。 任意に選択されたものであることが検証可能(randomly verifiable)。 X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。 各種特殊攻撃を適用できないことも確かめられている。 任意のパラメータに適用できる高速手法を使用。 パラメータには依存しない実装を行っている。
ECDHS in SEC1 163bit 標数 2 の体		RSA1024bit 相当以上	

［システムパラメータ生成］

・ 楕円曲線パラメータ生成処理は、確率的アルゴリズムであるため、正確な予想処理時間を出すことは困難。

・ 事前の実験では、160 ビット楕円曲線の場合、PentiumIII 700MHz にて、平均 10 分以内。

測定結果

測定対象	平均実行時間	備考
ECDHS 160bit 素体	6.0 min	18 分間に 3 本の楕円曲線パラメータを生成

［初期化/終了処理］

測定結果

測定対象		平均実行時間	備考
ECDHS 160bit 素体	PC/PF	0.004 ms	なし
	PI	13.525 ms	なし
	PV	60.438 ms	なし
	WC/WF	0.032 ms	なし
ECDHS 163bit 標数 2 の体	PC/PF	0.004 ms	なし
	PI	15.283 ms	なし
	PV	43.917 ms	なし
	WC/WF	0.032 ms	なし

PC/PF :楕円曲線パラメータ領域の獲得/解放処理。

PI :ファイルから読んだパラメータを領域に設定。

PV :パラメータ検証処理。パラメータが与えられた最初の時にのみ必要。

生成元が曲線上の点か、点を位数倍すると無限遠点となるかなど、正当性をチェック。

WC/WF :作業領域獲得/解放処理。

［鍵対生成］

（1）素数判定条件

・ 方式上、不要。

（2）乱数生成方式

・ 富士通オリジナル擬似乱数生成アルゴリズム。DES/SHA-1 などを使用。

（3）測定結果

・　鍵対の検証処理は、自分で鍵を生成したときには不必要。

〔鍵対生成〕

測定対象	平均実行時間	備考
ECDHS in SEC1 160bit 素体	1.9ms	鍵対生成
	5.8ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
ECDHS in SEC1 163bit 標数 2 の体	3.2ms	鍵対生成
	8.0ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。

［鍵共有処理］

（1）パディング

・　使用せず。

（2）ハッシュ関数

・　使用せず。

（3）測定結果

［鍵共有処理（片側)］

測定対象	平均実行時間	備考
ECDHS in SEC1 160bit 素体	6.6ms	なし
ECDHS in SEC1 163bit 標数 2 の体	8.8ms	なし

（1）　その他

KDF : ANSI-X9.63-KDF with SHA-1

鍵共有プリミティブ : standard elliptic curve Diffie-Hellman primitive

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
ECDHS in SEC1	356,352byte	性能測定用実行形式テストプログラムのサイズ

4.4.20 ECMQVS in SEC1

1. 暗号技術

1. 1 技術概説

1999 年に、SECG(Standards for Efficient Cryptography 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。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・ 4,745,568: Computational method and apparatus for finite field multiplication, issued May 17, 1988. This patent includes methods for efficient implementation of finite field arithmetic using a normal basis representation.

・ 5,761,305: Key Agreement and Transport Protocol with Implicit Signatures, issued June 2, 1998. This patent includes versions of the MQV protocols.

・ 5,787,028: Multiple Bit Multiplier, issued July 28, 1998.

・ 5,889,865: Key Agreement and Transport Protocol with Implicit Signatures, issued March 30, 1999. This patent includes versions of the MQV protocols.

・ 5,896,455: Key Agreement and Transport Protocol with Implicit Signatures, issued April 20, 1999. This patent includes versions of the MQV protocols.

上記提案者所有の特許および著作権は、提案者により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される。

なお、詳細は、特許所有者、及び SECG のウェブサイト（www.secg.org）を参照のこと。

http://www.secg.org/patent_policy.htm

http://www.secg.org/collateral/certicom_secg_patent.pdf

特許所有者が、特許技術の使用許諾を求める申請者に対し、妥当かつ差別待遇のない取引条件で、使用を許諾することに同意する文書を、SECG に提出済みかどうかを確認すること。

応募暗号技術仕様の公開 Web アドレス http://www.labs.fujitsu.com/theme/crypto/public_key.html

http://www.secg.org/drafts.htm

1.2 技術仕様

鍵共有スキームの概要を示す。

楕円曲線パラメータ (p, a, b, G, l)

素体 F_p 上の楕円曲線 $y^2 = x^3+ax+b$ に素数 l を位数としてもつ有理点 G があることを示す。

楕円曲線パラメータ (k, f, a, b, G, l)

既約式 f で定義される標数 2 の k 次拡大体 F_2^k 上の楕円曲線 $y^2+xy = x^3+ax^2+b$ に素数 l を位数としてもつ有理点 G があることを示す。

【初期設定】

1. 利用者 U および V は楕円曲線パラメータ(p, a, b, G, l)または(k, f, a, b, G, l)を生成する。
2. 利用者 U および V は上記楕円曲線パラメータに属する秘密鍵 d－公開鍵 Q の組をそれぞれ 2 組ずつ生成する:

利用者 U: $d_{1,U} \leftarrow$ 1 以上 n-1 以下の整数, $Q_{1,U} \leftarrow d_{1,U} \cdot G$

$d_{2,U} \leftarrow$ 1 以上 n-1 以下の整数, $Q_{2,U} \leftarrow d_{2,U} \cdot G$

利用者 V: $d_{1,V} \leftarrow$ 1 以上 n-1 以下の整数, $Q_{1,V} \leftarrow d_{1,V} \cdot G$

$d_{2,V} \leftarrow$ 1 以上 n-1 以下の整数, $Q_{2,V} \leftarrow d_{2,V} \cdot G$

【鍵共有】

利用者 U と V は以下のようにして、秘密情報 K を共有する:

利用者 U:

1. $s \leftarrow d_{2,U} + h(Q_{2,U}) \cdot d_{1,U} \pmod n$
2. $P \leftarrow s \times (Q_{2,V} + h(Q_{2,V}) \cdot Q_{1,V})$、 $z \leftarrow x(P)$
3. $K \leftarrow$ Hash(z || SharedInfo)

利用者 V:

1. $s' \leftarrow d_{2,V} + h(Q_{2,V}) \cdot d_{1,V} \pmod n$
2. $P \leftarrow s' \times (Q_{2,U} + h(Q_{2,U}) \cdot Q_{1,U})$, $z \leftarrow x(P)$
3. $K \leftarrow$ Hash(z || SharedInfo)

ただし、上で、楕円曲線上の点 Q に対して、x(Q)は Q のｘ座標を表し、h(Q)は以下のようにして計算される整数である:

1. $x' \leftarrow x(Q) \pmod{2^{\mathrm{Ceiling}((\log n)/2)}}$
2. $h(Q) \leftarrow x' + 2^{\mathrm{Ceiling}((\log n)/2)}$

ここで、Ceiling(m)はm以上の最小の整数を表す。

また、SharedInfo はオプションである。

1.3 その他

ECMQVS in SEC1 は IEEE P1363 でも記載されている。

2. 評価結果

2.1 安全性評価

2.1.1　暗号プリミティブの安全性

ECMQVS in SEC1 はその基本的な安全性を楕円曲線上の離散対数問題によっている。

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、ECMQVS in SEC1 においては、実用上有効な各種のビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC2 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からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC2 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。

2.1.2　スキームの安全性

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、能動的な攻撃者に対する安全性には問題がある。ECMQVS in SEC1 では、2 組の秘密鍵－公開鍵ペアを用いるが、その一方は固定鍵もう一方は一時鍵を使用することが想定されているようである。[注] そのような想定をすれば, ECDHS in SEC1 と比較して安全性が増していると考えられるものの、現時点では能動的攻撃に対する安全性は証明されていない。実際、ECMQVS in SEC1 の安全性証明には LDH 仮定と呼ばれる仮定が必要であり（十分かどうかはわからない)、LDH 仮定の正当性は現時点では不明との指摘がある。

鍵共有スキームが実際に運用される場合には、能動的な攻撃者を想定する必要があるので、電子署名との組み合わせなどを検討すべきである。因みに、ECMQVS in SEC1 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 を実現する鍵共有スキームが提案されている。

注 ただし、提案された仕様書にはそのような記述は見当たらない。誤って 2 組ともに固定鍵を使わないよう注意すべきである。

2.2　ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

- 以下の２種類の曲線を使用。
  - secp160r1 ： 160 ビット素体楕円曲線パラメータ
  - sect163r2 ： 163 ビット標数 2 の体楕円曲線パラメータ
- どちらも SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。

（２）パラメータの設定方式

- 楕円曲線パラメータをファイルから読み込んで使用。
- パラメータには依存しない実装を行っている。

（3）使用パラメータの検証状況

- 楕円曲線の２つの係数が、(コントロールすることのできない)SHA-1 の出力値によって関係づけられているため、任意に選択されたものであることが検証可能(randomly verifiable)。
- X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。

・ 各種特殊攻撃を適用できないことも確かめられている。

（4）パラメータサイズのバリエーション

・ 固定パラメータだけではなく、任意の楕円曲線パラメータの利用可能。

・ 素体楕円曲線パラメータで扱えるビット長は以下の通り。

112,128,160,192,224,256,384,521

・ 標数 2 の体楕円曲線パラメータで扱えるビット長は以下の通り。

113,131,163,193,233,239,283,409,571

［実装上で使用した手法］

・ ライブラリは、a=0，a=-3 などの特殊パラメータに限定した高速手法は使用していない。

・ 任意のパラメータに適用できる高速手法を使用。

・ パラメータには依存しない実装を行っている。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
ECMQVS in SEC1 160bit 素体	ECDLP	RSA1024bit 相当以上	SEC1 仕様の推奨パラメータの一つ。ANSI X9.62 仕様に準拠。 任意に選択されたものであることが検証可能(randomly verifiable)。 X9.62 仕様どおり、SHA-1 の入力値もパラメータの一部として記載。 各種特殊攻撃を適用できないことも確かめられている。 最適化すればさらなる高速化の可能性あり。 任意のパラメータに適用できる高速手法を使用。 パラメータには依存しない実装を行っている。
ECMQVS in SEC1 163bit 標数 2 の体		RSA1024bit 相当以上	

［システムパラメータ生成］

・ 楕円曲線パラメータ生成処理は、確率的アルゴリズムであるため、正確な予想処理時間を出すことは困難。

・ 事前の実験では、160 ビット楕円曲線の場合、PentiumIII 700MHz にて、平均 10 分以内。

測定結果

測定対象	平均実行時間	備考
ECMQVS 160bit 素体	6.0 min	18 分間に 3 本の楕円曲線パラメータを生成

［初期化/終了処理］

測定結果

測定対象		平均実行時間	備考
ECMQVS 160bit 素体	PC/PF	0.004 ms	なし
	PI	13.525 ms	なし
	PV	60.438 ms	なし
	WC/WF	0.032 ms	なし
ECMQVS 163bit 標数 2 の体	PC/PF	0.004 ms	なし
	PI	15.283 ms	なし
	PV	43.917 ms	なし
	WC/WF	0.032 ms	なし

PC/PF :楕円曲線パラメータ領域の獲得/解放処理。

PI :ファイルから読んだパラメータを領域に設定。

PV :パラメータ検証処理。パラメータが与えられた最初の時にのみ必要。

生成元が曲線上の点か、点を位数倍すると無限遠点となるかなど、正当性をチェック。

WC/WF :作業領域獲得/解放処理。

［鍵対生成］

（1）素数判定条件

・ 方式上、不要。

（2）乱数生成方式

・ 富士通オリジナル擬似乱数生成アルゴリズム。DES/SHA-1 などを使用。

（3）測定結果

・ 鍵対の検証処理は、自分で鍵を生成したときには不必要。

〔鍵対生成〕

測定対象	平均実行時間	備考
ECMQVS in SEC1 160bit 素体	1.9ms	鍵対生成
	5.8ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。
ECMQVS in SEC1 163bit 標数 2 の体	3.2ms	鍵対生成
	8.0ms	鍵対の検証。鍵対の検証処理は、自分で鍵を生成したときには不必要。

［鍵共有処理］

（1）パディング

・ 使用せず。

（2）ハッシュ関数

・ 使用せず。

（3）測定結果

［鍵共有処理（片側)］

測定対象	平均実行時間	備考
ECMQVS in SEC1 160bit 素体	13.2ms	なし

ECMQVS in SEC1 163bit 標数 2 の体	16.9ms	なし

（1） その他

- KDF : ANSI-X9.63-KDF with SHA-1

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
ECMQVS in SEC1	356,352byte	性能測定用実行形式テストプログラムのサイズ

4.4.21 HDEF-ECDH

1. 暗号技術

1.1 技術概要

HDEF-ECDH は、1999 年に、電子情報通信学会情報セキュリティ研究会において、宮地らにより提案され た暗号技術であり、ECDLP の安全性根拠に基づく鍵共有を実現する統合方式の暗号技術である。

参考文献

A. Miyaji and H. Shizuya, ``Integration of DLP-based cryptosystems'', IEICE Japan Tech. Rep., bf ISEC99-48(1999-09), 73-80.

(応募者特許とその扱い) (応募者資料による)

楕円曲線に関する特許に関しては、以下の特許を申請中である。

（1）特願 2000-126692 発明の名称:統合装置 詳細（出願日 H12.3.23）

（2）特願 2000-243434 発明の名称:楕円曲線生成装置 詳細（出願日 H12.7.6）

(関連特許)

利用した DH 鍵共有法に対する特許は期限が切れており、他社特許抵触の問題はないと考えられる。

· 関連する提案者が保持する特許

応募暗号技術仕様の公開 Web アドレス

http://grampus.jaist.ac.jp:8080/miyaji-lab/IPA/index.html

1.2 技術仕様

(1)楕円曲線パラメータ生成仕様

1. $d \equiv 19 (\mathrm{mod}\, 24)$を満たす整数$d$を選ぶ

2. 整数lに対して$p = dl^2 + dl + \frac{d+9}{4}$とおく

3. $(p, p-2)$のどちらかが合成数のとき1.に戻る

4. dによって定まる類多項式$P_d(x)$を求める

5. $P_d(x) \equiv 0 (\mathrm{mod}\, p)$の解 j_0 を求める

6. j_0 を j-invariantとするF_p上楕円曲線$\{E_{j_0}\}$を構成する。ここでE_{j_0}は

$$E_{j_0} : y^2 = x^3 + a_{j_0} x + b_{j_0}$$

$$a_{j_0} = \frac{3j_0}{1728 - j_0} (\mathrm{mod}\, p), b_{j_0} = \frac{2j_0}{1728 - j_0} (\mathrm{mod}\, p)$$

となる楕円曲線で、$\{E_{j_0}\}$はその同型な楕円曲線の集合

7. $E : y^2 = x^3 + ax + b \in \{E_{j_0}\}$を, $\#E(F_p) = p - 2$かつ$a = -3$あるいは小さいaをとる

・$d = 403$に固定した場合のアルゴリズムの記述あり。
・p のビット長は 160 以上を推奨。特に、160、192、224 ビットを推奨。
・素数は、$p = 2^u - c$ (cは小さい整数)の形のものを推奨。

(2)鍵共有スキーム

1)ユーザーA の初期設定

E_A / F_{p_A}：素数位数楕円曲線
$G_A \in E_A(F_{p_A})$　：　生成元をランダムに選択
$d_A (0 < d_A < p_A - 2)$　：　ランダムに選択し秘密鍵とする
$U_A = d_A G_A$　　を計算する
$(E_A / F_{p_A}, U_A, G_A)$　：　公開鍵とする。

2)ユーザーB の初期設定

同様にして、

$(E_B / F_{p_B}, U_B, G_B)$　：　公開鍵とする。

3)同じ曲線を使用する場合

A の処理: $K = d_A U_B = d_A d_B G$
B の処理: $K = d_B U_A = d_B d_A G$
により、K を共有する。

4)異なる曲線を使用する場合

A の処理:

$r_A (0 < r_A < p_B - 2)$　：　ランダムに選択し秘密鍵とする
$R_A = r_A G_B$　　を計算する
R_A　をＢに送る

B の処理:

$r_B (0 < r_B < p_A - 2)$　：　ランダムに選択し秘密鍵とする
$R_B = r_B G_A$　　を計算する
R_B　をＡに送る

A の処理: $K_A = d_A R_B = d_A r_B G_A, K_B = r_A U_B = r_A d_B G_B$
B の処理: $K_B = d_B R_A = d_B r_A G_B, K_A = r_B U_A = r_B d_A G_A$
により、K_A, K_B を共有する。

ただし、実際の共有鍵は、K_A, K_B から計算できる値としてよい。例えば、それらの x 座標の排他的論理和など。

2. 評価結果

2.1 安全性評価

(1) 提案された曲線パラメータの安全性

a)既知の効率的な攻撃法は適用できない。

b)トレースが 3 であり、かつ discriminant が小さな CM 体を持つという、非常に限られたクラスの楕円曲線である。現在は効率的に適用できる攻撃法は存在しないが、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。

c)提案者が非常に限られたクラスの楕円曲線を生成する理由は、「容易な楕円曲線の生成が可能になれば、ユーザー毎に異なる楕円曲線を生成できる」からと主張している。生成のための時間は、「406.582×素数判定時間」と見積もられているが、より具体的な実験数値は示されておらず、ソフトウェア実装評価も行われていない。生成がどの程度容易かを、実際に検証したデータはない。

d)提案者は「安全性の観点からは、ユーザー毎に異なる楕円曲線を使うことが望ましい」と主張しているが、その根拠は明確に述べられてはいない。この根拠が明確でないと、上記 b)の危険を冒して非常に限られたクラスの楕円曲線を使用する意義も明確にならない。提案の良否については、更なる検討が必要。

(2) 鍵共有スキームの安全性

a)鍵共有スキームに関する応募者による詳細な自己評価は行われていない。

b)提案スキームは、各ユーザーが同じ楕円曲線を使う基本形と、各ユーザーごとに異なる楕円曲線を使う変形版とからなる。
ECDH プロトコルには多くのバリエーションが存在するので、個々のプロトコル毎の評価が必要である。
変形版に関して、提案者は「安全性の観点からは、ユーザー毎に異なる楕円曲線を使うことが望ましい」と主張しているが、その根拠は明確に述べられてはいない。変形版の提案の良否については、更なる検討が必要。

c)「秘密共有プロトコル」としての基本形は、受動的攻撃のみを仮定した場合、Diffie-Hellman 問題に帰着される。しかし、共有された点の x 座標を用いる場合、それは x^3+ax+b が $\bmod p$ で quadratic residue となるような x に限定される。それゆえ、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、x を上記範囲に限定した decisional Diffie-Hellman 問題に帰着される。本来は、ランダムな $|p|$ ビット列と見分けがつかなくなるような鍵導出関数(key derivation function)を用いるべきである。

d)「共有秘密をセッション鍵として使用するスキーム」においては、安全性を左右する様々な要因がある。提案スキームにおいては、これらの要因は明示されていない。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)。

e)「共有秘密をセッション鍵として使用するスキーム」においては、提案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。

f)【同じ曲線パラメータの場合】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倍することによち、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)s : 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。

g)【違う曲線パラメータの場合】

1)basic protocol の場合

-受動的攻撃に対しては、(key derivation function 修正により)ECDDH に帰着。

- 能動的攻撃に対する安全性に関しては何もいえない。

-forward secrecy は達成されていない。

2) one time key(ephemeral)の場合

-同じパラメータのときと同様にして、改良可能。

-mutual authentication を達成するためには、key confirmation flow を付け加える必要がある。

h)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。

2.2 ソフトウェア実装評価

［実装仕様］

（１）使用したパラメータの位置づけ

・N=1,024 ビットの RSA 暗号（署名同等以上の強度）

（２）パラメータの設定方式

- 本オブジェクトコードは、鍵共有ユーザーが同一の楕円パラメータを用いることを前提にした実装としている（すなわち、ユーザーＡとＢは、初期設定において、同一の素数位数楕円曲線とベースポイントを選択するケース)。
- システムパラメータの生成や、システムパラメータの読み込みを行うコードを保持しない。

（３）使用パラメータの検証状況

- 使用した楕円パラメータは、MOV(FR)帰着攻撃、SSSA 攻撃に対して安全であることを検証している。詳しくは HDEF-ECDH の自己評価書を参照いただきたい。

（４）パラメータ・サイズのバリエーション

- 実測したパラメータのサイズは、160 ビットの１種類

（５）パラメータの設定方式

- 本仕様書の第 3 章で規定する楕円パラメータを保持しており、テーブル初期化等のセットアップ処理を要しない。

（６）使用パラメータの検証状況

- 暗号技術仕様書で記述

（７）パラメータ・サイズのバリエーション

- 暗号技術仕様書に記述。

［実装上で使用した手法］

- 本オブジェクトコードではベースポイントが固定であるが、任意点のべき倍演算と同様のべき倍演算アルゴリズム(符号付き２進法とウィンドウ法の組合わせ)を用いている。つまり、

実際の使用においては、固定点アルゴリズムを利用できるのでさらに高速化が望める。

- 定義体上での乗算剰余演算（160×160→160 ビット）についてはアセンブラで記述。

〔特徴及び測定パラメータ〕

測定対象	安全性の根拠	実装パラメータと位置づけ	その他
HDEF-ECDH	楕円曲線離散対数問題	RSA1024bit 相当以上パラメータサイズは、160 ビット	使用した楕円パラメータは、MOV(FR)帰着攻撃、SSSA 攻撃に対して安全であることを検証している。 評価用コードではベースポイントが固定。 任意点のべき倍演算と同様のべき倍演算アルゴリズム（符号付き２進法とウィンドウ法の組合わせ）を使用。 固定点アルゴリズムを利用することで高速化が望める。 定義体上での乗算剰余演算（160×160→160 ビット）についてはアセンブラで記述。 基本形の測定であること 任意点べき倍算アルゴリズム使用 乗算剰余演算アセンブラ

［システムパラメータ生成］

・事前に実施されており、今回は測定しなかった。

［鍵対生成］

（１）判定条件

・定義体ビットサイズより１ビット小さいビットサイズの乱数を生成し、秘密鍵とする。

・前記秘密鍵を用いてベース点をスカラ倍し、公開鍵とする。

（２）乱数生成方式

・鍵対の生成では乱数を必要とする。

・この時に用いる乱数は予測不可能な精度のよい乱数を使うことが望ましい。

・本オブジェクトコードは独自に疑似乱数生成ルーチンを有していない。そのため、本オブジェクトコードでは、乱数の生成にＣ標準関数の rand()を用いている。

・実システムへの本アルゴリズム適用においては、安全性の高い疑似乱数ルーチンを用いるべきである。

（３）測定結果

〔鍵対生成〕

測定対象	平均実行時間	備考
HDEF-ECDH	1.5ms	片側処理。鍵サイズ:160 ビット

［鍵共有側処理］

（１）パディング

- 使用せず。

（２）ハッシュ関数

・ SHA-1 を使用。

（3） 測定結果

［鍵共有処理（片側)］

測定対象	平均実行時間	備考
HDEF-ECDH	1.8ms	鍵サイズ:160 ビット

［使用したパラメータ］

p = 730 75081 86654 51459 52396 17144 99640 83306 20843 44321

楕円曲線 E(F(p)) : $y^2 = x^3 + ax + b$

a = -3

b = 145 65097 62508 47204 31616 18686 75773 20971 18776 24364

ベース点 G = (g_x, g_y)、位数 q

g_x = 0

g_y = 524 69857 95054 95094 28091 57543 77106 79925 14989 32020

q = 730 75081 86654 51459 52396 17144 99640 83306 20843 44319

［コードサイズ（参考値)］

測定対象	コードサイズ	備考
HDEF-ECDH	73,758 Bytes	定義体上での乗算剰余演算(160×160→160 ビット)についてはアセンブラで記述。 その他はC言語。

5. 共通鍵暗号の評価

共通鍵暗号においては、64 ビットブロック暗号、128 ビットブロック暗号、ストリーム暗号の 3 つのカテゴリに分類し、暗号技術の評価を行った。対象暗号は下表である。5.1 節では、カテゴリ内の暗号技術を、その特徴、安全性、実装性の観点から相互比較し、短評を述べる。5.2 節では個別暗号に関し、それらの観点のより詳細な記述を行う。なお、暗号は暗号名のアルファベット順に配置してある。

表 5.1　共通鍵暗号名

64 ビットブロック暗号	CIPHERUNICORN-E、FEAL-NX、Hierocrypt-L1、MISTY1、Triple DES
128 ビットブロック暗号	Camellia、CIPHERUNICORN-A、Hierocrypt-3、MARS、RC6、SC2000、Rijndael
ストリーム暗号	MULTI-S01、TOYOCRYPT-HS1

5.1 暗号種別による評価

5.1.1 64 ビットブロック暗号

対象は、CIPHERUNICORN-E、FEAL-NX、Hierocrypt-L1、MISTY1 及び Triple DES の 5 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追加した。、評価概要を表 5.1.1 を含む次頁以降に示す。記述内容は以下の通り。

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（線形／差分攻撃耐性、代数的及びその他攻撃耐性、アバランシュ特性）から記述する。

- 線形／差分攻撃耐性では、汎用的な確率的攻撃方法である線形解読／差分解読法に対する強度評価指標として、最大差分／線形確率又は最大差分／線形特性確率を示す。
- 代数的及びその他攻撃耐性では、高階差分や補間攻撃、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、通常この手法の攻撃で問題になる事は少ない。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。
- アバランシュ評価は、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、直接に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(SW)実装評価

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、SW 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。データランダム化部と鍵スケジュール部＋データランダム化部の 2 種類の測定を実施した。

ハードウェア(HW)実装評価

評価のためのターゲットデバイスとしては、0.25～0.35μm の ASIC ライブラリであり、設計記述言語は Verilog-HDL、回路合成には Design Compiler を使用している。但し、実行速度、ゲート規模等に関しては、実装アーキテクチャの違いや最適化の状況が異なるため、あくまでも“目安”でしかあり得ない。従って実装者の経験に左右されるところが大きいが、実行速度、ゲート規模としては、概ね向上（高速化、小型化）が期待できる。

総合評価

安全性及び実装評価を総合した観点から、評価結果を下表に示す。

表 5.1.1　共通鍵暗号（64bit ブロック暗号）の総合評価一覧表

名称	CIPHERUNICORN-E	FEAL-NX	Hierocrypt-L1	MISTY1	Triple DES (3-Key)
特徴	NEC(1998) Feistel 型、16 段。段関数は複雑。安全性を高める意図で本流部と一時鍵生成部で構成。 F 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／線形解読法に耐性。 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段関数構造の設計。	NTT(1990) Feistel 型 N 段、N は 32 以上偶数。初期、最終処理として、拡大鍵 EXOR。 段関数は、4 個の S 関数で構成。 S 関数は 8bit 単位の演算処理。 加算、EXOR、巡回シフト演算。 8bit CPU の SW 実装に適する。 統計的評価でデータ攪拌+効果の高い構造を採用。 FEAL-N の、鍵処理部を拡張。 10 年以上経つ暗号である。安全性評価に応じ、段数増加。	東芝(2000) 入れ子型 SPN 構造 6 段。各段は XS 関数の 2 並列及び 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／線形解読法耐性。 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 層の設計には、活性 S-box 数の下限を符号理論で 保証。	三菱(1996) Feistel 構造 8 段。2 段毎に FL 関数を挿入。 段関数の内部構造で変形 Feistel 構造を再帰的に使用。 S-box は 7×7 の及び 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／線形解読法に耐性。HW 実装を考え、低い代数次数。 テーブル参照、EXOR、AND、OR 差分／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。	IBM(1979) DES を 3 回繰り返した組み合わせ暗号。 DES は、1977 年 FIPS で規格化。 DES は、Feistel 型 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。 HW 志向の設計。 DES は 20 年以上経つ歴史的暗号であり、現代暗号のルーツ。 AES に FIPS が引き継がれる予定。
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。 複雑な構造のため、正確な評価が難しく、継続的な評価が必要である。 処理速度は遅いグループである。	FEAL-32X は学術的に解読可能であり、長期の使用を考えた場合、推薦できない。 8bit CPU の SW 実装に適する。	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。	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。	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。

安全性の総評

線形／差分攻撃耐性

線形／差分攻撃に対する耐性は最大線形／差分確率で与えられる。この確率で安全性を保証しているのは MISTY1 と Hierocrypt-L1 である。MISTY1 は 3 段で 2^{-56} 以下であり、線形攻撃や差分攻撃には十分安全と

考えられる。Hierocrypt-L1 も 2 段でこの確率として 2^{-48} 以下が保証され、この保証を線形／差分攻撃に対する証明可能安全性という。

最大差分／線形確率の真値を求める事は困難であり、それに準じた指標として最大線形／差分特性確率がある。最大特性確率の評価は、

- 構成部品の最大差分／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／線形特性確率が 2^{-90} を超えない事が示されている。後者はその段関数の複雑な構造のため、解析が難しく、簡略化した段関数に対し truncated vector 探索を交えて 12 段差分特性確率及び 8 段線形特性確率の上界が 2^{-64} を下回ることが示されている。64 ビットブロック暗号のこれら特性確率が 2^{-64} 以下になる事を安全性の証とする手法を、線形／差分攻撃に対する実際的安全性保証という。

計算機探索の結果として、最大特性確率が評価されているのが DES と FEAL-NX である。DES は、差分特性確率 $2^{-54.1}$、線形特性確率 $2^{-44.9}$ である。DES を 2 回繰り返した、Double DES 以上で線形／差分攻撃に対しては安全と考えられる。FEAL-NX については 31 段差分特性確率 2^{-62}、25 段線形特性確率 $2^{-62.3}$ である。これを使って攻撃するならば、FEAL-32X は、学術的には 2^{99} の計算量で解読が可能である。

以上のように、FEAL-32X を除いて、学術的な線形攻撃／差分攻撃耐性が保証されている。しかし、FEAL-32X に関しても、この解読に要する計算量や必要データを考えると、現時点において実用的には安全と考える。

代数的及びその他の攻撃耐性

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)又は適切な拡大体における展開式を使い攻撃を行う。一般的に、こ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／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の攻撃が効果的となる場合が多い。

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、全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耐性。
- S-box の全単射性に基づく高階差分攻撃耐性（SQUARE 攻撃耐性）

結果として、いずれの暗号方式も提案段数においては、これら攻撃法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／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[5]。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^{37}、計

[5]大熊、佐野、村谷、本山、川村，ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 の安全性について，SCIS2001,11A-4,(2001)

算量 2^{75} で 5 段までとなる。

補間攻撃（又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、入力変数の取り方及び着目する出力変数によりその個数は変わり、全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 8bit 単位の小ブロック 8 個（64 ビットブロック暗号）又は 16 個(128 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が評価した。いずれの暗号方式も全数探索より効率のよい解読方法は発見されていない。

DES については、鍵全数探索により約 22 時間で解読に成功した報告もあり、現実的な意味で解読可能である。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、どの暗号方式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

「鍵スケジュール部を含む暗号化処理全体」では、全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、FEAL-NX、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、FEAL-NX、Hierocrypt-L1、MISTY1で期待値を満たさない部分を検出した。

名称	アバランシュ性評価
CIPHERUNICORN-E	ラウンド関数では特徴は見られない。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。
FEAL-NX	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では5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。
Hierocrypt-L1	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。
MISTY1	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。

ソフトウェア(SW)実装評価

データランダム化部

測定値は clock 数だが、分かりやすいように[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、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。測定プログラムは全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毎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

64 ビットブロック暗号	速度[Mbps]
CIPHERUNICORN-E	29.0(28.9) / 29.3(29.2)
FEAL-NX	117.8(113.9) / 117.2(111.8)
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

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、残りは充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、実装に於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

64 ビットブロック暗号	速度[Mbps]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

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。これらは暗号化/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

64 ビットブロック暗号	速度[Mbps]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

Alpha21264 は 64 ビット CPU で巨大な一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

測定値は clock 数だが、分かりやすいように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。従って、この表の値のみで断定するのは危険である。測定プログラムは、全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毎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。従って、数μsec で処理が終了することが望ましい。

1．PC 環境

64 ビットブロック暗号	速度[μsec]
CIPHERUNICORN-E	3.72(3.73) / 3.70(3.72)
FEAL-NX	1.23(1.27) / 1.24(1.27)
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

2．サーバ環境

64 ビットブロック暗号	速度[μsec]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

3．ハイエンド環境

64 ビットブロック暗号	速度[μsec]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

以上の結果から、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。

SW 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(HW)実装評価

HW 実装評価の対象となった 64bit ブロック暗号は、FEAL-NX、Hierocrypt-L1、MISTY1 の 3 暗号方式である。このうち、CIPHERUNICORN-E は、応募書類の中に、「HW による実装も可能」という記載があるだけで、具体的な HW 実装例（規模等）の記載が無いため、HW 実装評価の対象とはしなかった。

今回行ったブロック暗号の HW 実装評価に関しては 2 通りのアーキテクチャが考えられる。つまり、ループアーキテクチャを採用する場合と採用しない場合に大別される。ループアーキテクチャで評価したアルゴリズムは、MISTY1 と Triple DES であり、ループアーキテクチャで評価しなかったアルゴリズムは FEAL-NX と Hierocrypt-L1 である。この 2 つのグループに分けて比較を行った。

これら方式の HW 評価対象のパラメータは、以下の通りである。

評価対象	繰り返し段数	鍵長(bit)
FEAL-NX	32 段	128
Hierocrypt-L1	6 段	128
MISTY1	8 段	128
Triple DES(参考)	48(=16×3)段	168

評価結果

回路規模、クリティカルパス遅延、処理速度の評価結果は下表の通りである。

評価対象		回路規模（単位：Gate）			
		データランダム化部	鍵スケジュール部	制御回路部	Primitive 全体
FEAL-NX	*1	34,830	34,840	—	69,970
Hierocrypt-L1	*1	278,130	95,397	—	373,526
MISTY1	*2	19,935	44,773	94	64,809
		10,609	28,194	68	38,875
Triple DES (参考)	*1	124,888	23,207	—	148,147
	*2	4,218	1,333	151	6,496
		2,011	1,088	134	5,111

＊ 1：最適化することを行わず、アルゴリズムの全体を速度重視の設計を想定。パイプラインアーキテクチャは採用しない。

＊ 2：ループアーキテクチャを採用し、パイプラインアーキテクチャは採用しない。また、置換表はテーブルを用い、論理合成ツールを用いて最適化を行う。上段は速度優先、下段は回路規模優先。

この評価結果から、Triple DES との相対的な比較を試みると、ループ・アーキテクチャを採用しない場

合（暗号アルゴリズム全体実装を行う）グループ内での回路規模の比較では、FEAL-NX は Triple DES の約 1/2 倍であり、Hierocrypt-L1 は Triple DES の約 2.5 倍となっている。

一方、ループ・アーキテクチャを採用したグループ内での回路規模の比較では、MISTY1 が Triple DES の約 10～7.6 倍となっている。

次に、処理速度を規定するクリティカルパス遅延とクリティカルパス遅延から想定される処理速度は以下の通りと評価された。

評価対象		クリティカルパス(ns)	処理速度(Mbps)
FEAL-NX	*1	227.2	281.69
Hierocrypt-L1	*1	70.13	912.59
MISTY1	*2	11.86	600
		24.70	288
Triple DES (参考)	*1	157.09	407.4
	*2	4.44	244
		7.10	153

＊ 1：最適化することを行わず、アルゴリズムの全体を速度重視の設計を想定。パイプラインアーキテクチャは採用しない。

＊ 2：ループアーキテクチャを採用し、パイプラインアーキテクチャは採用しない。また、置換表はテーブルを用い、論理合成ツールを用いて最適化を行う。上段は速度優先、下段は回路規模優先。

まず、ループ・アーキテクチャを採用しない（暗号アルゴリズム全体実装を行う）グループでの処理速度の比較では、FEAL-NX は Triple DES の約 0.7 倍であり、Hierocrypt-L1 は Triple DES の処理速度の約 2.25 倍高速となっている。一方、ループ・アーキテクチャを採用したグループの比較では、MISTY1 は Triple DES の処理速度の約 2.5～1.9 倍となっている。

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事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、下表である。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。

表 5.1.2　各暗号の安全性余裕と速度（Pentium III）

	安全性余裕＝ 段数／攻撃可能段数	速度（データランダム化部）	速度(鍵スケジュール部込み)
CIPHERUNICORN-E	16 / -*	0.60	0.82
FEAL-NX	32 / 32	2.41	2.45
Hierocrypt-L1	6 / 3.5	4.25	3.97
MISTY1	8 / 5	4.07	5.57
Triple DES	48 / 48	1	1

＊　CIPHERUNICORN-E は、学術的な解読段数がまだ知られていない。

5.1.2 128 ビットブロック暗号

対象は、Camellia、CIPHERUNICORN-A、Hierocrypt-3、MARS、SC2000、RC6、Rijndael の 7 種類である。Camellia から RC6 までの応募があり、Rijndael は、その他評価が必要な暗号として CRYPTREC で追加した。概要を表 5.1.3 に示す。

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（線形／差分攻撃耐性、代数的及びその他攻撃耐性、アバランシュ特性）から記述する。

- 線形／差分攻撃耐性では、汎用的な確率的攻撃方法である線形解読／差分解読法に対する強度評価指標として、最大差分／線形確率又は最大差分／線形特性確率を示す。
- 代数的及びその他攻撃耐性では、高階差分や補間攻撃、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、通常この手法の攻撃で問題になる事は少ないが、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。
- アバランシュ評価は、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、通常直接に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こともある。

ソフトウェア(SW)実装評価

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、SW 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。データランダム化部と鍵スケジュール部＋データランダム化部の 2 種類の測定を実施した。

ハードウェア(HW)実装評価

基本的には、アルゴリズムを最適化することを行わず、またアルゴリズムの全体を速度重視の設計を想定した評価となっている。評価のためのターゲットデバイスとしては、0.35μm の ASIC ライブラリであり、設計記述言語は Verilog-HDL、回路合成には Design Compiler を使用している。

総合評価

安全性及び実装評価を総合した観点から、評価結果を下表に示す。

表 5.1.3　共通鍵暗号（128bit 暗号）の一覧表

名称	Camellia	CIPHER UNICORN-A	Hierocrypt-3	MARS	SC2000	RC6	Rijndael
特徴	NTT、三菱(2000) Feistel 型、18 段(128 bit 鍵)、24 段(192/256 bit 鍵)、6 段毎に FL/FL^{-1}関数。 初期、最終処理として拡大鍵 EXOR。 段関数は 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)上のべき乗演算をベースに設計し、差分／線形解読法に耐性。 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。	NEC(2000) Feistel 型、16 段。 段関数 F 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。 段関数は 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／線形解読法に耐性。 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段関数構造を設計。	東芝(2000) 入れ子型 SPN 構造 6 段(128 bit 鍵)、7 段(192 bit 鍵)、8 段(256 bit 鍵)。各段は XS 関数の 4 並列及び 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／線形解読法に耐性。 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。	IBM(1998) データランダム化部は、前方混合、暗号コア、後方混合の 3 部構成。鍵長は 128 から 448 bit。 暗号コアは 32bit 4 block からなる変形 Feistel 16 段。 コア部の非線形関数 E は、32bit 入力 96bit 出力。 E 関数は 9×32 の S- box、鍵乗算、鍵加算、巡回シフト、EXOR で構成。 S-box はランダムに作り、線形確率・差分確率他に配慮した基準で選択。	富士通(2000) Feistel 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 bit 鍵)、22 段(192 / 256 bit 鍵)。 SPN 構造部で 4×4 の S-box を、Feistel 部で、5×5 及び 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／線形解読法に耐性。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。	RSA セキュリティ(1998) 32bit 4 block からなる変形 Feistel 20 段。 段関数は、簡潔な構造。 32bit 入力、32 + 5 bit 出力。2 block に EXOR 及び、データ依存巡回シフトで影響。 F 関数は、乗算、加算、巡回シフトで構成。 演算は、何れも 32 bit word に対する処理で、32 bit 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。	J.Daemen and V.Rijmen(1998) SPN 構造 10 段(128 bit 鍵)、12 段(192 bit 鍵)、14 段(256 bit 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／線形解読法耐性。 拡散層 P は、byte 単位の転置 (shift row)、byte 処理による 4 byte 内拡散(Mix column)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。	安全性について、今のところ問題は見つかっていない。複雑な段関数のため、正確な評価が難しく、継続的な評価が必要である。処理速度は遅いグループである。	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。	安全性について、今のところ問題は見つかっていない。製品化の予定無との事で、ソフトウェア処理速度評価せず。	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	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。	AES 暗号であり信頼がおけると考えられる。電子政府としては、FIPS 版の再評価後の使用を推薦する。

安全性の総評

線形/差分攻撃耐性

線形／差分攻撃に対する耐性は、最大線形／差分確率で与えられる。この確率が 128 ビットブロック暗号としての安全性を十分保証する程小さいという形の安全性保証は、今回の応募には無い。

最大差分／線形確率の真値を求める事は困難であり、それに準じた指標として、最大線形／差分特性確率がある。最大特性確率の評価は、

- 構成部品の最大差分／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 と Rijndael である。Camellia は、FL 関数を除いて、12 段で差分／線形特性確率が 2^{-132} を超えず、Hierocrypt-3 は 2 段で、Rijndael は 4 段で差分／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、その段関数 F の複雑な構造のため、解析が難しく、簡略化した段関数 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が、段関数 F に対するより深い評価が望まれる。

MARS も複雑な構造をしており、正確な評価は難しいが、AES への応募以来、多くの研究者の評価を受け、16 段の暗号化コア部に関し、最大差分特性確率 2^{-156}、最大線形特性確率 2^{-120} と評価されている。なお、MARS は、提案組織より商品化計画無しとの通知があり、CRYPTREC の評価はこれで終了する。

RC6 は、構造は簡明であるが、32 bit word 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 に対する評価研究及び 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} 以下になる事を安全性の証とする手法を線形／差分攻撃に対する実際的安全性保証という。何れの暗号も、現在その値を下回っており、学術な線形攻撃／差分攻撃耐性が保証されている。

代数的及びその他の攻撃耐性

高階差分攻撃や補間攻撃耐性に関し 64 ビットブロック暗号と同様(5.1.1 節参照)に評価した。いずれの暗号方式も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／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 Rijndael 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 128bit 鍵に対しては 6 段中 3 段まで、192bit（又は 256bit）鍵に対しては 8 段（又

は 10 段）中 3.5 段まで攻撃可能である[6]。Rijndael 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 bit 鍵に対しては 10 段中 7 段まで、192 bit 鍵に対しては 12 段中 8 段まで、256 bit 鍵に対しては 14 段中 8 段までが全数探索より効率よく解読可能である。Rijndael 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 bit 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 段までが全数探索より効率よく解読できる[7]。

その他の攻撃の中で RC6 に関しカイ 2 乗攻撃が効果を挙げている。それにより、それぞれ 20 段中、128 bit 鍵に対しては 12 段まで、192 bit 鍵に対しては 14 段まで、256 bit 鍵に対しては 15 段まで全数探索より効率よく解読可能である[8]。

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、どの暗号方式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

「鍵スケジュールを含む暗号化処理全体」では、全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、MARS、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、MARS、RC6、SC2000で期待値を満たさない部分を検出した。

名称	アバランシュ性評価
Camellia	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。
CIPHERUNICORN-A	ラウンド関数では特徴は見られない。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。
Hierocrypt-3	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。
MARS	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では特徴は見られない。鍵スケジュール部では乗算処理で使用する拡大鍵に特徴が見られる。
RC6	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。
SC2000	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 bitおよび256 bitの時に特徴が見られる。

[6]大熊、佐野、村谷、本山、川村， ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 の安全性について，SCIS2001, 11A-4,(2001)

[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. R. Knudsen and W. Meier, Correlations in RC6 with a reduced number of rounds, Fast Software Encryption Workshop, 2000

ソフトウェア(SW)実装評価

データランダム化部

測定値は clock 数だが、分かりやすいように[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、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。測定プログラムは全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毎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

128 ビットブロック暗号	速度[Mbps]
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	48.7(48.6) / 48.7(48.6)

暗号化:最速値(平均値) / 復号:最速値(平均値)

最終段に比較のため Triple DES の測定値を記載した。Triple DES は 64bit 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 以外は充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、実装に於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

128 ビットブロック暗号	速度[Mbps]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

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 環境に於いて最速の 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充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

128 ビットブロック暗号	速度[Mbps]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

提案から開発期間までの期間が短い暗号こともあり、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。Alpha21264 は 64 ビット CPU で巨大な一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

測定値は clock 数だが、分かりやすいように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。従って、この表の値のみで断定するのは危険である。測定プログラムは、全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毎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。従って、数μsec で処理が終了することが望ましい。

1．PC 環境

128 ビットブロック暗号	速度[μsec]
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	3.02(3.03) / 3.03(3.04)

暗号化:最速値(平均値) / 復号:最速値(平均値)

2．サーバ環境

128 ビットブロック暗号	速度[μsec]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

3．ハイエンド環境

128 ビットブロック暗号	速度[μsec]
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)

暗号化:最速値(平均値) / 復号:最速値(平均値)

以上の結果から、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。

SW 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(HW)実装評価

HW 実装評価の対象となった 128bit ブロック暗号は、Camellia、CIPHERUNICORN-A、Hierocrypt-3、MARS、RC6、SC2000 の 6 暗号方式である。これら方式の HW 評価対象のパラメータは、以下の通りである。CIPHERUNICORN-A、SC2000 は、応募時点では、応募書類の中に、「HW による実装も可能」という記載があるだけで、具体的な HW 実装例（規模等）の記載が無いため、HW 実装評価の対象とはしなかった。

今回行ったブロック暗号の HW 実装評価に関しては 2 通りのアーキテクチャが考えられる。つまり、ループ・アーキテクチャを採用する場合と採用しない場合に大別される。ループ・アーキテクチャを採用しなかったグループに属するアルゴリズムは、Hierocrypt-3、MARS、RC6 の 3 方式であり、ループ・アーキテクチャを採用したのアルゴリズムは Camellia のみであった。

これら方式の HW 評価対象のパラメータは、以下の通りである。

評価対象	繰り返し段数	鍵長(bit)
Camellia	24 段	256
CIPHERUNICORN-A	16 段	128
Hierocrypt-3	6 段	128
MARS	32 段	128
RC6	20 段	128
SC2000	19 段	128
Rijndael(参考)	10 段	128

評価結果

回路規模、クリティカルパス遅延、処理速度の評価結果は下表の通りである。

評価対象		回路規模（単位：Gate）			
		データランダム化部	鍵スケジュール部	制御回路	Primitive 全体
Camellia	*2	16,327	22,755	266	39,348
		9,668	13,304	141	23,124
RC6	*1	77,785	975,391	—	1,753,076
SC2000（参考）[2]	—	—	—	—	62,000
MARS	*1	739,069	2,316,846	—	3,055,914
Hierocrypt-3	*1	538,078	106,302	—	724,380
Rijndael(参考)	*1	518,508	93,708	—	612,843

＊1:最適化することを行わず、アルゴリズムの全体を速度重視の設計を想定。パイプラインアーキテクチャは採用しない。

＊2:ループアーキテクチャを採用し、パイプラインアーキテクチャは採用しない。また、置換表はテーブルを用い，論理合成ツールを用いて最適化を行う。上段は速度優先、下段は回路規模優先。

この回路規模による評価結果から、Triple DES との相対的な比較を試みると、ループ・アーキテクチャを採用しない（暗号アルゴリズム全体実装を行う）グループにおける回路規模の比較では、Hierocrypt-3 が Triple DES の約 4.8 倍であり、Rijndael は Triple DES の約 4.1 倍と小型であり、RC6 と MARS の回路規模は、Triple DES の 10 倍を越えている。

ループ・アーキテクチャ（小型アーキテクチャ）を採用したグループでは、Camellia(256bit 鍵)が速度優先ならば Triple DES の約 6 倍､面積優先ならば約 4 倍の回路規模となっている｡この傾向を見る限りは、HW 実装規模からは､今回応募された暗号技術のうち､Camellia(256bit 鍵)､Hierocrypt-3 と Rijndael は、回路規模的には小型の部類であり、RC6、MARS は、回路規模的には大きな部類に属すると見て良いと判断される。

処理速度を規定するクリティカルパス遅延とクリティカルパス遅延から想定される処理速度は以下の通りと評価された。

評価対象		クリティカルパス (ns)	Key Setup Time(ns)	処理速度(Mbps)
Camellia(256bit 鍵)	*2	5.46	—	837
		11.51	—	397
RC6	*1	698.05	2,112.26	183.36
SC2000（参考）	—	—	—	914
MARS	*1	612.64	1,740.99	208.93
Hierocrypt-3	*1	75.55		1,694.24
Rijndael(参考)	*1	65.64	57.39	1,950.03

＊1:最適化することを行わず、アルゴリズムの全体を速度重視の設計を想定。パイプラインアーキテクチャは採用しない。

＊2:ループアーキテクチャを採用し、パイプラインアーキテクチャは採用しない。また、置換表はテーブルを用い，論理合成ツールを用いて最適化を行う。上段は速度優先、下段は回路規模優先。

まずは、ループ・アーキテクチャを採用しない（暗号アルゴリズム全体実装を行う）グループ内での処理速度の比較では、Hierocrypt-3 と Rijndael は Triple DES の約 4 倍を越え、RC6 と MARS は、Triple DES のスループットを越えなかった。

ループ・アーキテクチャ（小型アーキテクチャ）を採用したグループでは、Camellia は Triple DES の約 2.5～3 倍のスループットとなっている。

なお、SC2000 に関しては、ISEC 研究会（2000 年 9 月）にて、HW 実装（0.25μm ルール CMOS-GA）での報告（回路規模とスループット）がなされているので、参考までに記載した。[9]

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事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 の 3 通りの仕様で提案されている。学術的な解読可能段数を 256bit 鍵の仕様に対し示すと以下のようになる。

Camellia(FL 関数無し)	24 段中　7 段	丸め差分攻撃(random permutation との区別)[10]
Hierocrypt-3	8 段中 3.5 段	高階差分攻撃（SQUARE 攻撃）[11]
MARS(Core 部のみ)	16 段中　11 段	丸め差分攻撃[12]
RC6	20 段中　15 段	カイ 2 乗攻撃[13]
SC2000	22 段中　13 段	差分攻撃[14]

[9] 下山、屋並、横山、武仲、伊藤、矢島、鳥居、田中，共通ブロック暗号 SC2000，信学技報 ISEC2000-72,(2000-9)
[10] 渋谷、下山、辻井，Byte-Oriented な暗号に対する Truncated Linear Attack，SCIS2001, 11A-3,(2001)
[11] 大熊、佐野、村谷、本山、川村，ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 の安全性について，SCIS2001, 11A-4,(2001)
[12] J.Kelsey and B.Schneier, MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants, 3rd AES conf.(2000), http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html
[13] L. R. Knudsen and W. Meier, Correlations in RC6 with a reduced number of rounds, Fast Software Encryption Workshop, 2000

Rijndael	14 段中	8 段	高階差分攻撃（SQUARE 攻撃）[15]
		9 段	関連鍵攻撃

CIPHERUNICORN-A に付いては、未知である。この攻撃可能段数は、2001 年 3 月 31 日時点の公開資料及び CRYPTREC 詳細評価に基づくものである。Camellia では、仕様段数と現在知られている攻撃可能段数に比較的大きな違いがみられる。これは提案者の意識する攻撃可能段数の上界値と現在までの攻撃技術との乖離であり、仕様段数を減じて良いことを意味するものではない。128 ビットブロック暗号の多くは、最近提案されたものであり、今後の研究が期待される。大きな乖離を含む現時点のデータを基に、安全性余裕を示すことは適切ではないと考え、安全性余裕対速度の表は作成しない。

[14] 屋並、下山，共通鍵ブロック暗号 SC2000 の差分／線形探索，SCIS2001,12A-2,(2001)

[15] N.Ferguson,et al. , Improved Cryptanalysis of Rijndael, in the preproceedings of the Fast Software Encryption Workshop 2000, April 10-12, 2000.

5.1.3 ストリーム暗号

安全性及び実装評価を総合した観点から、評価結果を下表に示す。

ストリーム暗号の一覧表

名称	MULTI-S01	TOYOCRYPT-HS1
特　徴	MULTI-S01 は、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 K (256 bit)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 は擬似乱数生成器として PANAMA を用いている。	同期型鍵ストリーム暗号であり、鍵ストリーム生成アルゴリズムの生成する擬似乱数系列と平文系列を1bit ずつ EXOR することにより暗号化を実現。 鍵ストリーム生成アルゴリズムは非線型フィルタ・ジェネレータに分類される。 線形フィードバックシフトレジスタの出力を非線形変換関数を介して出力することにより、1bit ずつ乱数値を生成する。 非線形変換関数は、小規模なハードウェアとなるように、算術演算を用いず、論理演算を基本演算として構成。
安全性	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。	提案アルゴリズムに、何がしかの改善を行ってから、実システムには採用すべきであると考える。鍵長は、固定鍵が既知ならば、見かけの 128 bit から実質的には高々96bit に激減する。これは、現在の計算機処理能力では安全な範疇であるが、将来を見越した安全性は運用者が判断する必要がある。
総合評価	ストリーム暗号としての安全性については、今のところ問題は見つかっていない。 現時点では学会等で厳密な評価が得られておらず、継続的な評価が必要である。SW における処理速度は速いグループに属する。	提案アルゴリズムに、何がしかの改善を行ってから、実システムには採用すべきであると考える。HW 実装向きである。

ソフトウェア(SW)実装評価

ストリーム暗号の測定においては、一般的には鍵のセットアップが無いため、データランダム化部の測定のみ行った。ただし、64 ビットブロック暗号と同一の測定プログラムを用いたため、ストリーム暗号の実装性を犠牲にしている場合がある。使用目的からすれば、ストリーム暗号はブロック暗号と比較して HW 指向が強いので、SW 評価よりも HW 評価を主眼にすべきである。従って、ストリーム暗号に対する SW 評価は、SW 実装したとしても最低条件の使用に耐えうる性能を実現しているかの確認を目的とした。

測定値は clock 数だが、分かりやすいようにスループット(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。ブロック暗号と同様、この表の値のみで評価するのは危険である。ストリーム暗号は、HW 指向の強い暗号である。従って SW 実装に不向きなものも多い。本測定は PC 環境のみで測定し、測定プログラムには 64 ビットブロック暗号と同一のものを用いた。このため、実際の性能を充分に引き出していない可能性がある。従って、本測定は最低条件の使用に耐えうる性能を実現しているかどうかの確認を目的とした。

PC 環境

ストリーム暗号	速度[Mbps]
MULTI-S01	237.7(233.7)
TOYOCRYPT-HS1	3.0(3.0)

最速値(平均値)

一般的なストリーム暗号は、平文に乱数列(鍵系列)を排他的論理和して暗号文を得る。復号はこの逆であり、暗号化と復号の速度は全く同じである。従って、本測定では暗号化のみ測定した。MULTI-S01 はこのような一般的なストリーム暗号の構造をしていないので、実際には暗号化と復号で速度に違いが生じる可能性がある。また、MULTI-S01 には MAC 機能が付加されているので、上記結果にはこれも含まれる。

さらに、MULTI-S01 で使用される疑似乱数生成器 PANAMA は初期動作が必要であり、これはブロック暗号の鍵スケジュールに相当するが、本測定にこれは含まれていない。TOYOCRYPT-HS1 は、測定プログラムのインターフェースと合わせるため、シリアル実装されている。本来はパラレル実装を施されるので、この値の約 8 倍程度高速な実装が見込まれると提案者は主張している。両者とも、何らかの制約を本 SW 実装評価において受けているが、期待される SW 実装の最低条件は実現可能であると考えられる。尚、詳細は詳細評価等に譲るが、MULTI-S01 と TOYOCRYPT-HR1 は、その設計方針、機能共に異なっているので本測定による単純な速度比較は無意味である。

SW 実装の性能は､応募者の開発により日々向上している｡現在では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(HW)実装評価

評価方法

アルテラ社の FPGA(Field Programmable Gate Array)上で、C 言語で作成されたプログラムに対して、Verilog HDL により回路記述し、シミュレーションを行った。使用した開発環境は、

- ModelSim VHDL/Verilog Version 5.4e (Model Technology)
- Synplify (Synplicity Inc.)

である。

評価結果

なお、下表の実行速度の見積もりには、鍵データの設定時間は含んでいない。

評価対象		動作周波数 (MHz)	処理速度 (Gbps)	リソース使用量	使用 FPGA
MULTI-S01		18.8	1.203	19,811/42,240 ATOMs(46%)	EP20K1000E
TOYOCRYPT-HS1	回路規模優先	58.1	0.052	11,883/24,320 LCs	EP20K600E
	処理速度優先	45.2	1.446	16,144/24,320 LCs	EP20K600E

ストリーム暗号に関しては、処理速度優先の設計条件では、汎用の FPGA を用いても、リーゾナブルな回路規模で、Gbps クラスの処理速度を実現できると評価された。尚、使用した FPGA については、EP20K1000E の方が EP20K600E よりも大規模な回路規模を実現できる。使用した FPGA を考慮に入れた場合、回路規模の面からは、MULTI-S01 の方が TOYOCRYPT-HS1 よりも大規模となることは否定できない。

参考文献

[1]ICHIKAWA, KASUYA, MATSUI, Hardware Evaluation of the AES Finalists, The Third Advanced Encryption Standard Candidate Conference, April 13-14, 2000

5.2 個別暗号評価

5.2.1 CIPHERUNICORN-E

1 暗号技術

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、全ての項目について、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。

知的財産権 (応募者資料による)

(提案者特許)

特許： 出願平 9－213274

応募暗号技術仕様の公開 Web アドレス

http://www.mesh.ne.jp/hnes/products/security/angou.html

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。鍵スケジューリングは、秘密鍵を攪拌しながら、2624 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(関数鍵とシード鍵) 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(function key)は本流部に、シード鍵(seed key)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）

差分解読法や線形解読法は、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

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。

2 評価結果

2.1 安全性評価

CIPHERUNICORN-E のラウンド関数の構成は非常に複雑であり、差分解読法や線形解読法を始めとする、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。さらに、提案から 3 年以上経過しているにも関わらず、学会等での第三者による外部評価結果がないなど、安全性に関する厳密な理論的評価は得られていない。このため、何段以上ならば期待する暗号強度に達しているのか、どの程度の安全性余裕度があるのかなど、正確には明らかになっていない。

しかし、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、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。また、独立した拡大鍵に影響されていない S-box の線形特性確率を掛け合わせていることによる影響を排除した mF*関数を利用したモデルにおいて、mF*関数の線形特性確率は $2^{-17.68}$ となるものがある。それが mF*

関数の最大線形特性確率になると仮定すれば、8 段での最大線形特性確率の上界が $2^{-70.72}$ となる。実際の暗号に対する正確な暗号強度を得るためには、mF 関数や mF*関数を実際のラウンド関数に置き換え、より詳細な評価を行うことが必要となるが、少なくとも、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。これより、仕様段数が 16 段であることを総合的に考慮すれば、現在の理論的な解読技術によって解読することは不可能と考えられる。したがって、電子政府用の暗号としては安全であると考えられる。

ただし、ラウンド関数の構成上、実装攻撃に対する耐性は高くない恐れがあるので、実装攻撃が想定される環境において利用する場合には防御策を注意深く講じることが望まれる。

A) 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全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。

B) 差分解読法

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。

CIPHERUNICORN-E においても、自己評価書では、(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ている。このモデルに対し、truncated vector 探索を利用して、15 段での最大差分特性確率の上界が 2^{-84} であると評価している。この評価からは、12 段での最大差分特性確率の上界が 2^{-72} になることも求められる。また、同様のモデルに対し、10 段での最大差分特性確率が 2^{-70} 以下との見積もりもある。以上より、mF 関数を利用したモデルでは 12 段以上の差分パスの存在しか認められないことから、CIPHERUNICORN-E の段数が 16 段であることを考慮すれば、差分解読法に対して安全であると考えてよい。また、mF 関数を実際のラウンド関数に置き換え、より詳細な評価を行うことにより、より正確な暗号強度が得られると期待される。

なお、中間に L 関数が入っているものの基本的にビット単位の演算であるので、入力差分値のハミング重みが小さければ L 関数によって差分特性確率が大きく低下する可能性は低く、それほど(特に段数が少ないときの)攻撃可能段数には影響を与えないと考えられる。

C) 線形解読法

差分解読法と同様に、自己評価書では、mF 関数を利用したモデルとして、truncated vector 探索を利用して、ラウンド関数の最大線形特性確率が $2^{-63.90}$、15 段での最大線形特性確率の上界が $2^{-447.30}$ であると評価している。

ここで、mF 関数の最大線形特性確率において、独立した拡大鍵に影響されていない s-box の特性確率を掛け合わせていることによる影響を排除するため、mF 関数における一時鍵生成部及び一時鍵依存関数部分での s-box の特性確率の影響を完全に排除した（線形特性確率が 1 で成立する）mF*関数を利用したモデルを考える。このとき、自己評価書と同じ線形パスにおける mF*関数の線形特性確率は $2^{-17.68}$ となる。これが mF*関数の最大線形特性確率となると仮定すれば、8 段での最大線形特性確率の上界が $2^{-70.72}$ となる。

以上の結果、mF*関数を利用したモデルでも、8 段以上の線形パスが存在することはほとんどないと期待されることから、CIPHERUNICORN-E の段数が 16 段であることを考慮すれば、線形解読法に対して安全であると考えてよい。なお、mF*関数を実際のラウンド関数に置き換え、より詳細な評価を行うことにより、より正確な暗号強度が得られると期待される。

D) 高階差分攻撃、補間攻撃

これらの解読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。

E) 鍵衝突攻撃

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。

F) 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 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。

（実装攻撃に対する安全性）

CIPHERUNICORN-E のラウンド関数は、a) データ依存により構成が変わる部分が存在し、b) 内部構成が、本流部と一時鍵生成部という二系統の処理部分を有しており、同一の入力データが分岐処理される。

一般に、a)のようなデータ依存型の処理ではタイミング攻撃が、また b)のような同じデータが複数の処理を行う場合には電力解析攻撃が有効に働く場合が多いとされていることから、タイミング攻撃や電力解析攻撃などの実装攻撃に対する耐性は高くない恐れがある。したがって、実装攻撃に対する脅威がある環境において利用する場合には、実装攻撃対する防御策を注意深く抗じることが望まれる。

2.2 ソフトウェア(SW)実装評価

(PC 実装)

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	26232 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	“ /O2 /Oy- ”（実行速度）を指定	
1 回目	2 回目	3 回目
1435 / 1438	1434 / 1444	1436 / 1440
1424 / 1426	1422 / 1425	1422 / 1425

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	11848 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	“ -v -fast ”を指定	
1 回目	2 回目	3 回目
1462 / 1469	1462 / 1468	1462 / 1469
1462 / 1468	1462 / 1468	1462 / 1468

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	13552 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	“ -O4 ”を指定	
1 回目	2 回目	3 回目
1575 / 1583	1575 / 1583	1575 / 1583
1566 / 1579	1568 / 1582	1568 / 1580

上段：暗号化 下段：復号、 (最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C +アセンブラ	
プログラムサイズ	26232 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	“ /O2 /Oy- ”（実行速度）を指定	
1 回目	2 回目	3 回目
2421 / 2426	2418 / 2428	2420 / 2424
2406 / 2453	2406 / 2424	2410 / 2414

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	11848Byte (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	
1 回目	2 回目	3 回目
2882 / 2892	2882 / 2890	2883 / 2890
2936 / 2944	2935 / 2944	2935 / 2944

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	13552 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	“ -O4 ”を指定	
1 回目	2 回目	3 回目
2381 / 2393	2381 / 2390	2381 / 2390
2621 / 2634	2619 / 2635	2623 / 2634

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

暗号化及び復号処理、鍵生成まで含めた暗号化及び復号処理の全ての測定項目について、今回応募された 64 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。特に、PentiumIII 上では、全ての測定項目について Triple DES よりも遅い。暗号化・復号処理では Triple DES の約 60%、鍵生成＋暗号化・復号処理で約 80%の処理速度である。

また、IC カードを代表とする、8 ビット CPU 上でのソフトウェア実装に関する記述・公開情報は存在していない。

2.3　ハードウェア(HW)実装評価

提案者はハードウェア実装も可能であると述べているが、ハードウェア実装に関する記述・公開情報は存在していないため、今回は実装評価対象としなかった。

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.

5.2.2 FEAL-NX

1. 暗号技術

1.1 技術概説

FEAL-NX の基本部は 1987 年に EUROCRYPT において清水ら（NTT）によって発表された。FEAL-NX の仕様は、1990 年に宮口ら（NTT）によって発表されたブロック長 64 ビット、鍵長 128 ビットの Feistel 型共通鍵ブロック暗号であり[19]、今回 NTT から電子政府用暗号として応募された。 FEAL-NX は 8 ビットマイクロプロセッサコードを意識して設計されており、特に小型プロセッサ、小容量メモリなどリソースが限定される領域に適するよう設計されている。暗号の発表以来 10 年以上経つ歴史のある暗号であり、これまでに多くの解析がなされたことによって安全性、処理性能共に評価実績が充実している。

知的財産権 （応募者資料による）

（提案者特許とその扱い）

応募暗号技術に関連する提案者するの特許の状況は下記の通り。

No	出願番号（公開番号）	登録番号 （公告番号）	発明の名称	出願人
1	60-250398 (62-109083)	2028919 (7-060292)	データ乱数化処理方法	NTT、NTT データ
2	60-252650 (62-113191)	1992651 (3-033269)	データ拡散装置	NTT、NTT データ
3	62-037231 (63-204289)	1991129 (6-090597)	データ拡散装置	NTT、NTT データ
4	62-232957 (1-074582)	2834450	暗号装置	NTT
5	01-340384 (3-129384)	2825205	暗号装置	NTT
6	EP 86115279.1 US 4,850,019	EP 221538	Data randomization equipment	NTT

なお下記に示した特許等について提案社は、非排他的かつ妥当な条件で他者に実施許諾すると述べている。

応募暗号技術仕様の公開 Web アドレス

http://info.isl.ntt.co.jp/

1.2 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 Feistel 型共通鍵ブロック暗号である。

・Feistel 構造の内部関数である F 関数では、入力 32 ビットを 8 ビット の 4 変数に分割しそれぞれを入力として対して攪拌処理を行い 32 ビットを出力する。

（設計方針）

・統計的にデータ撹拌効果の高い構造を内部関数として採用する。

・Feistel 構造の内部関数では、バイト単位の算術演算と論理演算を用いて攪拌処理を行う。

・拡大鍵生成関数は、8 ビットの算術演算からなる S-box と論理演算で構成する。

・推奨段数は 32 以上の偶数段である。

1.3 その他

1987 年に NTT で開発された、4 段 Feistel 構造を持つ 64 ビット長のブロック暗号 FEAL-4 の段数を、32 段以上とした暗号 FEAL-N が FEAL-NX のベースである[19,20,26,27]。
FEAL-NX の暗号化部分は、FEAL-N と同じであり、FEAL-N の拡大鍵生成部を改良して秘密鍵長を 64 ビットから 128 ビットに拡張した構造となっている。

暗号の基本構造が提案されて以来 10 年以上という長い時間が経つが、その間、ブロック暗号に対する新しい攻撃法や高速実装法が開発された。そのいくつかは FEAL-N に対して適用が試みられるなど、FEAL-N は共通鍵ブロック暗号理論の試金石となることが多く、共通鍵ブロック暗号の歴史を物語る上で重要な位置を占めている[1,5,8,9,10,11,12,13,14,15,16,17,25,28,29,30]。

2. 評価結果

FEAL-NX の推奨段数は 32 以上の偶数段と記述されており段数が一意に定まっていない。応募暗号は段数を含め暗号アルゴリズムに関する全てのパラメータを固定させなければならないと規定されているため、評価委員会では最小推奨段数である 32 段を FEAL-NX の段数として各種評価を行う事とした。

2.1 安全性評価

FEAL-NX のデータランダム化部は FEAL-N と全く同じ構造を用いており、FEAL-N は 64 ビット鍵長であるのに対し、FEAL-NX は鍵スケジュール部に改良を加え 128 ビット鍵長へと拡張されている点においてのみ異なっている。
1991 年に Biham と Shamir らによって発表された FEAL に対する差分解読法を適用することで、31 段の FEAL-N は、2^{64} 個の選択平文暗号文組 (データ量は 2^{63} 個)と 2^{63} 回の計算量を用いることで、有意な確率以上で拡大鍵を推定することが可能である。

この FEAL-N に対する解読法を 128 ビット鍵 32 段 FEAL-NX の前半 31 段に適用し、最後の一段について 32 ビットの拡大鍵を総当たりすることによって、FEAL-NX は 2^{63} 個のデータ量、ならびに 2^{99} 回の計算量を用いることで、拡大鍵を推定することが可能である。

本暗号を電子政府で用いた場合、実際の運用上においては、解読までに必要となる計算量は非現実的であり、この攻撃が成功し何らかの問題が発生するとは現時点では考えられないものの、本攻撃に必要な計算量は、暗号研究者が共通鍵ブロック暗号を攻撃する際に目標の一つとする鍵の総当たりに必要な計算量である 2^{128} 回を下回っており、学術的立場から言えば FEAL-NX が差分攻撃法に対し十分な安全性を持っていると結論づけるには、無理があるように思われる。

また、FEAL-NX で用いられている鍵スケジュール部に関して言えば、2^{128} の鍵空間のうち、2^{32} 個の鍵については、6 段毎に同じ拡大鍵が繰り返されるという結果が報告されており、これらの鍵についてはスライド攻撃が適用可能で、弱鍵となる恐れがある。ただし本攻撃から、実際に暗号化鍵を推定することが可能かどうかについては検討の余地が残っており、さらに 2^{32} という鍵集合は、鍵空間全体 2^{128} に比べると十分小さいため、電子政府用暗号として用いた場合に実用上問題となることはおそらくないものと思われるが、学術的立場から鍵スケジュール部の安全性に問題が無いと言い切ることは難しいものと思われる。

○データランダム化部に対する攻撃

FEAL-NX は、差分攻撃法に対しては、全数探索を下回る計算量で暗号化鍵を推定する手段がある。それ以外の攻撃法、線形攻撃法、高階差分攻撃法、補間攻撃法、mod n 攻撃法に対しては、安全性に問題ないものと思われる。

下記の表は、各段数における最も効率的な攻撃法をまとめたものである。なお 32 段を超える段数に対する攻撃については未検討である。

段数	データ量	必要計算量	手法,発表者,年 [参考文献]
4	16	2^4	既知平文, 栗田 金子,1993 [15]
8	12	2^{33}	差分線型解読,青木ら,1995 [1]
31	2^{63}	2^{63}	差分解読,Biham Shamir,1991 [7]
32	2^{63}	2^{99}	差分解読([7]の改良)

FEAL-NX の攻撃段数と解読必要計算量

○鍵スケジュール部に対する攻撃

2^{128} の鍵空間のうち、2^{32} 個の鍵については、スライド攻撃に対する弱鍵となる恐れがあり、安全性を検討する必要がある。鍵関連攻撃、中間一致攻撃については安全性に問題ないものと思われる。

○実装上の攻撃

タイミング攻撃, 電力差分攻撃、故障利用攻撃等については、実装時に注意する必要がある。

2.2 ソフトウェア(SW)実装評価

(PC 実装)

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	16779 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	/G6 /Gr /Zp16 /MLd /W1 /GX- /Ogx /D "WIN32" /D "_DEBUG" /D "_CONSOLE" /D "_MBCS"\ /Fp"$(INTDIR)\ipafeal.pch" /YX /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c\ /I"C:\Program Files\Microsoft Visual Studio\VC98\Include"	
1 回目	2 回目	3 回目
354 / 365	353 / 365	353 / 366
356 / 372	355 / 372	355 / 372

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	11953 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	/G6 /Gr /Zp16 /MLd /W1 /GX- /Ogx /D "WIN32" /D "_DEBUG" /D "_CONSOLE" /D "_MBCS"\ /Fp"$(INTDIR)\enck.pch" /YX /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c\ /I"C:\Program Files\Microsoft Visual Studio\VC98\Include"	
1回目	2回目	3回目
802 / 823	802 / 823	802 / 823
803 / 824	804 / 824	803 / 824

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

復号の処理時間は暗号化処理時間に比べ1cycle程度の差が見られたものの、ほぼ同じ値であった。

応募者による32ビットプロセッサ上の処理性能値として、PentiumIII(1GHz)上での実装結果が報告されており、その速度は32段FEAL-NXで128Mbpsであったとされている。また、鍵スケジュール部の速度は、実装の最適化を行っていないという但し書きのもとで1.375 μsであったとされている。

今回の評価をPentiumIII(1GHz)に換算すると平均値172Mbps、最速値180Mbps となり、応募時よりさらに高速化がはかられていることがわかる。

応募時における鍵スケジュール部の処理速度を換算すると727 clock cycle/blockであるから、この値に今回計測した暗号化処理のみの速度を加算すると、平均1099 cycle, 最速 1082 cycle となり、応募時よりさらに高速化が計られている。

（スマートカード等への実装評価）

応募者によって、8 ビットプロセッサとして 8MHz の Z80H 上での実装結果、16 ビットプロセッサとして 10MHz の Intel i80286 上での実装結果が示されている。Z80Hにおける速度は18.2kbps, i80286上では220 kbpsとの値が示されている。

2.3 ハードウェア(HW)実装評価

以下の環境でHW実装評価を実施した。評価結果は以下の通りである。

ターゲットデバイス：0.35μm ASICライブラリ
設計記述言語：Verilog-HDL
回路合成：Design Compiler

回路規模(Gate)	データランダム化部	34,830
	鍵スケジュール部	34,840
	Primitive 全体	69,970
クリティカルパス(ns)		227.2
処理速度(Mbps)		281.69

応募者によって、1.5 μm CMOS ゲートアレイで 8 段 FEAL-N を実装した結果、ならびに 0.8μm CMOS ゲートアレイで 32 段 FEAL-N を実装した結果から、それぞれ FEAL-NX の処理速度を予想した評価値が示されている。

テクノロジ	処理速度	クロック数	ゲート数
1.5μm	24 Mb/s	12MHz	3.9KGate
0.8μm	23 Mb/s	12.5MHz	---

ハードウェア評価値を一概に比較することは難しいが、応募者による評価は、テクノロジの進歩とクロック数の向上にもかかわらず、処理速度が落ちていることに疑問を感じさせるものの、過去の実装からの予想値であるという前提を考えればやむを得ないかもしれない。

参考文献

[1] K.Aoki, K.Ohta, Differential-linear cryptanalysis of FEAL-8, IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences Japan, Vol. E79-A, No.1, pp. 20--27, 1996.

[2] K.Aoki, K.Kobayashi, S.Moriai, The best differential characteristic search of FEAL. IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences Japan, Vol. E81-A,No.1,pp.98--104,1998. (Japanese preliminary version was presented at ISEC96-31).

[3] K. Aoki, K. Ohta, S. Moriai, and M. Matsui, Linear cryptanalysis of FEAL, IEICE Transactions Fundamentals of Electronics, Communications and Computer Science, Vol.E81-A, No.1, pp.88--97, 1998.

[4] 青木和麻呂, 太田和夫, 盛合志帆, 共通鍵暗号 FEAL の安全性評価, pp.734- 739, NTT R&D Vol.48, No.10,1999 年 10 月.

[5] B. Den Boer, Cryptanalysis of F.E.A.L., Advanced in Cryptology EUROCRYPT'88, LNCS 330, pp.293--299, 1988.

[6] E.Biham, A.Shamir, Differential Cryptanalysis of DES-like Cryptosystems (extended abstract), CRYPTO'90, 1990.

[7] E. Biham and A. Shamir, Differential cryptanalysis of Feal and N-hash, Advances in Cryptology EUROCRYPT'91, LNCS 547, pp.1--16, 1991.

[8] H. Gilbert and G. Chasse, A statistical attack of the FEAL-8 cryptsystem, Advances in Cryptology Crypto'90, LNCS 537, pp.22--33, 1991.

[9] B. S. Kaliski Jr. and M. J. B. Robshaw, Linear cryptanalysis using multiple approximations and FEAL, Fast Software Encryption, Second International Workshop, LNCS 1008, pp.249--264, 1995.

[10] Walter Fumy, Siemens AG, On the F-function of FEAL, CRYPTO'87, 1987.

[11] H.Gilbert, G.Chasse, A Statistical Attack Of The FEAL-8 Cryptosystem, CRYPTO'90, 1990.

[12] 金子敏信, 既知平文攻撃による FEAL-4 の解読, ISEC91-25,1991.

[13] T.Kaneko, A known-plaintext attack of FEAL-4 based on the system of linear equations on difference, IEICE Transactions Fundamentals of Electronics, Communications and Computer Science, Vol.E76-A,No.5, pp.781--786, 1993.

[14] 栗田大, 金子敏信, 差分方程式を用いた FEAL-6 の既知平文攻撃, 信学会秋季大会 A-190, 1992.

[15] M. Kurita, T. Kaneko, A known plaintext attack of FEAL-4, SCIS93-3B,1993.

[16] 増田 孝志, 金子 敏信, FEAL 暗号方式の解読における松井法と差分方程式の関係, SCIS94, 1994.

[17] M. Matsui and A. Yamagishi, A new method for known plaintext attack of FEAL cipher, IEICE Transactions Fundamentals of Electronics, Communications and Computer Science Vol. E77-A, No.1, pp.2--7,1994.

[18] Mitsuru Matsui, On correlation between the order of S-boxes and the strength of DES, Advances in Cryptology EUROCRYPT'94, volume 950 of Lecture Notes in Computer Science, pp.366--375. Springer-Verlag, 1995.

[19] 宮口庄司, 磐田雅彦, 太田和夫, FEAL の仕様拡張, Proceedings of SCIS90, SCIS90-14E, 1990.

[20] S.Miyaguchi, The FEAL Cipher Family, CRYPTO'90, 1990.
[21] 宮口庄司, 森田光, 藤岡淳, FEAL 暗号とそのアプリケーション, 電子情報通信学会論文誌, 情報社会における通信網の安全・信頼性シンポジウム, pp.29--34, Aug, 1991.
[22] S.Moriai, K.Aoki, K.Ohta, The best linear expression search of FEAL, IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences (Japan), Vol. E79-A, No.1, pp.2-11, 1996.
[23] 森田光, 宮口庄司, FEAL-LSI とその応用, NTT R&D, Vol.40, No.10, pp.1371-1380, Oct. 1991.
[24] Hikaru Morita,Kazuo Ohta,and Shoji Miyaguchi, Results of switching- closure- test on feal, ASIACRYPT'91, volume 739 of Lecture Notes in Computer Science, pp.247-252, 1993.
[25] S. Muphy, "The cryptanalysis of FEAL-4 with 20 chosen plaintexts," Journal of Cryptology, Vol.2, No.3, pp.145--154, 1990.
[26] 清水明宏, 宮口庄司, 高速データ暗号アルゴリズム FEAL, 電子情報通信学会論文誌, Vol. J70-D, No.7,pp.1413--1423,July 1987.
[27] A.Shimizu, S.Miyaguchi, Fast Data Encipherment Algorithm FEAL, Advances in Cryptology EUROCRYPT'87, volume 304 of Lecture Notes in Computer Science, pp.267--278. Springer Verlag, Berlin, 1988.
[28] A.Tardy-Corfdir and H. Gilbert, A known plaintext attack of FEAL-4 and FEAL-6, Advances in Cryptology Crypto'91, LNCS576, pp.172--182, 1992.
[29] 角尾幸保, 岡本栄司, 土井洋,既知平文による FEAL-4 の解析的攻撃と FEAL-4 の改良,SCIS93, 3A, 1993.
[30] Y.Tsunoo, E.Okamoto, T.Uematsu, Ciphertext Only Attack for One-way function of the MAP using One Ciphertext, CRYPTO'94, 1994.

5.2.3 Hierocrypt-L1

1 暗号技術

1.1 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、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 4×4MDS 行列を 2 個並列に並べたバイト置換層(MDS_L)、GF(2^{32})の 2×2MDS 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。段関数の一段は 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、段関数を 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

知的財産権 (応募者資料による)

(提案者特許とその扱い)

特願 2000-060482「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」

(2000/03/06 出願)

特願 2000-198478「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録媒体」

(2000/06/30 出願)

特願 2000-210484「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」

(2000/07/11 出願)

特願 2000-211686「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」

(2000/07/12 出願)

特願 2000-212175「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」

(2000/07/13 出願)

提案者は、Hierocrypt-L1 を非排他的かつ妥当的な条件で他社に実施許諾する方針である。

応募暗号技術仕様の公開 Web アドレス

http://www.toshiba.co.jp/rdc/security/hierocrypt/

1.2 技術仕様

（入出力鍵サイズ）

入出力サイズ：64 ビット

鍵サイズ：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／線形解読法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。

1.3　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。両者共、データランダム化部のみの復号処理速度は暗号化のそれより僅かに遅い。

2.　評価結果

2.1 安全性

6 段（12 層）中 7 層までは SQUARE 攻撃が可能であり、特に SQUARE 攻撃の拡張には今後注意する必要があると思われるが、現在のところ脅威となる攻撃方法は見つかっていない。

(解読可能段数)

Ferguson らの手法 1 [2]に Type1 の拡張[3]を適用することで、6 段（12 層）中 6 層まではショートカット可能である。その際に必要な選択平文数は 2^{35} であり、計算量は 2^{72} 回の段関数計算量と同等である[5]。さらに、もう一段分の鍵を推定することにより 7 層までは攻撃が可能である。その際に必要な選択平文数は 2^{37} であり、計算量は 2^{117} 回の段関数計算量と同等である。

具体的には、左(あるいは右)4 バイトのみ活性なΛ集合を与えると、4 層目の入力は 8 バイト全てが活性なΛ集合となり、5 層目の入力はバランスする。7 層目の出力からさかのぼり 5 層目の入力の各バイトがバランスしているか否かを判定することにより 5 層から 7 層の関連する鍵の妥当性を検証できる。

(安全性の根拠)

自己評価書及び詳細評価において最大差分／線形特性確率は、活性 S-box 数の下限より、2 段（4 層）で 2^{-90} を超えないことが示されている。最大差分／線形確率は、各段の鍵が独立かつ一様との仮定において Hong らの手法[4]により 2 段以降 2^{-48} を超えないことが証明可能である。また、3 段以上の段数においても、

現在証明されている上限は 2^{-48} である。なお、自己評価書で求められている 3 段以上の最大差分／線形確率は、自己評価書において説明されているように正確な値ではなく証明可能な数値ではない。3 段以降の最大差分／線形確率の上限が 2^{-48} より小さくなることが必ずしも言えない理由は以下のとおりである。3 段以降の Hierocrypt-L1 は最大差分確率が 2^{-48} である（bijective な）S-Box が直列につながっていると考えることができる。最大差分確率が 2^{-48} である S-Box を直列に複数つなげたとしても最大差分確率が 2^{-48} より小さくなることは必ずしも言えない。

Truncated 差分に関しては、6 段中 3 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、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としている。

2.2 ソフトウェア(SW)実装評価

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ(486 命令)	
プログラムサイズ	52982 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	VC++6.0 速度優先オプション使用	
1 回目	2 回目	3 回目
199 / 201	199 / 201	200 / 201
204 / 206	204 / 206	204 / 205

Ultra SPARC IIi (400MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	24496 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 -native -fast - xarch=v9 - xCC	
1 回目	2 回目	3 回目
378(332) / 380(336)	378(332) / 380(336)	378(332) / 380(336)
500(304) / 504(307)	500(304) / 504(308)	500(304) / 504(308)

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	84328 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 -03	
1回目	2回目	3回目
210(179) / 214(182)	210(179) / 214(182)	210(179) / 213(182)
210(179) / 212(182)	210(179) / 212(182)	210(179) / 212(182)

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ(486 命令)	
プログラムサイズ	52982 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	VC++6.0 速度優先オプション使用	
1回目	2回目	3回目
374 / 375	374 / 377	374 / 375
616 / 618	616 / 617	616 / 618

Ultra SPARC IIi (400MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	24496 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 -native -fast -xarch=v9 - xCC	
1回目	2回目	3回目
718(616) / 721(620)	718(616) / 721(619)	718(616) / 721(620)
1203(1014) / 1215(1031)	1203(1012) / 1215(1030)	1203(1015) / 1215(1031)

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	84328 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 -03	
1回目	2回目	3回目
390(386) / 394(389)	390(386) / 394(389)	390(386) / 394(389)
625(617) / 654(648)	625(617) / 653(648)	625(617) / 653(648)

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

備考：Ultra SPARC IIi と Alpha21264 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。

応募者からは以下の自己評価が報告されている。
CPU: PentiumIII　550MHz
1次キャッシュ:32KB、2次キャッシュ:512KB
RAM: 256MB
OS: Windows 2000 Professional build 2195
速度
鍵生成：3.07Mkeys/sec、179.0 cycles
暗号化：139.13 Mbps、253.0 cycles
復号　：67.56 Mbps、521.0 cycles

機種: JT6N5
プロセッサ: Z80 (5MHz)
ROM: 48KB
RAM: 1KB
EEPROM 8KB
記述言語: Z80 アセンブラ

メモリ使用量
ROM: 26 bytes
RAM: 2,447 bytes
速度
暗号化：3.88 ms

2.3　ハードウェア(HW)実装評価
以下の環境で HW 実装評価を実施した。評価結果は以下の通りである。

評価環境：三菱 0.35μm CMOS ASIC ライブラリ
回路記述：Verilog-HDL
Synthesizer：Design Compiler

回路規模(Gate)	データランダム化部	278,130
	鍵スケジュール部	95,397
	制御回路部	—
	Primitive 全体	373,526
クリティカルパス(ns)		70.13
処理速度(Mbps)		912.59

その他：クリティカルパス長の短縮（処理速度向上）を重視し、回路規模は大きくても構わないとした場合の実装評価である。また、クリティカルパスに鍵スケジュールは含まれていない

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。

設計環境: SYNOPSYS 社製 Design Compliler 1999.10-3
シミュレーション条件: 1.35V 70℃ (標準ケースでは、1.5V25℃)
スループット: 586 Mbps (128.2MHz,14clock,6 ラウンド)

参考文献

[1] 大熊、佐野、村谷、本山、川村, ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 の安全性について, SCIS2001 , 2001
[2] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner and D. Whiting, Improved Cryptanalysis of Rijndael, available from http://www.counterpane.com/rijndael.html 2000
[3] J. Daemen, L.R. Knudsen and V. Rijmen, The block cipher SQUARE, Fast Software Encryption : LNCS 1267, pp.149-165, 1997
[4] S.Hong, S.Lee, J.Lim, J.Sung, and D.Cheon, Provable Security against Differential and Linear Cryptanalysis for the SPN Structure, FSE2000, 2000.4
[5]村谷、大熊、本山、川村, 64 ビット版 Hierocrypt の提案, 情報処理学会研究報告 CSEC11-9, 2000/09

5.2.4 MISTY1

1. 暗号技術

1.1 技術概説

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、暗号化鍵長 128 ビットの共通鍵ブロック暗号であり[5,6]、今回、三菱電機から電子政府用暗号として応募された。MISTY1 は Feistel 構造と鍵依存線形変換を用いており、Feistel 構造の内部関数には、変形 Feistel 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[3,4]。 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エ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 68cycle/block) [7,8,9,10,11]。さらにハードウエアでは 10K ゲート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 年が経過しており、豊富な実績をもつ暗号である。

知的財産権(応募者資料による)

（応募者特許とその扱い）

特許番号　特許第 2025358 号　データ変換装置およびデータ変換方法

本特許に関しては応募者指定の契約書に同意頂く事を条件として、無償で実施許諾をしている。

無償許諾の方針に関しては 下記の URL において公開されている。

応募暗号技術仕様の公開 Web アドレス

http//www.security.melco.co.jp

1.2 技術仕様

(大まかな構造)

・ブロック長 64 ビット、暗号化鍵長 128 ビットのブロック暗号である。

・Feistel 構造と鍵依存線形変換(FL 関数)を用いており、Feistel 構造の内部関数には、変形 Feistel 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法と線形攻撃法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エ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。
・ハードウエ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エ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エアで最適化されるように考慮されている。

1.3 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構造とその安全性に関する理論、並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、Algorithm1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[3,4]。
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[5,6]。

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[15]。

2. 評価結果

2.1 安全性評価

MISTY1のデータランダム化部は3段で最大平均差分・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、線形攻撃法、差分攻撃法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。

実装型の攻撃について言えば、原理的にはほぼ全てのブロック暗号に適用可能であるため、本暗号についても実装時に注意した方がいいと考えられる。また一般的にICカード上に暗号を実装する場合には、既に対策が知られているS-boxに関してだけでなく、鍵依存の線形関数であるFL関数等についてもについて電力解析攻撃が有効とならないようソフトウエア実装時に配慮した方がよいと思われる。

○ データランダム化部

線形攻撃法、差分攻撃法、 丸め差分攻撃法、カイ2乗攻撃法、分割攻撃法、高階差分攻撃法、補間攻撃法、不能差分攻撃法、mod n攻撃法、非全単射攻撃法、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6段以下)であれば、改良型高階差分攻撃法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[13,14]。しかし本報告より、MISTY1そのものの安全性に影響を与えることはないと考えられる。

攻撃段数と解読必要計算量

段数	MISTY1		FL 関数を除いた MISTY1	
	データ量	計算量	データ量	計算量
4 段	$2^{8.4}$	2^{85}	2^{5}	$2^{6.5}$
5 段	$2^{38.4}$	2^{116}	$2^{10.5}$	2^{17}
6 段	-	-	$2^{10.5}$	2^{93}

○ 鍵スケジュール部

全数探索法、弱鍵・準弱鍵、関連鍵解読法、スライド解読法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 $K_1,K_2,...,K_8$ 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 $K_1=K_2=...=K_8$ が成り立つ場合には、全ての段の拡大鍵が等しくなる。この性質より、全秘密鍵 2^{128} 個中 $K_1=K_2=...=K_8$ 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。

○ 実装に関する攻撃法

タイミング攻撃については、ソフトウエア、ハードウエアともに、対策は必要ないか、あるいは極めて容易であると考えられる。電力解析攻撃については、一般的に知られている S-box への攻撃以外に、FL 関数ついても拡大鍵に依存して使用電力量が変化する可能性があり、IC カードへのソフトウエア実装時には注意した方がよいと考えられる。

2.2 ソフトウェア(SW)実装評価

（PC 実装）

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650Mhz)		
言語	アセンブラ	
プログラムサイズ	21353 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		
1 回目	2 回目	3 回目
213 / 215	213 / 215	213 / 214
208 / 210	208 / 210	209 / 211

Alpha21264 (463Mhz)		
言語	アセンブラ	
プログラムサイズ	15632 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		
1 回目	2 回目	3 回目
203 / 205	203 / 206	203 / 205
206 / 208	206 / 208	206 / 208

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650Mhz)		
言語	アセンブラ	
プログラムサイズ	17681 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		
1 回目	2 回目	3 回目
357 / 358	357 / 358	357 / 358
350 / 351	350 / 351	350 / 351

Alpha21264 (463Mhz)		
言語	アセンブラ	
プログラムサイズ	10088 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		
1 回目	2 回目	3 回目
334 / 338	337 / 338	334 / 338
337 / 340	337 / 340	337 / 340

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

復号の処理時間が暗号化処理時間に比べ最大で 5cycle 程度の増減が見られるが、概ね同じ値であった。

応募者による実装例として、PentiumIII(800MHz) と Alpha21264(667MHz) 上でのアセンブリ言語による速度評価結果が示されており、それぞれ 193 cycle/blcok, 192 cycle/blcok であるとしている。

（IC カード実装他）

応募者による実装例として、組み込み用 16 ビットマイコン M16C (20MHz)による実装、8 ビットマイコン H8/300 (3.57MHz) による実装が示されている[11]。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[12]。暗号化、鍵スケジュール速度の単位はいずれも cycle/block である。

プロセッサ	暗号化、鍵スケジュール速度	ROM	RAM
M16C H8/300 Z80	1877,743 6018,1240 25486(鍵スケジュール込)	3400byte 1900byte 1598Byte	64byte 43byte 44byte

2.3 ハードウェア(HW)実装評価

以下の環境でハードウエア実装評価を行った。評価結果は以下のとおりである。

記述言語：VHDL, シミュレータ：ModelSim5.4a, デザインライブラリ：0.25 μCMOS ASIC Design Library, 論理合成ツール：Design Compiler . 2000.05-1

回路規模(Gate)	データランダム化部	*1	19,935
		*2	10,609
	鍵スケジュール部	*1	44,773
		*2	28,194
	制御回路部	*1	94
		*2	68
	Primitive 全体	*1	64,809
		*2	38,875
クリティカルパス(ns)		*1	11.86
		*2	24.70
処理速度(Mbps)		*1	600
		*2	288

*1 スピード優先にて論理合成

*2 規模優先にて論理合成

応募者による実装例として、Mitsubishi 0.35 micron CMOS ASIC Design Library を用いて、シミュレートした結果が示されている[1,2]。

実装 1 : 7.6K ゲート 72 Mbps,

実装 2 : 50K ゲート 800Mbps

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, SCIS97-9.D, 1997.

[3] 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.

[4] Matsui,M., 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.

[5] 松井充, ブロック暗号アルゴリズム MISTY, 信学技報 ISEC96-11, 1996.

[6] Matsui,M., 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.

[7] 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.

[8] 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.

[9] Junko Nakajima, Mitsuru Matsui, Fast Software Implementation of MISTY1 on Alpha Processors, IEICE Trans. Funcationals, Vol E82-A, No.1 January 1999.

[10] 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.

[11] 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.

[12] F. Sano, M. Koike, S. Kawamura, M. Shiba, Performance Evaluation of AES Finalists on the High-End Smart Card, 3rd AES Conference, New York, 2000.

[13] Tanaka,H., Hisamatsu,M., Kaneko,T., Higher Order Differential Attack of MISTY1 without FL functions, JWIS ' 98, ISEC98-66, pp.143-150, 1998.

[14] 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.

[15]3GPP, KASUMI, ETSI/SAGE Specification, 1999.
(http://www.etsi.org/dvbandca/3GPP/3gppspecs.htm)

[15] 3GPP, KASUMI, ETSI/SAGE Specification, 1999.
(http://www.etsi.org/dvbandca/3GPP/3gppspecs.htm)

5.2.5 Triple DES

1. 暗号技術

1.1 技術概要

DESは、1979年にIBMのTuchmanにより提案された共通鍵ブロック暗号である[1]。Triple DESは、1977年に米国政府標準(FIPS：Federal Information Processing Standard)の暗号として認定されたDES（Data Encryption Standard）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共に規定され、FIPS化（FIPS46-3）も行われている[2][3]。

知的所有権　(応募者資料による)

(提案者特許とその扱い)

DESに関する特許は米国や日本において成立しているが、現在では米国、日本ともに失効している。

米国　Patent Number 3,962,529

（出願日：1975年 2月24日　発効日：1976年 6月 8日　失効日：1993年 6月 8日）

日本　公告番号 昭59-45269

（出願日：1976年 2月18日　公告日：1984年11月5日　失効日：1996年 2月18日）

1.2 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Kによる暗号化及び復号処理をE_K及びD_Kとすると、暗号化処理及び復号処理は次のように表すことができる。

暗号化: $C = E_{K3}(D_{K2}(E_{K1}(P)))$

復号　: $P = D_{K1}(E_{K2}(D_{K3}(C)))$

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[2]。

①K_1、K_2、K_3が独立

②K_1とK_2が独立で、$K_1 = K_3$

③$K_1 = K_2 = K_3$

特に③は､3つの鍵を全て同じとすることで､通常のDESとの間で互換性がとられている｡一般に①はTriple DES（3−Key)、②はTriple DES（2−Key）と呼ばれる。DESの有効鍵サイズが56bitであることから、①～③の鍵サイズは各々、168bit、112bit、56bitである。

Triple DESの利用モードはANSI X9.52の中で、ISO8372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つの利用モードが規定されている[2]。

1.3 その他

発表当初からDESの56ビットという鍵長は短かく、鍵の総当り攻撃に対して安全ではないという懸念が出されていた[5]。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回繰り返しでなく3回繰り返しにしているのは、中間一致攻撃（meet-in-the-middle attack）[4]を避けるためである。DESは発表から20年後の1997年に米国RSA社が主催する解読コンテスト（DES Challenge-I）で解読に成功され、現在は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[6]。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。

2. 評価結果

2.1 安全性

Triple DES（①3-key Triple DES, ②2-key Triple DES, ③DES）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5-1に示す。

DESは代表的なShort Cut Methodである差分解読法や線形解読法に対して、鍵の全数探索法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、DESの全数探索法に対する計算量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[6]、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。

Triple DES（2-key）およびTriple DES（2-key）は代表的なShort Cut Methodである差分解読法や線形解読法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、鍵の全数探索法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。特にTriple DES（2-key）は、2^{57}程の計算量（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DESの全数探索法の2倍程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、Triple DES（3-key）も$2^{108.2}$程の計算量（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、Triple DESを電子政府向け暗号として使用する場合は、Triple DES（3-key）であれば当面の間の使用は問題ないと言える。

表 5-1　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	Triple DES(2-key)	Triple DES(3-key)
Brute Force Method			
全数探索法	2^{56}	2^{112}	2^{168}
Merkle-Hellman 中間一致攻撃		2^{57}（選択平文数 2^{56}）	2^{112}（選択平文数 2^{56}）
Lucks による攻撃		―	$2^{108.2}$
Oorshot-Wiener 既知平文攻撃	―	$2^{120-\log_2 N}$（既知平文数N）	―
Short Cut Method			
差分解読法	2^{37}（選択平文数 2^{47}）	＜最大差分特性確率 $2^{-162.3}$ 以下＞[2]	
線形解読法	2^{42}（既知平文数 2^{43}）	＜最大線形特性確率 $2^{-134.7}$ 以下＞[3]	
関連鍵攻撃[4]	―	―	2^{56} ～ 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。

以下、これらに関してもう少し詳しく示す。

（1）Brute Force Method に対する安全性

Triple DES（2-key, 3-key）に関しては鍵の全数探索法に対して現時点で十分安全であると考えられている。DESの56ビット鍵に関しては、1997年に米国RSA社が主催する解読コンテスト（DES Challenge-I）で解読に成功し、現在は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[6]、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、Triple DES（2-key）の場合は、2^{57}（全数探索法では 2^{112}）、Triple DES（3-key）の場合は 2^{112}（全数探索法では 2^{168}）と、全数探索法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[7]。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、

Triple DES（3-key）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、Triple DES（2-key）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[10]。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[8]。

（２）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解読法と線形解読法がある。DESは、差分解読法によって、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[11]。また、線形解読法によって、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[12]。

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／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、Triple DES（3-key）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[13]。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}～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、Triple DES（2-key）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。

2.2　ソフトウェア(SW)実装評価

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650Mhz)		
言語	アセンブラ	
プログラムサイズ	44385 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		
1 回目	2 回目	3 回目
854 / 856	854 / 857	854 / 856
854 / 856	854 / 856	854 / 857

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650Mhz)		
言語	アセンブラ	
プログラムサイズ	44679 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		
1回目	2回目	3回目
1963 / 1967	1967 / 1971	1963 / 1967
1971 / 1975	1971 / 1975	1971 / 1975

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

2.3 ハードウェア(HW)実装評価

Triple DES の HW 実装評価としては、市川らによって ASIC による高速実装結果が報告されている[14]。以下にそれらの評価結果を示す。

＜評価環境＞ ○評価対象： ASIC（三菱電機製 0.35μルール ASIC ライブラリ）

○記述言語： Verilog－HDL

○評価条件： Worst ケース

＜評価結果＞ ●ゲートサイズ（NAND ゲート換算）：

トータル：148147 Gate（暗号化＆復号部 124888／鍵スケジュール部 23207）

●スループット：

407.4［Mbps］

参考文献

[1]W.Tuchman, Hellman Presents No Shortcut Solutions to DES, IEEE Spectrum, 16(7),pp.40-41,1979.

[2]American National Standards Institute,X9.52-1998, Triple Data Encryption Algorithm Modes of Operation, 1998.

[3]National Institute of Standards and Technology, Data Encryption Standard, Federal Information Processing Standards Publication 46-3,1999.

[4]W.Diffie and M.E.Hellman, Exhaustive cryptanalysis of the NBS data encryption standard, Computer,10,6,pp.74-84,June 1977.

[5]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 年.

[6]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 年.

[7]R.C.Merkle and M.Hellman, On the Security of Multiple Encryption, Communications of the ACM, Vol.24,No.7,1981,pp.465-467.

[8]K.Kusuda and T.Matsumoto, A Strength Evaluation of the Data Encryption Standard, IMES Discussion Paper, No.97-E-5, Institute for Monetary and Economic Studies, Bank of Japan,1997.

[9]S.Lucks, Attacking Triple DES, proceedings of Fast Software Encryption'98, LNCS, Vol.1372, 1998,pp.239-253.

[10]P.C.van Oorschot and M.J.Wiener, A known plaintext attack on two-key triple encryption, Advances in Cryptology - Proceedings of EUROCRYPT090, LNCS, Vol.473, Springer-Verlag, 1990, pp.318-325.

[11]E.Biham and A.Shamir, Differential Cryptanalysis of the Full 16-round DES, Advances in Cryptology - Proceedings of CRYPTO92, LNCS, Vol.740, Springer-Verlag, 1993,pp.487-496.

[12]M.Matsui, Linear Cryptanalysis Method for DES Cipher, Advances in Cryptology Proceedings of EUROCRYPT93, LNCS, Vol.765, Springer-Verlag, 1994,pp.386-397.

[13]J.Kelsey, B.Schneier, and D.Wagner, Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple DES, Advances in Cryptology CRYPT96, LNCS, Vol.1109, pp.237-251, Springer-Verlag,1996.

[14]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 Technology, Gaithersburg, MD, April 13-14,2000, pp.279-285.

5.2.6 Camellia

1. 暗号技術

1.1 技術概説

Camellia は日本電信電話（株）と三菱電機（株）の研究者によって共同開発された 128 bit 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（192/256 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、構造の同型性を崩している。

安全性と実装性とのバランスを重視した設計であり、SW と HW の両面での効率的な実装を目指している。とくに HW 実装ではゲート数当りの暗号化/復号処理速度(22.0 Mbits/(s.Kgates))およびゲート数（約 10Kgates）で現時点の世界最小グループに属する。

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、高速暗号通信から計算機資源に乏しいスマートカードまでの幅広い分野が考えられる。

キーワード：HW 小型実装、逆数関数、8 ビット S-box＋論理演算、1 段 SPN

知的財産権 (応募者資料による)

（提案者特許とその扱い）

- 出願番号(2000-064614)
- 出願日 2000 年 3 月 9 日
- 発明の名称「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」
- 出願人「NTT、三菱電機（株）」

非排他的かつ妥当な条件で他者に実施許諾するという方針である。

（関連する他社の特許）

特になし。

応募暗号技術仕様の公開 Web アドレス

http//info.isl.ntt.co.jp/camellia/index-j.html

1.2 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 FL/FL^{-1} 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および復号関数）

（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 構造と FL/FL^{-1} 関数により構成されている。Feistel 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 通りの 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、初期及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 から 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。これは構造の同型性を崩す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 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 暗号の復号は、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）

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 Ka と Kb を生成する。但し、Kb 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 バイトの RAM、192/256 ビット鍵で約 64 バイトの 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、つまり最大平均差分特性確率/最大平均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、不能差分利用攻撃、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。

1.3 その他

Camellia 暗号[3]の開発設計においては、いくつかの暗号技術が NTT 独自の暗号技術 E2[1]と三菱電機独自の暗号技術 MISTY[2]を土台にして開発設計が行われている。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)、HW での実装性能の向上にあると考えられる。

2. 評価結果

2. 1 安全性評価

スクリーニング評価結果および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。特に差分解読法や線形解読法に対しては、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、補助関数 FL/FL^{-1} を除いた 7 段変形 Camellia 暗号に対して攻撃に有効な特性が見出されている。[4])詳細評価結果の概要は以下の通りである。

・FL/FL^{-1} 関数を除いた変形 Camellia 暗号の 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バイトを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 Camellia６段で秘密鍵総当りよりも短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、秘密鍵総当り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 暗号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。

・鍵生成部の特性として秘密鍵 5 バイトと中間鍵 6 バイトから不明な秘密鍵１バイトを計算できる場合が存在した。

差分解読法や線形解読法の他、丸め差分/線形解読、高階差分解読、不能差分利用解読、補間解読、線形和解読、スライド解読などに関しても、安全性に関する問題は見つかっていないと判断できる。

2. 2 ソフトウェア(SW)実装評価

（PC 実装）

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	29285 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	/G6 /ML /O2 /Ob2 /Og /Oi /Ot /Ox /Oy /Gr /I "C:\Program Files\Microsoft Visual Studio\VC98\Include"	
1回目	2回目	3回目
326 / 327	326 / 327	326 / 327
326 / 328	326 / 327	326 / 327

Ultra SPARC IIi (400MHz)		
言語	アセンブラ	
プログラムサイズ	15240 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	-fast - xtarget=ultra -xarch=v9a	
1回目	2回目	3回目
355 / 360	355 / 358	355 / 357
355 / 357	355 / 358	355 / 357

Alpha21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	31552 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	-O -arch ev6	
1回目	2回目	3回目
282 / 288	282 / 289	282 / 288
282 / 288	282 / 288	282 / 289

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	20110 Byte (暗号化/鍵スケジュール含む) 20236 Byte (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 "C:\Program Files\Microsoft Visual Studio\VC98\Include	
1回目	2回目	3回目
467 / 487	467 / 487	467 / 487
474 / 493	474 / 494	474 / 493

Ultra SPARC IIi (400MHz)		
言語	アセンブラ	
プログラムサイズ	23992 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	-fast – xcrossfile -xtarget=ultra – xarch=v9a	
1回目	2回目	3回目
403 / 408	403 / 407	403 / 408
403 / 407	403 / 407	403 / 408

Alpha21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	25792 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	-O -arch ev6	
1回目	2回目	3回目
448 / 454	448 / 454	448 / 455
435 / 439	435 / 439	435 / 439

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

応募者による実装例として、32ビットPC実装に対しては、Java実装で24.07Mbits/s、アセンブラ言語による最適化実装ではPentiumIII(800MHz)上で276Mbps、Pentium Pro(200MHz)上で308 cyclesを実現している。

（ICカード実装）

Z80によるICカード実装結果を示す。（但し、128ビット鍵の場合、提案者よりのデータ）

・処理速度

鍵生成：5,146States

暗号化：28,382States

・メモリ

ROM：1,698bytes

RAM：62bytes

2. 3　ハードウェア(HW)実装評価

以下の環境でHW実装評価を実施した。評価結果は以下の通りである。但し、評価は256ビット鍵の暗号化回路に対するものであり、Verilogで設計し、同一の記述条件で速度優先の条件と面積優先の条件で合成することにより行った。

記述言語 Verilog-HDL

シミュレータ VCS5.1

デザインライブラリ 0.25 μCMOS ASIC Design Library
論理合成ツール Design Compiler . 2000.05-1
動作条件 0 ～70 ℃, 3.3V ±5 %

ハードウェアの評価結果を以下に示す。

回路規模(Gate)	データランダム化部	*1	16,327
		*2	9,668
	鍵スケジュール部	*1	22,755
		*2	13,304
	制御回路部	*1	266
		*2	141
	Primitive 全体	*1	39,348
		*2	23,124
クリティカルパス(ns)		*1	5.46
		*2	11.51
処理速度(Mbps)		*1	837
		*2	397

*1 スピード優先にて論理合成

*2 規模優先にて論理合成

これらの値は鍵長の違いを考慮すると妥当なものと考えられる。

ASIC と FPGA での実装が 128 ビット鍵に対して提案者によって検討されている。それによれば、暗号化/復号処理回路と鍵生成回路を約 10K ゲートで実装している。(0.35μm CMOS ASIC)
スループット：212.2Mbits/s

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, LNC, pp54-68,1997

[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, SAC2000pp.41-54, 2000 （日本語版は、128 ビットブロック暗号 Camellia, 信学技法 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

5.2.7 CIPHERUNICORN-A

1. 暗号技術

1.1 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 bit、鍵長 128, 192, 256 bit の 128 ビットブロック暗号[1]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。

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、全ての項目について、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。

知的財産権　(応募者資料による)

(提案者特許とその扱い)

特許：　出願平 9－213274

商標：　登録番号　第 4221077 号

著作物：　CIPHERUNICORN-A のプログラム

暗号評価の目的に必要な、発明、著作物及び商標の実施、複製・領布について、無償での利用を許諾。

応募暗号技術仕様の公開 Web アドレス

http://www.mesh.ne.jp/hnes/products/security/angou.html

1.2 技術仕様

ブロック長 128 bit、鍵長 128, 192, 256 bit、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。

鍵スケジューリングは、秘密鍵を攪拌しながら、2304bit 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(関数鍵とシード鍵) 32 ビット×4 (合計 128 ビット) を用いた 64 ビット入出力関数であり、s-box、32 ビット算術加算、32 ビット定数算術乗算、ローテーションにより構成される。なお、この関数は全単射関数ではない。

関数内部では、64 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(function key)は本流部に、シード鍵(seed key)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 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）

鍵スケジュール部は、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 関数及び 32 ビット定数算術乗算を利用する。

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）

差分解読法や線形解読法は、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

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、64 ビットブロック暗号である CIPHERUNICORN-E がある。

2 評価結果

2.1 安全性評価

（総評）

CIPHERUNICORN-A のラウンド関数の構成は非常に複雑であり、差分解読法や線形解読法を始めとする、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、現時点では、何段以上ならば期待する暗号強度に達しているのか、どの程度の安全性余裕度があるのかなど、正確には明らかになっていない。

しかし、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 段以上では最大差分特性確率の上界が、また 14 段以上で

は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。実際の暗号に対する正確な暗号強度を得るためには、mF 関数を実際のラウンド関数に置き換え、より詳細な評価を行うことが必要となるが、少なくとも、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。このことに加え、仕様段数が 16 段であることを総合的に考慮すれば、現在の理論的な解読技術によって解読することはほぼ不可能であると期待される。したがって、電子政府用の暗号としては安全であると考えられる。

ただし、ラウンド関数の構成上、実装攻撃に対する耐性は高くない恐れがあるので、実装攻撃が想定される環境において利用する場合には防御策を注意深く講じることが望まれる。

（理論的解読法ごとの安全性評価）

a) 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全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。

b) 差分解読法

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。

CIPHERUNICORN-A においても、自己評価書では、(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、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ている。このモデルに対し、truncated vector 探索を利用して、15 段での最大差分特性確率の上界が 2^{-120} であると評価している。また、同様のモデルに対し、同じ 15 段での最大差分特性確率の上界が 2^{-140} であるとの見積もりもある。以上より、mF 関数を利用したモデルにおいて最大でも 14 段までの差分パスの存在しか認められないことから、CIPHERUNICORN-A の段数が 16 段であることを考慮すれば、差分解読法の適用がほぼ不可能であり、差分解読

法に対して安全であると考えてよい。また、mF 関数を実際のラウンド関数に置き換え、より詳細な評価を行うことにより、より正確な暗号強度が得られると期待される。

c) 線形解読法

差分解読法と同様に、自己評価書では、mF 関数を利用したモデルとして、truncated vector 探索を利用して、ラウンド関数および 15 段での最大線形特性確率の上界がそれぞれ $2^{-22.47}$、$2^{-157.29}$ であると評価している。また、同様のモデルに対し、ラウンド関数および 14 段での最大線形特性確率の上界がそれぞれ $2^{-20.08}$、$2^{-140.56}$ との見積もりもある。以上より、mF 関数を利用したモデルにおいて最大でも 13 段までの線形パスの存在しか認められないことから、CIPHERUNICORN-A の段数が 16 段であることを考慮すれば、線形解読法の適用がほぼ不可能であり、線形解読法に対して安全であると考えてよい。また、mF 関数を実際のラウンド関数に置き換え、より詳細な評価を行うことにより、より正確な暗号強度が得られると期待される。

d) 高階差分攻撃、補間攻撃、スライド攻撃、mod n 攻撃

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。

e) 鍵関連攻撃

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。

(実装攻撃に対する安全性)

CIPHERUNICORN-A のラウンド関数は、a) データ依存により構成が変わる部分が存在し、b) 内部構成が、本流部と一時鍵生成部という二系統の処理部分を有しており、同一の入力データが分岐処理される。

一般に、a)のようなデータ依存型の処理ではタイミング攻撃が、また b)のような同じデータが複数の処理を行う場合には電力解析攻撃が有効に働く場合が多いとされていることから、タイミング攻撃や電力解析攻撃などの実装攻撃に対する耐性は高くないと恐れがある。したがって、実装攻撃に対する脅威がある環境において利用する場合には、実装攻撃対する防御策を注意深く講じることが望まれる。

2.2 ソフトウェア(SW)実装評価

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	3984 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	“ /O2 /Oy- ”（実行速度）を指定	
1 回目	2 回目	3 回目
1569 / 1574	1570 / 1574	1570 / 1574
1574 / 1578	1574 / 1577	1574 / 1578

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	5644 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	“ -v – fast ”を指定	
1 回目	2 回目	3 回目
2273 / 2282	2273 / 2282	2273 / 2282
2302 / 2326	2309 / 2327	2310 / 2327

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	8472 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	“ -O4 ”を指定	
1 回目	2 回目	3 回目
1834 / 1843	1828 / 1842	1828 / 1842
1769 / 1782	1769 / 1782	1769 / 1782

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C +アセンブラ	
プログラムサイズ	4306 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	“ /O2 /Oy- ”（実行速度）を指定	
1 回目	2 回目	3 回目
4788 / 4822	4788 / 4814	4787 / 4830
4799 / 4931	4798 / 4815	4806 / 4814

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	5644 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	“ -v – fast ”を指定	
1回目	2回目	3回目
7970 / 8160	7961 / 8164	7900 / 8161
8802 / 9025	8817 / 9034	8823 / 9028

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	8552 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	“ -04 ”を指定	
1回目	2回目	3回目
4610 / 4623	4610 / 4628	4610 / 4624
5071 / 5092	5071 / 5100	5071 / 5095

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

暗号化及び復号処理、鍵生成まで含めた暗号化及び復号処理の全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、PentiumIII 上では、全ての測定項目について Triple DES と同程度である。

また、IC カードを代表とする、8 ビット CPU でのソフトウェア実装に関する記述・公開情報は存在していない。

2.3 ハードウェア(HW)実装評価

提案者はハードウェア実装も可能であると述べているが、ハードウェア実装に関する記述・公開情報は存在していないため、今回は実装評価対象としなかった。

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.

5.2.8 Hierocrypt-3

1. 暗号技術

1.1 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、東芝により提案された共通鍵ブロック暗号(1)である。ブロック長は 128bit 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／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、特にスマート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。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

・特願 2000-060482 「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」
(2000/03/06 出願)

・特願 2000-198478 「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録媒体」
(2000/06/30 出願)

・特願 2000-210484 「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」
(2000/07/11 出願)

・特願 2000-211686 「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」
(2000/07/12 出願)

・特願 2000-212175 「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」
(2000/07/13 出願)

提案者は、Hierocrypt-3 を非排他的かつ妥当的な条件で他社に実施許諾する方針である。

応募暗号技術仕様の公開 Web アドレス

http://www.toshiba.co.jp/rdc/security/hierocrypt/

1.2 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／線形解読法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 bitに対し各々6、7、8 段である。

1.3 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。

2. 評価結果

2.1 安全性評価

現時点(2001 年１月)では、暗号が発表されてからの時間が少ないため、安全性評価については、限られた情報しか得られていない。しかし、この中ではどの鍵長の場合についても決定的な欠点となる結果は知られていない。

しかし、いくつかの解析結果が知られており、今後の解析結果に注目する必要がある。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。差分解読法、線形解読法については信頼性の高い評価を行っているが、Hierocrypt-3 の設計者が最も注目する攻撃法のひとつである Square 攻撃については新しい結果が得られた（3.5 段での解読可能性）。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。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。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。

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。しかし、今回採用された鍵スケジュール自体は新しい内容であり、今後の安全性評価に注目する必要がある。

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。

2.2 ソフトウェア(SW)実装評価

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ(MMX 命令)	
プログラムサイズ	68832 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	VC++6.0 Win32 Release(Default)	
1 回目	2 回目	3 回目
404 / 406	404 / 406	404 / 406
426 / 428	426 / 428	426 / 428

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	38936 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 -native -fast -xarch=v8plusa -xCC(暗号化) cc -native -fast -xarch=v9 -xCC(復号)	
1 回目	2 回目	3 回目
511(471) / 554(473)	510(471) / 556(473)	510(471) / 555(473)
759(612) / 826(616)	758(612) / 826(616)	757(612) / 826(616)

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	58152 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 -03	
1 回目	2 回目	3 回目
420(399) / 424(406)	420(399) / 424(406)	420(399) / 423(407)
427(386) / 429(393)	427(386) / 430(394)	427(386) / 430(393)

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ(MMX 命令)	
プログラムサイズ	68832 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	VC++6.0 Win32 Release(Default)	
1 回目	2 回目	3 回目
726 / 728	726 / 729	726 / 728
1345 / 1358	1344 / 1357	1346 / 1358

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	38936 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 -native -fast -xarch=v8plusa -xCC(暗号化) cc -native -fast -xarch=v9 -xCC(復号)	
1回目	2回目	3回目
823(761) / 828(822)	823(761) / 828(821)	824(761) / 828(823)
2673(2612)/2684(2627)	2671(2611) / 2683(2627)	2670(2610)/ 2683(2627)

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	58152 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 -03	
1回目	2回目	3回目
675(668) / 679(672)	675(668) / 678(673)	675(668) / 679(672)
1130(1130) / 1142(1141)	1130(1130) / 1142(1142)	1130(1130) / 1142(1142)

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

備考：Ultra SPARC IIi と Alpha21264 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。

2.3 ハードウェア(HW)実装評価

HW 実装評価は以下の通りである。ループ･アーキテクチャを採用せず、アルゴリズムの全体を速度を重視することを想定した評価例である。したがって、回路規模の削減は可能である。

回路規模(Gate)	データランダム化部	538,078
	鍵スケジュール部	106,302
	制御回路部	—
	Primitive 全体	724,380
クリティカルパス(ns)		75.55
処理速度(Mbps)		1,694.24

設計者らによる評価ではセルライブラリまたは FPGA を使った実装例、4 例が報告されている。

0.14μm CMOS ASIC 897Mbps 81.5 キロゲート

0.14μm CMOS ASIC 84.6Mbps 26.7 キロゲート

ALTERA Max+plusII 51.0Mbps 11 キロゲート(Flex 10K ファミリー)

ALTERA Max+plusII 4.1Mbps 6.3 キロゲート(Flex 10K ファミリー)

学会発表・参考文献等

[1]村谷、大熊、佐野、本山、川村, 64 ビット版 Hierocrypt の提案, 情報処理学会研究報告 CSEC11-9,2000/09

[2] 大熊・村谷・佐野・川村, 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.

[3] K.Ohkuma, H.Muratani, F.Sano, and S.Kawamura, The block cipher Hierocrypt, SAC2000, 2000.

5.2.9 MARS

1. 暗号技術

1.1 技術概要

MARS は、IBM によって開発された 128bit ブロック暗号アルゴリズムで、鍵長として 128～448 ビットをサポートしている。マルチプラットフォームに対応できる設計を行っているが、特に 32 ビットプロセッサ上でのパフォーマンスを重視している。

安全性について、既知の攻撃法に対する耐性のほかに未知の攻撃法に対する耐性を持つよう、複数の構造を組み合わせた。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

MARS に関する特許は米国で申請されているが、公開番号は明らかにされていない。また、同文書の中で、IBM は MARS に関する特許を世界中に無償で Tivoli Systems Inc.から提供すると表明している。

1.2 技術仕様

(1) MARS では、データランダム化部を forward mixing, core mixing, backward mixing に分割し、既知の攻撃法だけではなく、未知の攻撃法に対する耐性を持つように設計した。forward mixing, backward mixing は鍵ビットの強力な攪拌を目的とし、未知の攻撃法に対する強度を期待している。core mixing は既知の攻撃法に対して十分な安全性を持つように設計した。

(2) core mixing は、32 ビット 4 ブロックからなる変形型 Feistel network を採用している。変形型 Feistel network は、さまざまな構造の中でも速度・強度・解析のしやすさのバランスに優れている。

(3) MARS はあらゆる環境での使用に適しているが、特に 32 ビットプロセッサでの使用を重視している。このため、暗号化処理に用いる演算はすべて 32 ビットワードを前提としている。具体的には、排他的論理和、2^{32} を法とした加減乗算、テーブル参照、固定巡回シフト、およびデータ依存型巡回シフトを組み合わせて使用している。

1.3 その他

AES の 5 つの finalist に残った暗号である。なお、CRYPTREC への応募があった後、応募者の日本 IBM から IBM においては MARS を製品化する計画はないという連絡があった。

2. 評価結果

2.1 安全性評価

(1) MARS については、AES 提案よりさまざまな結果が知られているが、どの結果も仕様どおりの MARS の安全性に影響を与える結果ではなく、既存のどの攻撃法も MARS の解読には影響しないと考えられる。
(2) 参考までにこれまで知られる結果を簡単にまとめる。この注意として、ほとんどこれらの安全性評価は MARS をさまざまな方法で簡略化したものであり、単に「段数を x 段に減らしたもの」と表現するのは十分でない。
(3) 不能差分攻撃が適用できる環境は、forward/backward mixing を無視し、さらに、core mixing な 16 段の段関数のうち、8 段のみを考えた場合に、成立することが知られている。
(4) さらに理論的な考察結果として、鍵長が 256 ビットの場合には、forward/backward mixing と 5 段(標準 16 段)の core mixing の組み合わせ、あるいは、forward/backward mixing を無視し、かつ、core mixing 関数 11 段の変形 MARS、などについては、2^{256} を下回る鍵の攻撃手法が考えられることが示されている。
(5)線形解読法については、厳密な評価は得られていないが、概ね安全と思われる。

しかし、以下の点に注意するべきである。

(6) MARS に用いられているデータ依存巡回シフト、2^{32} を法とした乗算は Differential Power Analysis、タイミング攻撃に対する耐性がまだ十分に研究されていない。

2.2 ソフトウェア(SW)実装評価

今回は評価を実施していない。

2.3 ハードウェア(HW)実装評価

HW 実装評価の結果は以下である。0.35μm CMOS ASIC ライブラリでアルゴリズム全体の速度重視の実装を想定している。

回路規模(Gate)	データランダム化部	739,069
	鍵スケジュール部	2,316,846
	制御回路部	—
	Primitive 全体	3,055,914
クリティカルパス(ns)		612.64
Key Setup Time(ns)		1,740.99[1]
処理速度(Mbps)		208.93

参考文献

[1] 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.

5.2.10 RC6

1. 暗号技術

1.1 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 セキュリティ社として応募している可変ブロック長(標準 128 ビット)の共通鍵暗号である。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事を目指している。具体的には、データ依存巡回シフトや、ラウンド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、段関数内に掛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。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

RSA Security Inc.による RC6 に関する知的財産権として次の 4 件が報告されている。

・U.S. 特許

Title: Block Encryption Algorithm with Data-Dependent Rotations

Patent number: 5,724,428, Date of patent March 3, 1998

・Title: Block Encryption Algorithm with Data-Dependent Rotations

Patent number: 5,835,600, Date of patent: November 10, 1998

・U.S. 特許申請

Title: Enhanced Block Encryption Algorithm with Data-Dependent Rotations

Serial number: 09/094,649

・PCT 特許申請

Title: Enhanced Block Encryption Algorithm with Data-Dependent Rotations

Serial number: PCT/US99/13358

応募暗号技術仕様の公開 Web アドレス

http://www.rsasecurity.com/rsalabs/rc6/

1.2 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 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 ビット、鍵長 b=16,24,32 バイトで、段数 r=20 を推奨値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エア実装が可能である。その本体部分は 176 バイトの鍵スケジュール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、何れも、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。

2. 評価結果

2.1 安全性評価

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おらず、十分に使用可能な暗号と評価する。

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秘密鍵の総当たり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、これら経路毎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、差分解読で 12 段まで、線形解読で 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。完全仕様の 20 段 RC6 は安全と考える。

高階差分やその他の攻撃の中で、RC6 に対し、現在、最も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。2^{-60} の割合で存在する弱鍵ならば、16 段まで攻撃可能との報告がある。この範囲の段数では 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。

RC6 は、乗算やデータ依存型巡回シフトを使用しており、一般にはタイミング攻撃や差分電流解析等のサイド・チャンネル攻撃に対する配慮が必要である。その対策は、RC6 に関し、容易であるとの自己評価書の主張とそうでは無いとの意見が有るが、これら攻撃法に対する防御方法の研究は、まだ途上にあり、それらの成果を踏まえ、実装時にはシステム全体としての対策が求められよう。

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。

2.2　ソフトウェア(SW)実装評価

以下の環境でSW実装評価を実施した。評価結果は以下の通りである。

データランダム化部速度測定結果

PentiumIII (650MHz)		
言語	アセンブラ	
プログラムサイズ	1200Byte(暗号化/復号各々)	
コンパイラオプション	/o2 (マイクロソフト C コンパイラ)	
1 回目	2 回目	3 回目
258 / 260	258 / 260	258 / 259
262 / 266	262 / 265	262 / 265

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	3940 Byte (暗号化/復号各々)	
コンパイラオプション	xo5 (WS Compiler C/SPARC、オプティマイズ5)	
1 回目	2 回目	3 回目
2048 / 2088	2047 / 2088	2048 / 2089
2024 / 2076	2023 / 2074	2026 / 2077

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール部+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	1200Byte(暗号化/復号各々)1500Byte(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)	
1 回目	2 回目	3 回目
1631/ 1644	1630 / 1645	1630 / 1642
1633 / 1639	1633 / 1643	1633 / 1640

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	3940Byte(暗号化/復号各々)2196Byte(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)	
1 回目	2 回目	3 回目
4078 / 4111	4078 / 4111	4075 / 4112
4026 / 4054	4024 / 4055	4019 / 4054

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

備考： PentiumIII で測定したコードは Microsoft Windows9X 用の製品プログラム、Ultra SPARC 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。

PentiumIII における暗号化及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、PentiumIII の測定結果で、最も遅い暗号に近い。Ultra SPARC における暗号化及び復号さらに拡大鍵生成まで含めた全ての速度は、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何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。

（スマートカード他の SW 実装評価）

自己評価書には、第 3 者実装を含め、Java、スマートカード、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。

スマート・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スマート・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。

2.3 ハードウェア(HW)実装評価

HW 実装評価の結果は以下である。0.35μm CMOS ASIC ライブラリでアルゴリズム全体の速度重視の実装を想定している。

回路規模(Gate)	データランダム化部	77,785
	鍵スケジュール部	975,391
	制御回路部	—
	Primitive 全体	1,753,076
クリティカルパス(ns)		698.05
Key Setup Time(ns)		2,112.26[1]
処理速度(Mbps)		183.36

この評価結果では、ループ･アーキテクチャを採用したグループでは、データランダム化部の回路規模が最小である。これは、暗号アルゴリズムの簡易性が寄与しているものと考えられる。

応募者自身による評価では、より小規模実装例も報告されている。

FPGA[3]

暗号化	XCV1000	127（Mビット/秒）	フィードバック・モード
暗号化	XCV1000	2.4（Gビット/秒）	非フィードバック・モード

ASIC[4]

暗号化	0.5 um	104（Mビット/秒）	反復方式
暗号化	0.5 um	2.2（Gビット/秒）	パイプ・ライン方式

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] 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)

[4] 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)

5.2.11　SC2000

1. 暗号技術

1.1 技術概説

・本暗号は富士通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 と同じインターフェイスである 128bit データ入出力、128,192,256bit 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 構想と SPN 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 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、スマートカードの認証処理および暗号データの送受信がある。

知的財産権　(応募者資料による)

(提案者の特許とその扱い)

提案者からは次の 4 件の特許が出願されている。

・ 出願番号 00-016413 出願日 2000.1.26 名称「暗号設計装置および記録媒体」

・ 出願番号 00-212813 出願日 2000.7.13 名称「拡散性能に優れた暗号の設計方法ならびに暗号化装置」

・ 出願番号 00-212814 出願日 2000.7.13 名称「Feistel 構造と SPN 構造を組み合わせた演算装置」

・ 出願番号 00-212482 出願日 2000.7.13 名称「拡大鍵生成装置および記録媒体」

上記提案者所有の特許および著作権は、提案者により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される。

（関連する他社の特許）

特になし

応募暗号技術仕様の公開 Web アドレス

http://www.labs.fujitsu.com/theme/crypto/sc2000.html

1.2 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 XOR 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繰り返す。使用する拡大鍵は、128 ビット鍵時は 32 ビット拡大鍵が 56 個、192、256bit 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繰り返す。

（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。

2. 評価結果

2.1 安全性評価

次の 3 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。

(1) データランダム化部の従来型攻撃に対する安全性

差分解読法あるいは線形解読法への耐性を保証するために、特性差分確率や特性線形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特性差分確率、特性線形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]。

差分解読法については、15 段の特性差分確率は、3 段繰り返し型を基本とする場合は 2^{-134} 以下、2 段繰り返し型を基本にする場合は 2^{-150} 以下になることが分かった。すなわち、差分解読法に利用できる特性差分近似式がないことを意味する。4段以上の繰り返しが存在するか否かは検討課題であるが、探索には膨大な計算量が必要なため実施は困難である。3 段繰り返しの結果から類推して、たとえ 4 段繰り返しがあったとしても現実的な計算量で差分解読法は適用困難と考えられる。

線形解読法に対しても truncated vector を利用した解読が可能である。15 段の特性線形近似確率は 3 段繰り返し型を基本とする場合は 2^{-142} 以下、2 段繰り返し型を基本にする場合は 2^{-150} 以下になる

ことが分かった。すなわち、線形解読法に利用できる特性線形近似式がないことを意味する。

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解読法が有効に働く。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／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／補間攻撃の立場では問題がないことが確認された。

truncated 差分解読法に対しては、通常の差分解読法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 乗解読法・分割解読法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。

不能差分解読法、ブーメラン解読法、mod n 解読法、非全射解読法の各解読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。

(2) 鍵スケジュール部の従来型攻撃に対する安全性

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 128bit 以上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、全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に関して問題となる欠点は認められなかった。

(3) 実装に関する攻撃に対する安全性

ハードウェア実装に対するタイミング攻撃は基本的には適用可能である。しかし、SC2000 は小規模のテーブル参照あるいは論理回路で実装されるため鍵データの値に依存した処理時間の差異は考えにくい。そのため対策は不要であるか極めて容易と考えられる。

同様の理由でソフトウェア実装およびＩＣカードの実装に対するタイミング攻撃に対する対処は不要であるか対処可能であると考えられる。

電力解析については、SC2000 は分岐処理を伴わずに実装可能であるのでタイミング解析や単純な電力解析に対しては耐性が高い。通常の単純電力解析や差分電力解析より一歩進んだ、複数データ間の消費電力波形の比較による単純電力解析および差分電力解析の適用可能性を構成要素ごとに検討を行なったが、小さいコストで対処可能であることを確認した。

このように、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、2 段 Feistel 型と SPN 型を交互に重ねた構造に対する暗号解析はほとんど行なわれていない。

ただし、2001年1月のSCIS2001における提案者グループによる発表[3]では、3段繰り返し型の差分/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、全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[YS01]。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。

2.2 ソフトウェア(SW)実装評価

ソフトウェア実装における測定結果は以下の通りである。

データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	21340 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	/G6 /O2 /ML /W3 /GX	
1回目	2回目	3回目
389 / 391	388 / 392	388 / 391
408 / 410	408 / 411	408 / 411

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	25548 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	-xtarget=ultra2 -xO5	
1回目	2回目	3回目
310(275) / 313(277)	310(276) / 313(278)	310(276) / 314(279)
309(283) / 312(286)	309(283) / 312(287)	309(282) / 312(285)

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	39854Byte (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	
1回目	2回目	3回目
289(262) / 297(276)	289(262) / 297(277)	289(262) / 296(276)
282(275) / 296(289)	282(275) / 288(289)	282(275) / 288(289)

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

鍵スケジュール+データランダム化部速度測定結果

Pentium III (650MHz)		
言語	ANSI C + アセンブラ	
プログラムサイズ	23700 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	/G6 /O2 /ML /W3 /GX	
1回目	2回目	3回目
800 / 803	800 / 803	800 / 803
818 / 822	818 / 821	818 / 819

Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	22524 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	-xtarget=ultra2 -xO5	
1回目	2回目	3回目
623 / 627	623 / 627	623 / 627
618 / 622	618 / 622	618 / 622

Alpha21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	39854 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	-fast - arch ev6	
1回目	2回目	3回目
572 / 578	572 / 578	572 / 578
586 / 594	586 / 595	586 / 594

上段：暗号化 下段：復号、(最速値) / (平均値) 単位 cycles/block

備考：Ultra SPARC IIi と Alpha21264 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、PentiumIII と Alpha においては数%大きくなり、UltraSparc 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。

設計者らにより PentiumIII あるいは Athlon を搭載した PC での実装が報告されている。データランダム化

部、鍵スケジュール部ともに相当の最適化がはかられている。

（IC カード実装）
設計者により Intel8051 を搭載した IC カードでの実装が報告されている。暗号化と復号ができるコードが 1751 バイトの ROM で実装可能である。暗号化速度には改良の余地がある。

2.3 ハードウェア(HW)実装評価
今回は評価を実施していない。

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] T. Shimoyama et al., 共通鍵ブロック暗号 SC2000, 信学技報 ISEC2000-72. 2000.
[3] H. Yanami el al., 共通鍵ブロック SC2000 の差分/線形探索、Proceedings of the SCIS 2001, 2001.

5.2.12 Rijndael

1. 暗号技術

1.1 技術概要

Rijndael は、1998 年に J.Daemen（Proton World International）と V.Rijnmen（Katholieke Universiteit Leuven）によって AES(Advanced Encryption Standard)に提案された共通鍵ブロック暗号であり、ブロック長・鍵長ともに 128、192、256bit が利用可能である[1]。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 に選定された[2]。2001 年中に FIPS（Federal Information Processing Standard）となる予定である。

知的財産権

(提案者特許とその扱い)

Rijndael に関する特許に関しては未調査。提案者は、Rijndael に関する特許は米国やその他外国の特許はないと主張している。Rijndael は AES に選定後、2001 年中に FIPS 化が行われ、ロイヤルティ・フリーで利用可能となる予定である。

暗号技術仕様の公開 Web アドレス

http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

1.2 技術仕様

Rijndael 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。Rijndael 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、128bit ブロックの場合は、鍵長が 128、192、256 bit に対応して、10 段、12 段、14 段のとなる。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。

1.3 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。

2. 評価結果

2.1 安全性評価

128 ビットブロック暗号の Rijndael 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 bit 鍵の仕様通りの Rijndael を解読可能な攻撃法は発見されていない。
・128 bit 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 bit 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 bit 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）と報告している[2]。Rijndael は AES 暗号であり、現時点ではその安全性において十分に信頼がおけると考えられるが、電子政府向け暗号としては、2001 年中に制定される予定である FIPS 版を再評価して使用することを推奨する。

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 の差分解読法、線形解読法、Truncated 差分解読法、SQUARE 攻撃、補間攻撃、弱鍵、鍵関連攻撃について考察し、全てのブロック長と鍵長の組合せにおいて、鍵の全数探索法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、差分解読法と線形解読法に対しては、差分特性確率及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法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法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、Rindael 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 bit 鍵および 256 bit 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 bit 鍵および 256 bit 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 bit 鍵の場合は７段まで、256 bit 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 bit 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、これまで公開の場で Rijndael に関する安全性評価が進められてきたが、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]。

2.2 ソフトウェア(SW)実装評価

Rijndael の SW 実装評価としては、幾つかの評価環境（CPU、言語、他）のもとで実装結果が報告されている[2]。以下に評価結果例として 32 ビット CPU で PentiumIII 上でのＣ言語による実装評価[8]を示す。

＜評価環境＞　○評価対象（CPU）: Pentium III　600[MHz]
○プログラム言語: Visual C++　Ver.6.0
○その他:　128MB RAM、Windows98 4.10.1998

＜評価結果＞　●暗号化鍵セットアップタイム:
128 bit 鍵　1289［cycles］
192 bit 鍵　2000［cycles］
256 bit 鍵　2591［cycles］
●復号鍵セットアップタイム:
128 bit 鍵　1724［cycles］
192 bit 鍵　2553［cycles］
256 bit 鍵　3255［cycles］
●暗号化（ECB）速度:
128 bit 鍵　805［cycles］
192 bit 鍵　981［cycles］
256 bit 鍵　1155［cycles］
●復号　（ECB）速度:
128 bit 鍵　784［cycles］
192 bit 鍵　955［cycles］
256 bit 鍵　1121［cycles］

また、その他報告されている主な評価結果[2]について以下の表に示す。

表１　32 ビットプロセッサ(暗号化)

	A(C 言語)	B(C 言語)	C(C 言語)	D(C 言語)	E(Java)
	cycles	cycles	cycles	cycles	cycles
128 bit 鍵	237	1276	805	362	7770
192 bit 鍵			981	428	
256 bit 鍵			1155	503	

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 PentiumIII 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2　64ビットプロセッサ（暗号化：C言語＋アセンブリ言語）

	F	G	H	I
	cycles	cycles	cycles	cycles
128 bit鍵	168	125	490	293

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　8ビットプロセッサ（暗号化：C言語＋アセンブリ言語）

	J	K
	cycles	cycles
128 bit鍵	9464	25494

J: Motorola 6805 CPU Core, C.Ref.[17], Table 3.

K: Z80 CPU+coprocessor.Ref.[18],Table 8.

表4　32ビットプロセッサ(復号：C言語)

	B	C	D
	cycles	cycles	cycles
128 bit鍵	1276	784	358
192 bit鍵		955	421
256 bit鍵		1121	492

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.

表5　64ビットプロセッサ（復号：C言語＋アセンブリ言語）

	F	G
	cycles	cycles
128 bit鍵	168	126

F: Hewlett-Packard PA-RISC, ASM. Source: Ref.[14], Appendix A.

G: Hewlett-Packard IA-64, C. Source:Ref.[14], Appendix A., Ref.[15]

表6　32ビットプロセッサ(鍵セットアップ:C言語)

	B	C	D
	cycles	cycles	cycles
128 bit鍵	17742(18886)	1289(1724)	215(1334)
192 bit鍵		2000(2553)	215(1591)
256 bit鍵		2591(3255)	288(1913)

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.

表7　64ビットプロセッサ（鍵セットアップ：C言語＋アセンブリ言語）

	F	G
	cycles	cycles
128 bit鍵	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]

表8　8ビットプロセッサ（鍵セットアップ：C言語＋アセンブリ言語）

	K
	cycles
128 bit鍵	10318

K: Z80 CPU+coprocessor.Ref.[18],Table 8.

2.3 ハードウェア(HW)実装評価

今回は評価を実施していない。

RijndaelのHW実装評価としては、市川らによってASICによる高速実装結果が報告されている[9]。

＜評価環境＞　○評価対象：　ASIC（三菱電機製0.35μルールASICライブラリ）

○記述言語：　Verilog－HDL

○評価条件：　Worstケース

＜評価結果＞　●ゲートサイズ（NANDゲート換算）：

トータル：612,843 (Gate)

（暗号化＆復号部：518,508、鍵スケジュール部：93,708）

●鍵セットアップ：57.39(ns)

●スループット：1950.03(Mbps)

それ以外にもFPGAでの実装例が多数報告されている[2]。

参考文献

[1]J.Daemen and V.Rijmen,AES proposal: Rijndael, AES algorithm submission, September 3, 1999, http://nist.gov/aes (AES home page).

[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/

[3]V.Rijmen, et al., The Cipher SHARK, 3rd Fast Software Encryption, 1996, 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, March22-23, 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.82-9

5.2.13 MULTI-S01

1. 暗号技術

1.1 技術概説

MULTI-S01 は、2000 年に、ISEC 研究会において、古屋、渡辺、宝木により提案された暗号技術である。MULTI-S01 は、暗号化処理、復号処理からなり、それぞれ、擬似乱数生成器とデータランダム化部分の 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 bit)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×64 bit)、冗長符号 R (64 bit)、秘密鍵 A (A≠0、64 bit)、秘密鍵 Bi((n+2) × 64 bit)、秘密鍵 S (64 bit)を入力として、暗号文 C ((n+2)×64 bit)を出力する。復号化は、暗号文 C (64×n' bit)、冗長符号 R (64 bit)、秘密鍵 A (≠0,64 bit)、秘密鍵 B (64×n' bit)、秘密鍵 S (64 bit) を入力して、改ざん検出信号、またはメッセージ M (64×(n'-2) bit) 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。

知的財産権 (応募者資料による)

(提案者特許とその扱い)

応募者は、MULTI-S01 の技術に関し、下記の出願特許および登録特許が関連するとしている。また、下記特許を非差別的、かつ適正な対価条件でライセンスするとしている。

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ない。

(1) 特願 2000-108334 号　「暗号化装置および方法」

(2) 特願 2000-070994 号　「共通鍵暗号方法及び装置」

(3) 特願 2000-210690 号　「共通鍵暗号方法及び装置」

応募暗号技術仕様の公開 Web アドレス

http://www.sdl.hitachi.co.jp/crypto/

1.2 技術仕様

暗号化処理では、メッセージ M、冗長性 R (64 bit)、秘密鍵 K (256 bit)をそれぞれバイト列によるデータ(M $(8)_i$ (i =1,…,⌈m/ 8⌉), R $(8)_i$ (i =1,…, 8), K $(8)_i$ (i =1,…, 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×(⌈m/ 64⌉+ 2) bit で、バイト列として出力する。これに対応する復号化処理では、暗号文 C (c bit)、冗長性 R (64 bit)、秘密鍵 K (256 bit)をそれぞれバイト列によるデータ(C $(8)_i$ (i =1,…,⌈c/ 8⌉), R $(8)_i$ (i =1,…, 8), K $(8)_i$(i

=1,…, 32))として入力する。復号化処理の出力は復号化結果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・復号化処理の内部は64 bit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=⌈m/ 64⌉+ 2とする。擬似乱数生成器は、Kを入力として、A (64 bit)とB (64×(n+2) bit)、S (64 bit)を出力する。よって、暗号化処理のデータランダム化部分は、M,R,A,B,Sを入力としてCを出力し、復号化処理のデータランダム化部分は、C,R,A,B,Sを入力とし、復号化結果M'または改ざん検出信号を出力する。鍵、平文、暗号文、冗長データ、初期値はバイト単位の列として扱う。これらは64 bitのデータ型との変換の際、Big-Endianにより変換される。

1.3 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。

2. 評価結果

2.1 安全性評価

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、擬似乱数生成器と攪拌関数で構成されている。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、線形解読法、 差分解読法が行なわれたが、大きな危険性は報告されていない。ただし、差分解読法では、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。

2.2 ソフトウェア(SW)実装評価

以下の環境でSW実装評価を実施した。評価結果は以下の通りである。

ストリーム暗号速度評価結果

Pentium III (650MHz)		
言語	ANSI C	
プログラムサイズ	12868Kbyte	
コンパイラオプション	/nologo /G6 /ML /W3 /GX /O2 /Ob2 /D "WIN32" /D "NDEBUG" /D "_CONSOLE" /D "_MBCS" /Fas /Fa"Release/" /Fp"Release/VLIW64_NEW.pch" /YX /Fo"Release/" /Fd"Release/" /FD /GM /c	
1 回目	2 回目	3 回目
176 / 180	175 / 177	176 / 177

(最速値) / (平均値) 単位 cycles/64bits

備考：ストリーム暗号は本来、HW 実装を主眼としている。PANAMA に関しては応募者独自の実装ではなく、入手可能な最速のプログラムが使用されている。これは、特別な最適化は施されていない C 言語プログラムである。

- 自己評価書では、次の結果となっている。

言語　C：
コンパイラオプション（DEC cc)：最適化オプション　-tune ev56 -arch ev56 -06
CPU：Alpha 21164A 600 MHz　RAM：512 Mbyte
OS：DIGITAL UNIX 4.0E

	メモリ使用量：	速度 (Mbps)	(clock/byte)
初期化：	2.4Kbyte		
暗号化：	3.6Kbyte	270.7Mbps	17.7
復号化：	3.7Kbyte	267.3Mbps	18.0

2.3 ハードウェア(HW)実装評価

アルテラ社の FPGA(Field Programmable Gate Array)上で、C 言語で作成されたプログラムに対して、Verilog HDL により回路記述し、シミュレーションを行った。使用した開発環境は、

・ModelSim VHDL/Verilog Version 5.4e（Model Technology）
・Synplify（Synplicity Inc.）

である。評価結果は以下の通りである。

動作周波数(MHz)	処理速度(Gbps)	リソース使用量	使用 FPGA
18.8	1.203	19,811/42,240 ATOMs (46%)	EP20K1000E

自己評価書では、次の結果となっている。

0.35μm の CMOS 技術

(1)処理速度優先

140k ゲート、動作周波数 140MHz、9.1Gbps の処理スループット

(2)論理規模優先

68k ゲート、動作周波数 620(200)MHz、620(200)Mbps の処理スループット

2.4 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。

参考文献

電子情報通信学会技術研究報告書 ISEC2000-68 (2000 年 9 月)

5.2.14　TOYOCRYPT-HS1

1　暗号技術

1.1 技術概説

TOYOCRYPT-HS1 は 2000 年に開催された電子情報通信学会情報セキュリティ研究会（ISEC）において杉本浩一により提案された暗号技術であり、擬似乱数生成アルゴリズム（HR1）によって生成した擬似乱数と平文をビット毎に排他的論理和演算することにより、暗号化を実現する。生成される擬似乱数列は、128 ビットの固定鍵と 128 ビットのストリーム鍵によって決定される特徴をもつ。

知的財産権　(応募者資料による)

(提案者特許とその扱い)

特願平 10-039677　(公開番号　平 11-224183) 擬似乱数発生装置

特願平 10-129606　(公開番号　平 11-305661) シフトレジスタ構成用回路およびシフトレジスタ

特願平 10-267415　(公開番号　2000-081969) 擬似乱数発生装置

特願平 11-007826　(公開番号　2000-209165) 暗号通信システム

応募者は、ライセンス方針については保留としている。

応募暗号技術仕様の公開 Web アドレス

http://www.toyocom.co.jp

概要

TOYOCRYPT-HS1 は、擬似乱数生成アルゴリズム(HR1)によって生成した擬似乱数と平文をビット毎に排他的論理和演算することにより、暗号化を実現する。生成される擬似乱数列は、128 bit の固定鍵と 128 ビットのストリーム鍵によって決定される特徴を持つ。

技術のポイント

TOYOCRYPT-HS1 は小規模、省電力性を要求するハードウェア上超高速な暗号化が必要なアプリケーションにおいて最も効果を発揮するようにハードウェアパフォーマンスを重視した設計をとっている。また、TOYOCRYPT-HS1 はソフトウェアでは、コンパクトな実装が可能となるようにシンプルな構造をとっている。TOYOCRYPT-HS1 はガロア体理論を基に設計しており、基本演算は GF(2)上で定義される。

安全性の議論には、線形複雑度、非線形度等の理論、非線形コンバイナ固有の攻撃に関する理論を利用している。ソフトウェア実装に関しては、Bitslice 実装技術を採用している。

1.2 技術仕様

TOYOCRYPT-HS1 は擬似乱数生成アルゴリズムによって生成した擬似乱数と平文をビット毎に排他的論理和演算することにより、暗号化を実現する。

生成される擬似乱数列は、128 ビットの固定鍵と 128 ビットのストリーム鍵によって決定される。

TOYOCRYPT-HS1 における擬似乱数生成アルゴリズムは以下の（設計者による）設計基準に基づいて設計されている。

(1) 0/1 の等頻度性を理論的に保証する。

(2) 長周期性を理論的に保証する（周期の下限を保証）。

(3) 線形複雑度の下限を理論的に保証し、滑らかに上昇するようにする。

(4) 非線形コンバイナ特有の攻撃に対し十分耐性のある構成にする。

(5) ハードウェアで構成した場合、高速動作し、十分小さな規模となるようにする。

上記基準の内、(1)～(4)は擬似乱数生成アルゴリズムの安全性の要件即ち、生成された擬似乱数の部分系列から、他の部分ビットを推定困難とするための条件である。特に、(4)については、代表的な攻撃である、Fast Correlation Attack, Inversion Attack, Conditional Correlation Attack を考慮している。

TOYOCRYPT-HS1 は同期型鍵ストリーム暗号に分類され、鍵ストリーム生成アルゴリズムの生成する擬似乱数出力と平文を１ビットずつ排他的論理和演算することにより暗号化を実現している。TOYOCRYPT-HS1 の鍵ストリーム生成アルゴリズムは非線形コンバイナに分類され、線形フィードバックシフトレジスタの出力を非線形変換関数を介して出力することにより、１ビットずつ擬似乱数を生成する。

線形フィードバックシフトレジスタの構成には、ガロア・コンフィギュレーションが採用されている。これはガロア・コンフィギュレーションがハードウェアで高速動作することに注目したものである。非線形変換関数は、小規模なハードウェアとなるように、算術演算を用いず、論理演算を基本演算として構成されている。

2. 評価結果

2.1 安全性評価

近い将来の計算機環境で解読可能となり得るため、現状のままでは電子政府用の暗号として推薦できない。提案アルゴリズムに、何がしかの改善を行ってから、実システムには採用すべきであると考える。実質的な鍵長は、固定鍵が既知ならば、128 ビットから高々96 ビットに減少する。これは、現在の計算機処理能力では安全な範疇であるが、将来を見越した安全性は運用者が判断する必要がある。運用上の課題として、LFSR の Seed を如何にして与えるか、少なくともその指針を明確にしておくべきである。Seed がもれることが、この乱数発生器の安全性に影響する。

さらに、しらみつぶし攻撃よりも効果的な分割統治攻撃が可能である。この攻撃の計算量評価は 2^{96} 程度である。

[1]では TOYOCRYPT-HS1 に対する解析論文が発表されている。ここでは、Biryukov-Shamir [Proc. ASIACRYPT 2000]による最新の解析手法であるタイム・メモリ・トレードオフ法を利用している。解析結果は、2^{32} ビットの出力データから、メモリ 2^{32} , 計算量 2^{64} 程度で 128bit 鍵が求まるというものである。

2.2 ソフトウェア(SW)実装評価

以下の環境で SW 実装評価を実施した。評価結果は以下の通りである。

ストリーム暗号速度評価結果

Pentium III (650MHz)		
言語	ANSI C	
プログラムサイズ	5292byte	
コンパイラオプション	/nologo /G6 /ML /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_CONSOLE" /D "_MBCS" /Fp"Release/toyo-hs1.pch" /YX /Fo"Release/" /Fd"Release/" /FD /c	
1 回目	2 回目	3 回目
13867 / 13880	13869 / 13895	13865 / 13892

(最速値) / (平均値) 単位 cycles/64bits

備考：ストリーム暗号は本来、HW 実装を主眼としている。また、測定プログラムのインターフェースに合わせる都合上、シリアル実装を行っている。パラレル実装を施すと、約 8 倍の高速動作が見込まれる。

自己評価書には、以下のような実装データが記載されている。

実装言語および評価条件

プログラム言語　ANSI C

対象プロセッサ　SH1

評価環境　日立製 SH1 コンパイラシミュレータ Windows95 Ver. 5.0

コンパイル条件　速度優先、ループ展開インライン展開なし

結果

シリアル実装

実行速度	66991 cycle/word, 9.56 Kbps (at 20MHz)
鍵セットアップ	591112 cycle, 30.0 ms (at 20MHz)
プログラムコード	910 Byte
定数（鍵を含む）	368 Byte
スタック（RAM）	1184 Byte

パラレル実装 (Bitslice 実装)

実行速度	3678 cycle/word, 174 Kbps (at 20MHz)
鍵セットアップ	673332 cycle, 33.7 ms (at 20MHz)
プログラムコード	800 Byte
定数（鍵を含む）	1360 Byte
スタック（RAM）	2052 Byte

しかし、ソフトウエア実装は、相対的に遅い。設計者は高速化の手法として、並列処理を提案しているが、鍵長が増える問題点がある。

また、新たな原始多項式を生成することに鍵を利用することは（アプリケーションによっては）効率が悪い場合もある。

2.3 ハードウェア(HW)実装評価

アルテラ社の FPGA(Field Programmable Gate Array)上で、C 言語で作成されたプログラムに対して、Verilog HDL により回路記述し、シミュレーションを行った。使用した開発環境は、

・ModelSim VHDL/Verilog Version 5.4e（Model Technology）

・Synplify（Synplicity Inc.）

である。評価結果は以下の通りである。

評価対象		動作周波数 (MHz)	処理速度 (Gbps)	リソース使用量	使用 FPGA
TOYOCRYPT-HS1	回路規模	58.1	0.052	11,883/24,320 LCs	EP20K600E
	処理速度	45.2	1.446	16,144/24,320 LCs	EP20K600E

提案者による FPGA での評価は以下の通りである。

VHDL によるシミュレーション

鍵セットアップ部分に必要な付加回路としては、9 ビットカウンタ１つ程度と非常に小規模であり、セットアップ時間は 256 clock と非常に短時間である。

FPGA によるシミュレーション結果

HDL エディタ	: HDL Turbo Writer Ver5.1C Build3 (Excel Consuitants Ltd)
論理合成ツール	: Synplify-Lite 3.0 (Synplicity Inc)
配置配線ツール	: SpDE 8.1
シミュレーションツール	: SILOS Ⅲ ver 99.070 (SIMCAD.Inc)
ターゲットデバイス	: QL3012-4
パッケージタイプ	: 84 pin PLCC
デバイスプロセス	: C-MOS 0.35μm 4層

コンパイルオプション	項目	小項目	性能
速度優先実行速度	Worst case		226MHz(Mbps)
	Normal case		339MHz(Mbps)
	Best case		418MHz(Mbps)
回路規模		Cell	235/320 (73.4%)
		Routing resources	3329/42812 (7.8%)
回路規模優先実行速度	Worst case		75MHz(Mbps)

		Normal case			112MHz(Mbps)
		Best case			138MHz(Mbps)
回路規模				Cell	195/320 (60.9%)
				Routing resources	2953/42812 (6.9%)
※	Worst case：	70℃	3.00V		
	Normal case：	25℃	3.30V		
	Best case：	0℃	3.60V		

ゲートアレイによる性能見積もり（提案者による）

ゲートアレイ（東芝製 TC200G52, 3.3V, 3層 0.4μm プロセス）で構成した場合、約 3.3K ゲートで 223Mbps(Worst case, 70℃, 3.0V)を達成できるとの見積もり報告がある。

	条件	遅延[ns]	速度[MHz]
	Worst case	4.5	223
	Normal case	2.5	403
	Best case1.3	790	
※	Worst case：	70℃	3.00V
	Normal case：	25℃	3.30V
	Best case：	0℃	3.60V

参考文献・学会発表

[1] 杉本浩一、力石徹也、森住徹也, ストリーム暗号の設計法と評価, 信学技報 ISEC2000-69

[2] M. Mihaljevic and H.Imai, Effective secret key size of TOYOCRYPT-HS1 stream cipher, Proc. SCIS2001

[3]杉本、力石、森住、佐藤、黒沢, 安全な Filter Generator の一構成法, 1998 年暗号と情報セキュリティシンポジウム、SCIS98-5.1.C, 1998 年 1 月

[4]杉本、力石、森住、佐藤、黒沢, 安全な Filter Generator の一構成法, 1998 年信学総大, A-7-12, 1998 年 3 月

[5]杉本、佐藤、黒沢, Modular LFSR 型 Filter Generator に対する Fast Correlation Attack, 信学技報（ISEC), ISEC98-2（PP.11-20), 1998 年 5 月.

[6]杉本、黒沢, ソフトウェアによるストリーム暗号の高速実装, 1998 信学ソ大, A-7-3, 1998 年 9 月.

[7]杉本、黒沢, ソフトウェアによるストリーム暗号の高速実装, 1999 年暗号と情報セキュリティシンポジウム（SCIS99), F1-2.1（pp.795-799), 1999 年 1 月.

6. ハッシュ関数の評価

対象のハッシュ関数は下表である。6.1 節では、ハッシュ関数カテゴリについて、その特徴、安全性、実装性の観点から相互比較し、短評を述べる。6.2 節では、個別ハッシュ関数に関し、それらの観点のより詳細な記述を行う。なお、ハッシュ関数は、ハッシュ関数名のアルファベット順に配置してある。

ハッシュ関数名

ハッシュ関数	MD5、RIPEMD-160、SHA-1

6.1 種別による評価

表 6.1.1 ハッシュ関数の一覧表

<table>
<tr><td></td><td>MD5</td><td>RIPEMD-160</td><td>SHA-1</td></tr>
<tr><td rowspan="8">特 徴</td><td colspan="3">メッセージダイジェスト長</td></tr>
<tr><td>128 bit</td><td>160 bit</td><td>160 bit</td></tr>
<tr><td colspan="3">基本処理単位のビット長</td></tr>
<tr><td>512 bit</td><td>512 bit</td><td>512 bit</td></tr>
<tr><td colspan="3">総処理ステップ数</td></tr>
<tr><td>64（4 ラウンド×16 ステップ）</td><td>160（5 ラウンド×16 ステップ×2 ライン）</td><td>80（4 ラウンド×20 ステップ）</td></tr>
<tr><td colspan="3">最大入力可能メッセージ長</td></tr>
<tr><td>∞</td><td>$2^{64}-1$ bit</td><td>$2^{64}-1$ bit</td></tr>
<tr><td>安全性</td><td colspan="3">MD5 に対する攻撃はいくつか報告されているが、いずれも実用的な攻撃とは言い難い。MD5、RIPEMD-160、SHA-1 に対する実用的な攻撃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、もちろん、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。例えばハッシュ値の長さが n bit である場合、バースデー攻撃により $2^{n/2}$ 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があるため、ハッシュ値を十分長くする必要がある。MD5 は 128 bit のハッシュ値であり、Birthday 攻撃に対して十分な耐性を有さないという意見もある。最近の研究では少なくとも n = 160 bit 以上必要であると考えられている。
以下にハッシュ関数で考えられる攻撃パターン、すなわちハッシュ関数にとって不利となりうる性質の発見に対する耐性を与える。“－”は現在、その有効な攻撃方法が発見されていないことを示し、“△”はその攻撃が成功する場合があることを示している。
<table>
<tr><td>攻撃の種類</td><td>MD5</td><td>RIPEMD-160</td><td>SHA-1</td></tr>
<tr><td>Preimage</td><td>－</td><td>－</td><td>－</td></tr>
<tr><td>Pseudo-preimage</td><td>－</td><td>－</td><td>－</td></tr>
<tr><td>2nd preimage</td><td>－</td><td>－</td><td>－</td></tr>
<tr><td>Collision</td><td>－</td><td>－</td><td>－</td></tr>
<tr><td>Pseudo-collision</td><td>△</td><td>－</td><td>－</td></tr>
</table>
</td></tr>
</table>

6.2 個別評価

6.2.1 MD5

1. 暗号技術

1.1 技術概説

MD5 は R. Rivest により 1991 年に提案されたハッシュ関数である。これはビット長が 512 ビットの倍数になるようにパディングされたメッセージを入力として、128 bit のハッシュ値を出力する関数である。MD5 は 32 bit 計算機に適合するよう設計されたハッシュ関数であり、ソフトウェアによる実装でも高い処理速度を達成できる。

1.2 技術仕様

32 bit 計算機において高速処理が可能となるように、32 bit の算術加算、論理演算、巡回シフト演算などを主要演算として用いて構成されている。MD5 は入力・圧縮・出力の 3 つの部分で構成される。MD5 は 512 bit メッセージブロックと 4 つの 32 bit 変数を入力とし、4 つの新しい変数値を出力とする関数の繰り返しにより 128 bit のハッシュ値を出力する。圧縮関数は 4 ラウンド 64 ステップで構成される。

(1)入力

入力はリトルエンディアン方式により 32 bit 整数に変換され、512 bit ブロックに分けられる。512 bit のメッセージブロックを X=（X[0],X[1],...,X[15])と表現すると各 X[i]（0≦i≦15）は 32 bit であり、次に説明する圧縮関数のステップ関数への入力として使われる。

(2)圧縮関数

圧縮関数の計算には 4 つの連鎖変数（A,B,C,D）を用いる。（A,B,C,D）の初期値 IV=(h_1,h_2,h_3,h_4)としては

$$h_1 = 0x67452301, \quad h_2 = 0xefcdab89, \quad h_3 = 0x98badcfe, \quad h_4 = 0x10325476$$

を用いる。また MD5 で使われるブール関数は次のように定義されている。

$$f(x,y,z) = (x \wedge y) \vee (\overline{x} \wedge z)$$
$$g(x,y,z) = (x \wedge z) \vee (y \wedge \overline{z})$$
$$h(x,y,z) = x \oplus y \oplus z$$
$$i(x,y,z) = y \oplus (x \vee \overline{z})$$

ここで、記号∧，∨，⊕はそれぞれビット毎の論理積、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。このブール関数により、MD5 の圧縮関数のステップ関数は次のようにあらわされる。

１ラウンド　$FF(A,B,C,D,X[i],s,K[j]) : A = B + (A + f(B,C,D) + X[i] + K[j])^{<<s}$

２ラウンド　$GG(A,B,C,D,X[i],s,K[j]) : A = B + (A + g(B,C,D) + X[i] + K[j])^{<<s}$

３ラウンド　$HH(A,B,C,D,X[i],s,K[j]) : A = B + (A + h(B,C,D) + X[i] + K[j])^{<<s}$

４ラウンド　$II(A,B,C,D,X[i],s,K[j]) : A = B + (A + i(B,C,D) + X[i] + K[j])^{<<s}$

ただし、$X^{<<s}$ は変数 X をｓビット左巡回シフトする演算を表す。ここで、$K[j]\,(1 \le j \le 64)$ は

$2^{32} \cdot abs(\sin(j))$ の整数部分とする。ただし sin 関数の引数 j はラジアンを単位とする。X[0]～X[15]はあらかじめ定められた順序でステップ関数に入力される。また、左巡回シフト量 s も各ステップ毎にあらかじめ定義された値を用いる。

(3)出力

MD5 は圧縮関数の最後のステップで求められた連鎖変数の値と最初の初期値 IV を加えた後、(A,B,C,D)の４つの変数を結合することで 128 bit のハッシュ値を出力する。MD5 の詳細仕様については[1]を参照。

1.3　その他

MD5 はこれより先に同じく R.Rivest によって発表された MD4 の改良版として提案された。

2　評価結果

2.1　安全性評価

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) Pseudo-collision の発見の手間、である。MD5 について(1)への耐性は充分であると考えられるが、(2)についてはハッシュ値のビット数が 128 bit と比較的短いことから衝突の発見が脅威となる用途への利用には注意が必要である。以下、補足説明する。

・MD5 固有の攻撃

MD5 はハッシュ関数のアルゴリズム中で用いられる初期値に対する衝突の探索が最も効率の良い攻撃と考えられている。この種の攻撃としては、これまでに Boer と Bosselaers による攻撃（Pseudo-collision を 2^{16} 回の操作で発見）[2]や、Dobbertin の攻撃[3]が発表されている。しかしいずれの攻撃法も実用上問題となる攻撃とはいえない。

・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ハッシュ値のパターンは 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ような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。MD5 では n=128 であり、この方法を適用した場合攻撃には 2^{128} 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量であると考えられている。

・衝突発見の手間

一方、Birthday パラドックスによれば、$2^{n/2}$ 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入力値ペアを発見することができる。MD5 は n=128 であるため、2^{64} 個程度の入力値を用意することができれば Birthday 攻撃が適用できることになる。

2.2　ソフトウェア(SW)実装評価

CRYPTREC 独自の実装評価はしていないが、報告書作成時において[4]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 850 [MHz]

OS および使用言語: Window 2000 SP1, C++

処理性能　　: 100.738 [Mbyte per sec]

2.3　ハードウェア(HW)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。

[1] R.Rivest, The MD5 message-digest algorithm, Request For Comments (RFC) 1321, Internet Activities Board, Internet Privacy Task Force, April 1992 (http://www.roxen.com/rfc/rfc1321.html)

[2] B. den Boer and A.Bosselaers, Collisions for the compression function of MD5, Advances in Cryptology -- Eurocrypt'93, pp.293-304, 1993.

[3] H.Dobbertin, The status of MD5 after recent attack, CryptBytes, 2 (2), Sep., pp.1-6, 1996.

[4] http://www.eskimo.com/~weidai/benchmarks.html

6.2.2 RIPEMD-160

1. 暗号技術

1.1 技術概説

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 などと共に ISO の国際規格にも採用されている[1]。RIPEMD-160 はビット長が 512 bit 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 bit のハッシュ値を出力する。

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

入力メッセージはリトルエンディアン方式により 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]～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。

$$h_1 = 0x67452301, h_2 = 0xefcdab89, h_3 = 0x98badcfe, h_4 = 0x10325476, h_5 = 0xc3d2e1f0$$

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。

$$f(x,y,z) = x \oplus y \oplus z$$
$$g(x,y,z) = (x \wedge y) \vee (\bar{x} \wedge z)$$
$$h(x,y,z) = (x \wedge \bar{y}) \oplus z$$
$$k(x,y,z) = (x \wedge z) \vee (y \wedge \bar{z})$$
$$l(x,y,z) = x \oplus (y \vee \bar{z})$$

ここで、記号∧, ∨, ⊕はそれぞれビット毎の論理積、論理和、排他的論理和を表し、$\bar{x}$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通りである。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。

$$K_L[j]=0 \qquad K_R[j]=0x50a28be6 \qquad (1\le j\le 16)$$
$$K_L[j]=0x5a27999 \qquad K_R[j]=0x5c4dd124 \qquad (17\le j\le 32)$$
$$K_L[j]=0x6ed9eba1 \qquad K_R[j]=0x6d703ef3 \qquad (33\le j\le 48)$$
$$K_L[j]=0x8f1bbcdc \qquad K_R[j]=0x7a6d76e9 \qquad (49\le j\le 64)$$
$$K_L[j]=0xa953fd4e \qquad K_R[j]=0 \qquad (65\le j\le 80)$$

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通りである。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 $(1\le j\le 16)$

$$FF_L(A_L,B_L,C_L,D_L,E_L,X[i],s_L[j],K_L[j]):$$
$$A_L=(A_L+f(B_L,C_L,D_L)+X[i]+K_L[j])^{<<s_L[j]}+E_L,\quad C_L=C_L^{<<10}$$

$$LL_R(A_R,B_R,C_R,D_R,E_R,X[i],s_R[j],K_R[j]):$$
$$A_R=(A_R+l(B_R,C_R,D_R)+X[i]+K_R[j])^{<<s_R[j]}+E_R,\quad C_R=C_R^{<<10}$$

2 ラウンド $(17\le j\le 32)$

$$GG_L(A_L,B_L,C_L,D_L,E_L,X[i],s_L[j],K_L[j]):$$
$$A_L=(A_L+g(B_L,C_L,D_L)+X[i]+K_L[j])^{<<s_L[j]}+E_L,\quad C_L=C_L^{<<10}$$

$$KK_R(A_R,B_R,C_R,D_R,E_R,X[i],s_R[j],K_R[j]):$$
$$A_R=(A_R+k(B_R,C_R,D_R)+X[i]+K_R[j])^{<<s_R[j]}+E_R,\quad C_R=C_R^{<<10}$$

3 ラウンド $(33\le j\le 48)$

$$HH_L(A_L,B_L,C_L,D_L,E_L,X[i],s_L[j],K_L[j]):$$
$$A_L=(A_L+h(B_L,C_L,D_L)+X[i]+K_L[j])^{<<s_L[j]}+E_L,\quad C_L=C_L^{<<10}$$

$$HH_R(A_R,B_R,C_R,D_R,E_R,X[i],s_R[j],K_R[j]):$$
$$A_R=(A_R+h(B_R,C_R,D_R)+X[i]+K_R[j])^{<<s_R[j]}+E_R,\quad C_R=C_R^{<<10}$$

4 ラウンド $(49\le j\le 64)$

$$KK_L(A_L,B_L,C_L,D_L,E_L,X[i],s_L[j],K_L[j]):$$
$$A_L=(A_L+k(B_L,C_L,D_L)+X[i]+K_L[j])^{<<s_L[j]}+E_L,\quad C_L=C_L^{<<10}$$

$$GG_R(A_R,B_R,C_R,D_R,E_R,X[i],s_R[j],K_R[j]):$$
$$A_R=(A_R+g(B_R,C_R,D_R)+X[i]+K_R[j])^{<<s_R[j]}+E_R,\quad C_R=C_R{}^{<<10}$$

5 ラウンド $(65\le j\le 80)$

$$LL_L(A_L,B_L,C_L,D_L,E_L,X[i],s_L[j],K_L[j]):$$
$$A_L=(A_L+l(B_L,C_L,D_L)+X[i]+K_L[j])^{<<s_L[j]}+E_L,\quad C_L=C_L{}^{<<10}$$

$$FF_R(A_R,B_R,C_R,D_R,E_R,X[i],s_R[j],K_R[j]):$$
$$A_R=(A_R+f(B_R,C_R,D_R)+X[i]+K_R[j])^{<<s_R[j]}+E_R,\quad C_R=C_R{}^{<<10}$$

(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 とのラインを使うため以下のように計算する。

$$A=h_2+C_L+D_R,\quad B=h_3+D_L+E_R,\quad C=h_4+E_L+A_R$$
$$D=h_5+A_L+B_R,\quad E=h_1+B_L+C_R$$

1.3　その他

RIPEMD-160 はこれより先の 1995 年に RIPE プロジェクトに提案された RIPEMD に対する攻撃法が見出されたため、その改良方式として提案された[2]。また RIPEMD-160 は MD4 をベースとするハッシュ関数のひとつでもある。

2　評価結果

2.1　安全性評価

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することが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、右左 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

MD5 の項でも述べたように、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、Birthday 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 Birthday 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。

2.2 ソフトウェア(SW)実装評価

CRYPTREC では実装評価はしていないが､報告書作成時において[4]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 (850[MHz])

OS および使用言語: Window 2000 SP1, C++

処理性能 : 30.725 [Mbyte/sec]

2.3 ハードウェア(HW)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] http://www.eskimo.com/~weidai/benchmarks.html

6.2.3 SHA-1

1. 暗号技術

1.1 技術概説

SHA-1 は NIST(National Institute of Standards and Technology)によって提案された SHA(Secure Hash Algorithm)を改良したハッシュ関数である。SHA-1 の特徴は入力メッセージをブロック暗号における鍵のように考えて使うところにある。すなわち、入力メッセージを用いて各ステップで新しいメッセージを生成し、それを入力としてステップ関数を作用させる。この新しいメッセージの生成はメッセージ操作に基づく攻撃に耐性を持つと考えられる。SHA-1 は各メッセージブロックに対し、4 ラウンド 80 ステップの演算を行い、160bit のハッシュ値を出力する。

1.2 技術仕様

SHA-1 は、任意長のメッセージを入力し、160 bit のハッシュ値を出力する。入力されたデータは 512 bit のブロックとして処理される。処理の流れは次の 5 つのステップとなる。

- Step1: パディングビットの付加

  MD5 と同様に、入力メッセージは、そのビット長が 448 mod 512 bit となるように加工する。このとき 448 mod 512 の長さを満たしていたとしても、付加ビットは必ず付けられる。パディングビットは、先頭が“ 1 ”で必要な数だけ“ 0 ”という形となり、メッセージビットの直後に付加される。

- Step2: 長さ情報の付加

  MD5 と同様に、パディングされる前の元の入力メッセージのビット長の mod 2^{64} をパディングビットのあとに 64 bit の長さで付加する。

ここまでで、元のメッセージは512 bitの倍数のビット長に変換されている。拡大されたメッセージは、512 bit のブロック系列 $Y_0, Y_1, \ldots Y_{L-1}$ のように表現でき、合計ビット長は $L \times 512$ bit となる。

- Step3: バッファの初期化

  ハッシュ値を保存するバッファ（連鎖変数）を 5 個の 32 bit レジスタを使って (A, B, C, D, E) と表現する。この 160 bit のバッファは、ハッシュ値の中間値を保存して圧縮関数の入力として利用されたり、最終結果を格納する。最初の圧縮関数に入る前に、これらのレジスタに次の値が格納される。

$$A = 0\text{x}67452301$$
$$B = 0\text{xEFCDAB89}$$
$$C = 0\text{x}98\text{BADCFE}$$
$$D = 0\text{x}10325476$$
$$E = 0\text{xC3D2E1F0}$$

これらの値は RIPEMD-160 と同じである。そして最初の 4 つは MD5 と同じである。

- Step4: 圧縮処理

  SHA-1 のメインは、20 ステップを 1 ラウンドとした 4 ラウンドからなる圧縮関数である。この 4 ラウンドはそれぞれ同様の構造をしているが、各々 $f1, f2, f3, f4$ という異なる論理演算関数を使って圧縮を行う。各ラウンドは現在処理している 512 ビットブロック Y_q と 160 ビットバッファの値を入力とし、各ラウンドの出力値で 160 ビットバッファ $ABCDE$ の値を更新していく。

  4 ラウンド目（80 ステップ目）の出力は、その圧縮関数の 1 ラウンド目の入力 (CV_q) と加算され、CV_{q+1} となり、次の圧縮関数で利用される。この加算は 5 つのバッファで独立な $\bmod 2^{32}$ で加算される。

- Step5: メッセージダイジェストの出力

  L 個の 512 ビット長ブロックを圧縮関数ですべて処理した後、160 bit バッファ内にあるハッシュ値を出力する。

2. 評価結果

2.1 安全性評価

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 bit 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、バースデー攻撃により $2^{n/2}$ 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 bit 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。

7. 擬似乱数生成法の評価

対象の擬似乱数生成法は、下表である。7.1 節では、擬似乱数カテゴリについて、その特徴、安全性、実装性の観点から相互比較し、短評を述べる。7.2 節では個別擬似乱数生成法に関し、それらの観点のより詳細な記述を行う。

擬似乱数生成名

擬似乱数生成	TOYOCRYPT-HR1、 Pseudo-Random Number Generator based on SHA-1 (FIPS186:DIGITAL SIGNATURE STANDARD APPENDIX C)

7.1 種別による評価

擬似乱数生成の一覧表

		TOYOCRYPT-HR1	Pseudo-Random Number Generator based on SHA-1
特徴		128 段線形フィードバックシフトレジスタと非線形ブール論理関数とを併用する、非線形フィルタ・ジェネレータに分類される擬似乱数生成アルゴリズムである。	FIPS180-1 で規定した Secure Hash Algorithm (SHA-1) による擬似乱数生成器である。
安全性	統計的性質 擬似乱数の用途に対する適合性:推測可能評価出力系列に対する入力空間の大きさ	64 bit の既知平文による分割統治的攻撃法は、LFSR 既知の場合:$O(2^{95})$、未知の場合:$O(2^{215})$で可能である。さらに、秘密鍵長 128 bit に対する有効長は 96 bit であるので、2^{32} bit の出力系列に基づく、タイム・メモリ・トレードオフ法（2^{64} 回程度の計算時間と 2^{32} bit 程度の記憶容量）が存在し得る。以上の安全性評価の結果、TOYOCRYPT-HR1 は、電子政府用として推薦できないと判断した。 SHA-1 は 2^{80} 個のハッシュ値に対して衝突する可能性があるので、ハッシュ値を十分長くする必要がある。Pseudo-Random Number Generator based on SHA-1 は電子政府用として現在においては問題無いが長期の使用には注意が必要である。 できれば、2001 年の FIPS として規格化されるであろう、160 bit 以上のハッシュ値を出力する次世代 SHA（SHA-256、SHA-384、 SHA-512）を利用した擬似乱数生成器の方が望ましい。	

7.2 個別評価

7.2.1 TOYOCRYPT-HR1

1. 暗号技術

1.1 技術概説

TOYOCRYPT-HR1 は、東洋通信機株式会社から提案された、非線形・フィルタ・ジェネレータに分類される擬似乱数生成アルゴリズムである。128 bit の固定鍵と 128 bit のストリーム鍵（内部状態）を秘密情報とし、1 ビットずつ擬似乱数を生成する構造を有する。TOYOCRYPT-HR1 は、2000 年 9 月 29 日に仕様公開されたストリーム暗号 TOYOCRYPT-HS1 の中で用いられている擬似乱数生成手法であり同一の技術とみなせる。非線形コンバイナによく利用されるのはフィボナッチ・コンフィギュレーションであるが、ガロア・コンフィギュレーションがハードウェアで高速動作することに注目し、これを採用することにした、としている。安全性については、擬似乱数系列の長周期性、0/1 の等頻度性、線形複雑度、非線形度等を理論的に解析し、また、相関攻撃を代表とする非線形コンバイナ特有の攻撃に対して十分に安全となるように非線形変換関数を設計した、としている。また、ハードウェアパフォーマンスを重視した設計がなされているので、特に、小規模、省電力性を要求するハードウェアで、超高速な擬似乱数生成が必要なアプリケーションにおいて最も効果を発揮する、としている。また、TOYOCRYPT-HR1 は非常にシンプルな構造を有するので、ソフトウェアでは、コンパクトな実装が可能である、としている。

知的財産権 （応募者資料による）

(提案者特許とその扱い)

特願平 10-039677 （公開番号 平 11-224183）擬似乱数発生装置

特願平 10-129606 （公開番号 平 11-305661） シフトレジスタ構成用回路およびシフトレジスタ

特願平 10-267415 （公開番号 2000 年-081969）擬似乱数発生装置

応募者は、ライセンス方針について保留している。

応募暗号技術仕様の公開 Web アドレス

http://www.toyocom.co.jp

1.2 技術仕様

TOYOCRYPT-HR1 は非線形コンバイナに分類される擬似乱数生成アルゴリズムである。128 bit の固定鍵と 128 bit のストリーム鍵（内部状態）を秘密情報とし、1 bit ずつ擬似乱数を生成する構造を有する。非線形コンバイナは、線形フィードバックシフトレジスタの出力を非線形変換関数を介して出力することにより、1 bit ずつ擬似乱数を生成する。

TOYOCRYPT-HR1 の設計目標は、

(1)安全性を理論的に解析できること、

(2)ハードウェアにおいて、高速かつ小規模であること、

(3)構造がシンプルであること、

である、としている。さらに、

TOYOCRYPT-HR1 は以下の設計基準に基づいて設計した、としている。

(1) 0/1 の等頻度性を理論的に保証する。

(2) 長周期性を理論的に保証する(周期の下限を保証)。

(3) 線形複雑度の下限を理論的に保証し、滑らかに上昇するようにする。

(4) 非線形コンバイナ特有の攻撃に対し十分耐性のある構成にする。

(5) ハードウェアで構成した場合、高速動作し、十分小さな規模となるようにする。

上記基準の内、(1)～(4)は擬似乱数生成アルゴリズムの安全性の要件即ち、生成された擬似乱数の部分系列から、他の部分ビットを推定困難とするための条件であり、特に、(4)については、代表的な攻撃である、Fast Correlation Attack, Inversion Attack, Conditional Correlation Attack を考慮した、としている。

1.3 その他

特になし。

2. 評価結果

2.1 安全性評価

近い将来の計算機環境で解読可能となり得るため、現状のままでは電子政府用の暗号として推薦できない。提案アルゴリズムに、何がしかの改善を行ってから、実システムには採用すべきであると考える。実質的な鍵長は、固定鍵が既知ならば、128 bit から高々96 bit に激減する。これは、現在の計算機処理能力では安全な範疇であるが、将来を見越した安全性は運用者が判断する必要がある。運用上の課題として、LFSR の Seed を如何にして与えるか、少なくともその指針を明確にしておくべきである。Seed がもれることが、この乱数発生器の安全性に影響する。

詳細評価では、得られた乱数列の統計的性質として、長周期性、連、0/1 の等頻度性、自己相関等の白色雑音性等々の性質並びに線形複雑度については、満足すべき妥当な結果が得られたと報告されている。また、エントロピーテスト等に基づく通常の攻撃方法や、ストリーム暗号に対する代表的な攻撃法である相関攻撃に関しても、幾つか試みたが、有効な攻撃法には至らなかったと報告されている。

しかしながら、次のようなアルゴリズムとしての構造的欠陥も報告されており、特に以下の(1)、(2)に関しては、何がしかの改善を行ってから、実システムには採用すべきであると考えられる。

（1） 実質的な鍵長は、固定鍵が既知ならば、128 bit から高々96 bit に激減する。未知ならば、248 bit から高々215 bit に減る。さらに、この攻撃に必要な既知平文は僅かに 64 bit である。これは、分割統治攻撃やタイム・メモリ・トレードオフ法に繋がる。

＃類似の報告は SCIS2001 の次の報告にもある。

Mihaljevic, Imai, Effective Secret Key Size of TOYOCRYPT-HS1 Stream Cipher, PP.665-667,

Proc. of SCIS2001 年 2 月 7 日

解析結果は、2^{32} bit の出力データから、メモリ 2^{32}、計算量 2^{64} 程度で 128 bit の鍵が求まるとされている。

これを避けるために、g 関数やπ関数への入力を若干工夫するだけで、上記のような激減は避けられるとの報告があるので、何がしかの改善をおこなって使用すべきである。

（2） 固定鍵を乱数として発生する鍵として扱うと、on-line での応用によっては、それを生成するのに時間がかかりすぎ用をなさない場合が考えられる（例えば、暗号プロトコルでの共通鍵生成など)。そのため、固定鍵は最初から与えられているものとして扱う場合が考えられる。その場合は上記(1)の問題が顕在化する。運用上の課題として、LFSR の Seed を如何にして与えるか、少なくともその指針を明確にしておくべきである。Seed がもれることは、この乱数発生器の安全性に直接に影響する。

2.2 ソフトウェア(SW)実装評価

自己評価書では、次の結果となっている。

プログラム言語	ANSI-C
対象プロセッサ	日立製 SH1 （32 ビット RISC プロセッサ）
評価環境	日立製 SH1 コンパイラシミュレータ Windows95 Ver. 5.0
コンパイル条件	速度優先、ループ展開インライン展開なし

	シリアル実装	パラレル実装
実行速度	66991cycle/word,9.56Kbps(at 20MHz)	3678cycle/word,174Kbps(at 20MHz)
鍵セットアップ	591112cycle, 30.0ms(at 20MHz)	673332cycle, 33.7ms(at 20MHz)
プログラムコード	910Byte	800Byte
定数（鍵を含む）	368Byte	1360Byte
スタック（RAM）	1184Byte	2052Byte

TOYOCRYPT-HR1 はハードウェアで最もパフォーマンス（小規模性、高速性）が得られように設計されている。シリアル実装は、Bitslice 実装を利用して、効率はあまりよくないが、通常の実装と同等な出力が得られる方法であり、パラレル実装は、効率よくプロセッサのワード長分の独立な擬似乱数データを同時に出力する方法であるとしている。

2.3 ハードウェア(HW)実装評価

TOYOCRYPT-HS1 の評価として、アルテラ社の FPGA「EP20K600E」を使用した評価が行われているので、そちらを参照されたい。

自己評価書の、FPGA によるシミュレーション評価結果は以下の通りである。

シミュレーションツール ：SILOS Ⅲ ver 99.070 (SIMCAD.Inc)

ターゲットデバイス：QL3012-4、デバイスプロセス：C-MOS 0.35μm 4層

コンパイルオプション	項目	小項目	性能
速度優先	実行速度	Worst case	227MHz(Mbps)
		Normal case	340MHz(Mbps)
		Best case	418MHz(Mbps)
	回路規模	Cell	233/320 (71.6%)
		Routing resources	3340/42812 (7.8%)
回路規模優先	実行速度	Worst case	81MHz(Mbps)
		Normal case	122MHz(Mbps)
		Best case	150MHz(Mbps)
	回路規模	Cell	196/320 (61.3%)
		Routing resources	2928/42812 (6.8%)

自己評価書の、ゲートアレイによる性能見積もりの結果は以下の通りである。

ゲートアレイ（東芝製 TC200G52, 3.3V, 3 層 0.4μm プロセス）で構成した場合、約 3.3Kgate で 223Mbps(Worst Case, 70℃, 3.0V)を達成できるとしている。

条件	遅延[ns]	速度[MHz]
Worst case	4.5	223
Normal case	2.5	403
Best case	1.3	790

参考文献

[1] 杉本浩一、力石徹也、森住徹也, ストリーム暗号の設計法と評価, 信学技報, ISEC2000-69

7.2.2 PRNG based on SHA-1

(FIPS186-2:DSS Appendix 3.Random number Generator for the DSA)

1. 暗号技術

1.1 技術概説

Digital Signature Algorithm (DSA) では、ユーザの秘密鍵 x およびメッセージ署名毎の秘密鍵 k や $r = (g^k \bmod p) \bmod q$ （ p , q , g は公開パラメタ）が必要となる。

FIPS186（1994 年 5 月）(2000 年 1 月改定、FIPS186-2)、Digital Signature Standard (DSS) では、これらを生成するために、3 種類の擬似乱数生成法

(1) ANSI X9.31 の Appendix 2.4 の“Financial Institution Key Management (Wholesale)”による 160 ビットの一方向性関数 $G(t,c)$（t は 160 ビット、c は b ビットである。$G(\bullet)$ が SHA-1 による場合、$160 \le b \le 512$、$G(\bullet)$ が Data Encryption Algorithm (DEA)による場合(ANSI X9.17 の Appendix C では DES を使用)、$b = 160$ 固定）

(2) FIPS186-2 の 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) FIPS186-2 の 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 推奨版として規定している。

FIPS 186 では FIPS180 規格の Secure Hashing Algorithm (SHA)の使用を推奨、その後 1995 年 4 月に FIPS180-1 規格として、Secure Hash Standard(SHS)を規定し、SHS の唯一の推奨版として、160 ビット長のハッシュ値を生成する Secure Hash Algorithm (SHA-1)の仕様を明示。

NIST は暗号応用分野で使用される 2 値系列の乱雑さ検定のための各種統計テストの充実、各種テストのソフトウエア実装、テストの応用等のために、Random Number Generation and Testing をウェブ公開している。

1.2 技術仕様

（FIPS186-2 の 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) SHS での 512 ビット $H_0 \| H_1 \| \cdots \| H_4$ の初期値

$$t = 67452301 \| EFCDAB89 \| 98BADCFE \| 10325476 \| C3D2E1F0$$

を選択する。

(3) $0 \le j \le m-1$ として以下の(a)-(d)を繰り返す。

(a) ω_j を選択（ユーザーオプション）する。

(b) $c_j = (\omega_{xkey} + \omega_j) \bmod 2^b, 160 \le b \le 512$

(c) $x_j = G(t, c_j) \bmod q$

(d) $\omega_{xkey} = (1 + \omega_{xkey} + x_j) \bmod 2^b$

（FIPS186-2 の Appendix 3.2 の m 種類の r , k の生成法の技術仕様）

(1) 新しい秘密数 ω_{kkey} を選択する。

(2) SHS での 512 ビット $H_0 \| H_1 \| \cdots \| H_4$ の初期値をシフトした

$$t = EFCDAB89 \| 98BADCFE \| 10325476 \| C3D2E1F0 \| 67452301$$

を選択する。

(3) $0 \le j \le m-1$ として以下の(a)-(d)を繰り返す。

(a) $k = G(t, \omega_{kkey}) \bmod q$

(b) $k_j^{-1} = k^{-1} \bmod q$ を計算する。

(c) $r_j = (g^k \bmod p) \bmod q$

(d) $\omega_{kkey} = (1 + \omega_{kkey} + k) \bmod 2^b$

(4) m 個のメッセージを $M_1, M_2, \cdots, M_{m-1}$ として以下の(a)-(c)を繰り返す。

(a) $h =$ SHA-1(M_j)、SHA-1(・)は SHA-1 による一方向性関数を意味する。

(b) $s_j = (k_j^{-1}(h + xr_j)) \bmod q$ を計算する。

(c) (r_j, s_j) を M_j の署名とする。

(5) $t = h$

(6) (3)に戻る。

（SHA-1 による一方向性関数 $G(t,c)$ の技術仕様）

$G(t,c)$ は、下記(II)に掲げる Secure Hash Standard (SHS)の技術仕様の Sec.7 の手順(a)-(e)で計算できるが、これらの実行前に $\{H_j\}$ の初期化および入力メッセージパディングを以下の手順で行う。

(I) $\{H_j\}$の初期化とcのパディング

(i) 160 ビットのtの 32 ビット分割$t = t_0 \| t_1 \| \cdots \| t_4$で$H_j = t_j$,（$0 \le j \le 4$）とおく。

(ii) $X = c \| 0^{512-b}$

次に、以下の手順(a)-(e)を実行して手順(e)で得られた 160 ビットの文字列 $G(t,c) = H_0 \| H_1 \| \cdots \| H_4$ を出力とする。

(II) SHS の技術仕様の Sec.7 の手順 上記のパディングされたメッセージXは$16 \times n$の words(1 word は 32 ビット)からなる。これらのnブロック（1 ブロックは 16words、512 ビット）を$M_1, M_2, \cdots, M_n$とする。最初のブロックM_1は入力メッセージcの最初の数ビットを含む。
5 個の 32 ビットの words の最初のバッファ群A, B, C, D, E, Fと 2 番目のバッファ群$H_0, H_1, \cdots, H_4$および 80 個の words 群$W_0, W_1, \cdots, W_{79}$に対して、$H_0 \| H_1 \| \cdots \| H_4$の初期値を

$$t = 67452301 \| EFCDAB89 \| 98BADCFE \| 10325476 \| C3D2E1F0$$

と設定する。
n個のメッセージを$M_1, M_2, \cdots, M_n$が処理されたとする。M_iを処理するために、以下の(a)-(e)の手順を実行する。

(a) M_iを 16 個の words に分割 $M_i = W_0 \| W_1 \| \cdots \| W_{15}$

(b) $W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$, $16 \le t \le 79$.

ただし、$\oplus$は排他的論理和を意味し、$S^n(X)$は word X の左n $(0 \le n \le 32)$ビット巡回シフト演算$S^n(X) = (X << n) \cup (X >> 32 - n)$を意味する。
(c) $A = H_0, B = H_1, C = H_2, D = H_3, E = H_4$とする。
(d) $0 \le t \le 79$に対し、以下の計算をする。
64 ビット以下のビット長$\ell \le 2^{64} - 1$のビット文字列xに対し、SHA-1 に入力する前に以下のパディング

$$X = x \| 1 \| 0^m \| \ell$$

を行う。ただし、$m + \ell + 1 = 448 \bmod 512$である。

$$TEMP = S^5(A) + f_t(B,C,D) + E + W_t + K_t;$$

$$E = D; D = C; C = S^{30}(B); B = A; A = TEMP$$

ただし、$f_t(B,C,D)$は以下で定義される関数

$$f_t(B,C,D) = \begin{cases} (B \cap C) \cup (\overline{B} \cap D), & (0 \le t \le 19) \\ B \oplus C \oplus D, & (20 \le t \le 39) \\ (B \cap C) \cup (B \cap D) \cup (C \cap D), & (40 \le t \le 59) \\ B \oplus C \oplus D, & (60 \le t \le 79) \end{cases}$$

である。ただし、$\bar{\bullet}$, $\cap$, $\cup$ はそれぞれ、論理否定、論理積、論理和を意味する。また、K_t は以下で定義される定数

$$K_t = \begin{cases} 5A827999, & (0 \le t \le 19) \\ 6ED9EBA1, & (20 \le t \le 39) \\ 8F1BBCDC, & (40 \le t \le 59) \\ CA62C1D6, & (60 \le t \le 79) \end{cases}$$

(e)

$$\left.\begin{array}{l} H_0 = H_0 + A \\ H_1 = H_1 + B \\ H_2 = H_2 + C \\ H_3 = H_3 + D \\ H_4 = H_4 + E \end{array}\right\}$$

(III) 最終メッセージ M_n を処理後、160 ビットの $H_0 \| H_1 \| \cdots \| H_4$ をハッシュ値として出力する。

1.3 その他

A. CMV Program

NIST による Cryptographic Module Validation(CMV) Program では、Digital Signature Standard(DSS) や Secure Hash Standard(SHS)の規格作りのために Digital Signature Validation System(DSSVS), v2.3 を利用して、DSA,RSA,SHA-1 等の評価を実施している。2001 年の FIPS として、次期改定版の DSA や AES で用いるための、より長いメッセージダイジェストを出力可能な三種類の次世代 SHA が 2001 年の FIPS の草案として提案される予定である。

B. 次世代 SHA の技術仕様

AES では、セキュリティを強化した同程度の新しいハッシュ関数として、SHA-256、 SHA-512、SHA-384

の 3 種類が提案されている。

(1)SHA-256 MD4,MD5,SHA-1 と同様な設計である。

(ｉ)ビット文字列 x から 512 ビット長のパディングされたメッセージ

$$M = x\|1\|0^k\|\ell$$

を計算する。但し、x は 2^{32} を法とした整数、$\ell = |x| \bmod 2^{64}$, $\ell + 1 + k \equiv 448 \bmod 512$ である。

(ii) M を $N \equiv 0 \bmod 16$ 個の 512 ビット単位のブロック $\{M^{(i)}\}_{i=1}^{N}$ に分割する。但し、$M^{(i)}$ は 16 個の 32 ビット長のワード（word）からなる。

(iii)初期ハッシュ値 $H^{(0)}$ と SHA-256 圧縮関数 $C_{\cdot}(\cdot)$ と SHA-256 メッセージスケジュール関数を用いて以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}(H^{(i)}) , 1 \le i \le N$$

を計算する。ただし、＋は 32 ビットの word 単位毎の 2^{32} を法とした加算を意味する。$H^{(N)}$ が M のハッシュ値である。

(2)SHA-512 SHA-256 の word 長 32 を 64 に変更したものである。

(i)ビット文字列 x から 1024 ビット長のパッデングされたメッセージ

$$M = x\|1\|0^k\|\ell$$

を計算する。但し、x は 2^{64} を法とした整数、$\ell = |x| \bmod 2^{128}$, $\ell + 1 + k \equiv 896 \bmod 1024$ である。

(ii) M を $N \equiv 0 \bmod 16$ 個の 1024 ビット単位のブロック $\{M^{(i)}\}_{i=1}^{N}$ に分割する。但し、$M^{(i)}$ は 16 個の 64 ビット長のワード（word）からなる。

(iii)初期ハッシュ値 $H^{(0)}$ と SHA-512 圧縮関数 $C_{\cdot}(\cdot)$ と SHA-512 メッセージスケジュール関数を用いて以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}(H^{(i)}) , 1 \le i \le N$$

を計算する。ただし、＋は 64 ビットの word 単位毎の 2^{64} を法とした加算を意味する。$H^{(N)}$ が M のハッシュ値である。

(3)SHA-384 SHA-512 と殆ど同じで以下の２点だけが異なる。

①初期ハッシュ値 $H^{(0)}$ の変更、②512 ビットのハッシュ値 $H^{(N)}$ の左 384 ビットで打ち切ったものをハッシュ値とする。

C. Cryptographic Toolkit

Cryptographic Toolkit によると、乱数生成法は一般に、Random Number Generator (RNG) (非決定論的生成器と通称される“真の”乱数器)と Pseudorandom Number Generators (PRNGs)（決定論的生成法と通称される）とに大別される。前者は電気回路の雑音や計算機ユーザーの鍵ストローク、マウス移動等のタイミング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。しかし現在の所、後者の FIPS 規格は存在しない。PRNG の数列は物理源から生成された RNG より、生成速度も速いし、「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。

NIST は、Encryption、Modes of Operation、Digital Signatures、Secure Hashing、Key Management、Random Number Generation、Message Authentication、Entity Authentication、Password Usage and Generation 等の各種の Cryptographic Toolkit について、アメリカ政府や他の団体が暗号セキュリテイ技術を選定する際の標準化、推奨、ガイドライン作りのための包括的基準として充実させている。

D. RNG Testing

NIST Special Publication (SP) 800-22、A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications（2000 年 12 月改定）で乱数検定のために各種の統計的検定法、それらの使用法、検定結果の解釈、他の擬似乱数生成法等の情報を提供している。

参考文献

・Cryptographic Toolkit:http://csrc.nist.gov/encryption/
・Secure Hashing:http://csrc.nist.gov/encryption/tkhash.html
・Random Number Generation:http://csrc.nist.gov/encryption/tkrng.html
・New hashing algorithms(SHA-256,SHA-384, and SHA-512): Descrptions of SHA-256,SHA-384, and SHA-512
・FIPS Pub 186-2,Digital Signature Standard (DSS) (2000January 27 更新) Appendix3:Random Number Generation for the DSA
・NIST Special Pubilication 800-22(Dec. 2000 年更新）A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications
・FIPS140-1, Security Requirements for Cryptographic Modules,Cryptographic Module Validation(CMV) Program: http://csrc.nist.gov/cryptval/cmvp.html

2　評価結果

2.1　安全性評価

A. 概要： FIPS 規格の擬似乱数生成器(with SHA-1)を電子政府で利用することには問題ないと考える。 ただし、適用する暗号方式（電子署名）で、ランダムオラクルの実現として、この SHA-1 ベースの擬似乱数を利用すると、理論的証明の仮定が成立しないことに注意すべきである。

しかし、擬似乱数の利用される応用にもよるが、実装環境の制限等により、SHA-1 ベースになっても、現状問題はないといえる。ただし、SHA-1 自身を電子署名用ハッシュ関数に利用する場合には、バースデー攻撃により 2^{80} 程度の計算量で改ざんの危険性がある。これは、今日最低の安全性ラインである。できれば、今後 1、2 年内に規格化されるであろう長い出力の次世代 SHA を利用したほうが望ましい。

実装環境に制限がある場合には、SHA-1（あるいは DES）ベースの FIPS186 擬似乱数生成法でも、生成する擬似乱数の長さや回数を考慮すれば、安全に利用できるといえる。

B. DSA flaw ベル研の Daniel Bleichenbacher 研究員が指摘した DSA の乱数に偏りを発見したという DSA flaw のニュース（2001 年 2 月 6 日）が流されたが、DSA flaw に関しては、SHA-1 によるものではないらしいとの報告がある。

DSA flaw に関する ANSI X9F1 会議（X9.30 DSA 標準化を取り扱っている）での議論による正式な NIST の回答は以下の通り。

- 2 百万以下の署名であれば問題無し。
- SW 固定にするのであれば、ハッシュ長の少なくとも 2 倍の k を生成するために ECDSA を使用すること
- k を計算するために、32 ビット以上の追加ビット、すなわち、160 ビットの k のために 192 ビット以上のランダムビットが必要である。

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

8. その他

8.1 評価暗号一覧

（1）公開鍵暗号一覧表

番号	暗号種別	暗号名	評価結果
1	公開鍵暗号（守秘）	ACE Encrypt	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
2	公開鍵暗号（守秘）	ECAES in SEC1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
3	公開鍵暗号（守秘）	EPOC	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
4	公開鍵暗号（守秘）	ESIGN identification	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
5	公開鍵暗号（守秘）	HIME-2	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
6	公開鍵暗号（守秘）	PSEC	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
7	公開鍵暗号（守秘）	RSA-OAEP	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
8	公開鍵暗号（守秘）	素数が無限に存在することを証明する数式を用いた暗号	公募により提案された暗号でスクリーニング評価を実施 ［公募カテゴリーに該当していない（公開鍵暗号技術のカテゴリーとして成立しているとは認められない)］と判定
9	公開鍵暗号（署名）	ACE Sign	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
１０	公開鍵暗号（署名）	DSA	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第4章参照
１１	公開鍵暗号（署名）	ECDSA in SEC1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
１２	公開鍵暗号（署名）	ESIGN - signature	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
１３	公開鍵暗号（署名）	MY-ELLTY ECMR-160-h	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照
１４	公開鍵暗号（署名）	MY-ELLTY ECMR-160-p	公募により提案された暗号でスクリーニング評価を実施 ［安全性が十分でない（パディングの方法が安全でない)］と判定
１５	公開鍵暗号（署名）	MY-ELLTY ECMR-192-h	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第4章参照

１６	公開鍵暗号（署名）	MY-ELLTY ECMR-192-p	公募により提案された暗号でスクリーニング評価を実施 ［安全性が十分でない（パディングの方法が安全でない)］と判定
１７	公開鍵暗号（署名）	MY-ELLTY ECMR-OEF-h	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第４章参照
１８	公開鍵暗号（署名）	MY-ELLTY ECMR-OEF-p	公募により提案された暗号でスクリーニング評価を実施 ［安全性が十分でない（パディングの方法が安全でない)］と判定
１９	公開鍵暗号（署名）	RSA-PSS	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第４章参照
２０	公開鍵暗号（鍵共有）	DH	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第４章参照
２１	公開鍵暗号（鍵共有）	ECDHS in SEC1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第４章参照
２２	公開鍵暗号（鍵共有）	ECMQVS in SEC1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第４章参照
２３	公開鍵暗号（鍵共有）	EPOC 鍵共有	公募により提案された暗号でスクリーニング評価を実施 「「守秘」のカテゴリーで応募された暗号と同一の技術仕様記述であるため、「鍵共有」カテゴリーとしては詳細評価は行わない」と判定
２４	公開鍵暗号（鍵共有）	HDEF-ECDH	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第４章参照
２５	公開鍵暗号（鍵共有）	HIME-1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第４章参照
２６	公開鍵暗号（鍵共有）	PSEC 鍵共有	公募により提案された暗号でスクリーニング評価を実施 「「守秘」のカテゴリーで応募された暗号と同一の技術仕様記述であるため、「鍵共有」カテゴリーとしては詳細評価は行わない」と判定

（２）共通鍵暗号一覧表

番号	暗号種別	暗号名	評価結果
1	共通鍵暗号(64ビットブロック暗号)	CIPHERUNICORN-E	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第５章参照
2	共通鍵暗号(64ビットブロック暗号)	FEAL-NX	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第５章参照
3	共通鍵暗号(64ビットブロック暗号)	Hierocrypt-L1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第５章参照

4	共通鍵暗号 (64ビットブロック暗号)	MISTY1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
5	共通鍵暗号 (64ビットブロック暗号)	Triple DES	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第5章参照
6	共通鍵暗号 (128ビットブロック暗号)	Camellia	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
7	共通鍵暗号 (128ビットブロック暗号)	CIPHERUNICORN-A	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
8	共通鍵暗号 (128ビットブロック暗号)	FS-CES	公募により提案された暗号でスクリーニング評価を実施 ［(ブロック暗号としての) 処理速度が十分でない］と判定
9	共通鍵暗号 (128ビットブロック暗号)	Hierocrypt-3	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
１０	共通鍵暗号 (128ビットブロック暗号)	K-7	公募により提案された暗号でスクリーニング評価を実施 ①［応募カテゴリーに該当していない］、②［第三者が実装・評価するための記述が十分でない］と判定
１１	共通鍵暗号 (128ビットブロック暗号)	MARS	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
１２	共通鍵暗号 (128ビットブロック暗号)	RC6	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
１３	共通鍵暗号 (128ビットブロック暗号)	Rijndael	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第5章参照
１４	共通鍵暗号 (128ビットブロック暗号)	SC2000	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
１５	共通鍵暗号 (128ビットブロック暗号)	SXAL/MBAL	公募により提案された暗号でスクリーニング評価を実施 ［自己評価が十分でない］と判定
１６	ストリーム暗号	FSAngo	公募により提案された暗号でスクリーニング評価を実施 ［安全性が十分でない］と判定
１７	ストリーム暗号	GCCカオス暗号	公募により提案された暗号でスクリーニング評価を実施 ［第三者が実装・評価するための記述が十分でない（アルゴリズム評価が不能)］と判定
１８	ストリーム暗号	MDSR	公募により提案された暗号でスクリーニング評価を実施 ［自己評価が十分でない］と判定
１９	ストリーム暗号	MULTI-S01	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照
２０	ストリーム暗号	SEAL	応募取りやめの通知を受けたため、評価対象外
２１	ストリーム暗号	TOYOCRYPT-HS1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第5章参照

（3）ハッシュ関数一覧表

番号	暗号種別	暗号名	評価結果
1	ハッシュ関数	MD5	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第 6 章参照
2	ハッシュ関数	RIPEMD-160	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第 6 章参照
3	ハッシュ関数	SHA-1	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第 6 章参照

（4）擬似乱数生成一覧表

番号	暗号種別	暗号名	評価結果
1	擬似乱数生成	FSRansu	公募により提案された暗号でスクリーニング評価を実施 ［安全性が十分でない］と判定
2	擬似乱数生成	Pseudo-Random Number Generator based on SHA-1	公募への提案の有無にかかわらず評価が必要な暗号と認定、詳細評価を実施 評価結果は本報告書第 7 章参照
3	擬似乱数生成	TAO TIME（タオタイム）コード生成システム	公募により提案された暗号でスクリーニング評価を実施 「暗号技術仕様書が提出されていないため、評価を行うことができない」と判定
4	擬似乱数生成	TOYOCRYPT-HR1	公募により提案された暗号で、スクリーニング評価、詳細評価を実施 評価結果は本報告書第 7 章参照
5	擬似乱数生成	tsw-b	公募により提案された暗号でスクリーニング評価を実施 ①［安全性が十分でない］、②［第三者が実装・評価するための記述が十分でない］と判定
6	擬似乱数生成	メルセンヌ ・ツイスター	公募により提案された暗号でスクリーニング評価を実施 ［第三者が実装・評価するための記述が十分でない］と判定

8.2 暗号標準化関連の動き

8.2.1 DES/AES について

関連 Web アドレス　http://csrc.nist.gov/encryption/

(1) DES

DES (Data Encryption Standard)は、National Institute of Standards and Technology (NIST)の前身である U.S. National Bureau of Standard (NBS) がコンピュータのデータや通信の保護を目的として、十分に安全性を検証し、かつ、インターオペラビリティを持ち、実用に耐えうるものとして制定した米国政府標準の暗号アルゴリズムである。

DES は FIPS(FIPS46)、ANSI(X3.92) として過去 20 年以上にもわたり使われてきた共通鍵暗号アルゴリズムであり、商用レベルの暗号では初めてアルゴリズムを完全に公開した。

FIPS46-3 では DES は下位互換性のためのみ利用としており、暗号化・復号には Triple DES を利用することになっている。

DES Challenge

DES Challenge は米 RSA 社が主催した DES の解読コンテストである。数組の平文と暗号文（及び初期ベクトル）を与えて鍵を探すというルールなので、基本的には DES の鍵空間に対して総当り検索を争うコンテストと同等である。この DES Challenge は、1997 年から 1998 年にかけて合計４回行われた。1997 年１月 28 日開始された DES Challenge I は distributed.net のネットワーク分散方式により 140 日で解読された。この方法は distributed.net がインターネットに接続しているコンピュータをもっている多数のボランティアを募り、そのコンピュータ上で解読専用のソフトウェアを走らせる分散処理を行うというものである。1998 年 1 月 13 日から開始された DES Challenge II-1 も distributed.net が 39 日という期間で解読している。

1998 年 7 月 13 日から開始した DES Challenge II-2 では Electronic Frontier Foundation が DES Challenge のために設計した専用マシンを用意し 56 時間での解読に成功した。21 万ドルもの制作費かけた EFF DES Cracker は、鍵空間検索用に設計された専用 LSI AWT-6001 を 1536 個登載している DES Challenge 用の特殊な並列マシンである。1 つのチップは、さらに 24 個の鍵検索ユニットを内蔵している。この EFF DES Cracker プロジェクトのプロデューサーは EFF の John Gilmore だが、実際の開発は米 Cryptography Research 社と米 Advanced Wireless Technologies 社が共同であたった。EFF DES Cracker は毎秒約 900 億個分の鍵検索を行うので、56 ビットの鍵空間のすべてを検索するとしても９日強しかかからない。DES Challenge III では distributed.net と EFF が協力し 22 時間 16 分で解読する成果を出している。

(2) AES

AES (Advanced Encryption Standard)は DES に代わる米国の次世代標準暗号化アルゴリズムである。1990年代後半にもなると DES の安全性・信頼性が低下してきたため、Triple DES (FIPS 46-3)が利用されるようになる。ところが Triple DES は、DES を 3 回繰り返すため、効率的に問題があった。そこで今までの DES に代わって使うべく十分に安全で、かつ高速な暗号を選ぶ必要性が出てきた。また実際に使われている新しい暗号の多くは学術的な場で安全性を議論されており、ある程度安全だと認められているものが主であるが、それであっても徹底して安全性を議論・検証したものとはいえない。その点でも十分に安全性を議論し検証を重ねた、誰もが安心して使える暗号アルゴリズムが必要であった。

AES 選出のプロセス

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)が DES に代る次世代標準化暗号の選定計画を開始した。まず 1998 年中に AES 候補として世界中から公募を行い、そこから約 1 年をかけ評価し最終審査に残る 5 つの候補に絞る。さらに約 1 年をかけて、その 5 候補の評価を行い、AES を決める。後に AES に選出された暗号アルゴリズムは FIPS (連邦標準)化される。NIST が主催し、NSA が技術コンサルタントとして行われた。AES の審査プロセスにおいては関係するデータをすべて Web で公開している。最終審査に残る 5 候補はソースコードも含めて、アルゴリズムに関する資料すべてをダウンロードできるようになっている。またラウンド期間中、パブリックコメントを求めると同時に NIST の Web サイトではディスカッショングループを用意するなど、オープンな形で運営されていた。

2000 年 10 月、Rijndael が AES として選出された。現在（2001 年 3 月）は FIPS 文章のドラフトを公開し、パブリックコメントを求めている最中である。2001 年の 8 月から 10 月ぐらいには正式な FIPS として発行される予定である。

AES の技術的要求仕様

- 共通鍵暗号であること
- ブロック暗号であること
- ブロックサイズが 128bit をサポートしていること
- 鍵長が 128bit、192bit、256bit がサポートされていること

Rijndael 公式 Web ページ

http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

AES を決めるまでのプロセスは、ガラス張りのオープンプロセスだったといえる。DES の時のような不透明なプロセスはない。約 3 年近く、世界中の暗号学者から徹底的に調べられた上で、十分に安全であり、かつ、処理効率も良いということが明らかになっている AES(Rijndael)は、FIPS という規格という枠組みを越えてデファクトスタンダードの暗号アルゴリズムとして利用されていくだろう。また技術的な面よりも、米国の次世代標準化暗号がヨーロッパ生まれという事実がデファクトとなるのに心理的にプラスに働くように思われる。今後の Rijndael の広がり方は基本的に AES として FIPS 化された後、AES から ANSI 化、ISO 化、IETF RFC 化の道をたどるものと思われる。

8.2.2 NESSIE プロジェクトについて

関連 Web アドレス　http://cryptonessie.org/

NESSIE プロジェクト（New European Schemes for Signatures, Integrity and Encryption）は欧州における暗号アルゴリズム標準化活動であり、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るものである。NESSIE の計画は、2000 年 1 月から 2002 年 12 月までの 3 年間で、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、意欲的なものである。

NESSIE プロジェクトの主目的は、誰もが参加できる公募によって得られた暗号プリミティブ(ブロック暗号アルゴリズムや公開鍵暗号アルゴリズムなどに代表される暗号の部品)を、誰もが参加できる公開された手続きによって評価し､強い暗号プリミティブの一覧表を作成することである｡本プロジェクトは米国 NIST が主催する AES ブロック暗号標準化プロセスの最終段階への貢献を意図しているが、それだけではなく、守秘･完全性や認証に関するプリミティブも新たに募集する。より具体的には、これらのプリミティブは、ブロック暗号、ストリーム暗号、ハッシュ関数、メッセージ認証アルゴリズム、ディジタル署名方式、公開鍵暗号方式から構成されている。また本プロジェクトは安全性と処理性能に関する評価手法や、この評価を実施するソフトウエアツールを開発する。

NESSIE プロジェクトの目標は、その成果を広く普及させるとともに、公開討論の場を通してその成果にもとづいたコンセンサスを得ることにある。そして最終的には欧州の研究の強い地位を保つとともに、欧州産業の暗号に関する地位を強化することを目指すものである。

選定基準

NESSIE の選定基準は、長期間にわたる安全性、市場の要求、効率性（性能）と柔軟性である。より具体的には以下のようにアナウンスされている。

安全性はもっとも重要な基準である。これは暗号プリミティブの安全性は信頼性とコンセンサスを得るためには必須であるからである。この評価プロセスは本プロジェクトの外側からの進歩（たとえば新しい攻撃法や評価手法など）に影響されると考えられる。

2 番目の基準は市場の要求に関係する。市場の要求とは、プリミティブの必要性、可用性、世界的な利用の可能性にかかわる。

3 番目の基準は特定の環境でのプリミティブの性能である。ソフトウエアでは、この環境の範囲は、8 ビットプロセッサ（安価なスマートカードで使われるような)、32 ビットプロセッサ（例えば Pentium ファミリーなど）から、最新の 64 ビットプロセッサにわたる。ハードウエアでは、FPGA と ASIC の両方が考慮されることになる。

4 番目の基準は柔軟性である。広い範囲で利用できるプリミティブは明らかに望ましいものである。

募集プリミティブ

NESSIE では、広範にわたる暗号プリミティブを募集の対象としている。具体的には次の 10 種類のタイプに分類している。

【タイプ番号】	【タイプ名】
タイプ 1	ブロック暗号
タイプ 2	同期ストリーム暗号
タイプ 3	自己同期ストリーム暗号
タイプ 4	メッセージ認証符号（MAC）
タイプ 5	無衝突ハッシュ関数
タイプ 6	一方向ハッシュ関数
タイプ 7	疑似ランダム関数
タイプ 8	非対称暗号方式
タイプ 9	非対称ディジタル署名方式
タイプ 10	非対称認証方式

評価基準の詳細

評価基準は安全性の基準、実装性の基準、その他の基準、ライセンス条件の 4 つから構成され、それぞれ以下のように定められている。

安全性の基準

いかなる攻撃も、少なくともそのタイプのプリミティブに対する一般的な攻撃(全数探索やバースデー攻撃)と同程度の困難さでなければならない。プリミティブは、応募者が述べた安全性の主張に対して攻撃評価される。応募者の主張よりも少ない計算リソースで成立する攻撃があれば、通常それは応募暗号の価値を減ずることになる。プリミティブは、これまで述べた環境の範囲内で評価される。したがって、サイドチャネル攻撃（タイミング攻撃や Differential Power Analysis など）に対する安全性を考慮することは適当なことである。

実装性の基準

ソフトウエアとハードウエアの効率性は、類似の投稿と既存のプリミティブと比較で行われる。

実行コードとメモリーサイズは、それぞれ異なった条件下での妥当性という観点から評価される。スマートカードには特別の注目が払われる。

プリミティブは、応募者が述べた実装性の主張に対して評価される。もちろん幅広く柔軟に利用できるということは望ましいことである。

その他の基準

設計の単純性と明確性は重要な考慮点である。可変なパラメータということはあまり重要ではない。

ライセンス条件

応募されたプリミティブは、もしNESSIEに選択されたならば、無償(royalty-free)でなければならない。もしそれが可能でなければ、利用は非差別的(non-discriminatory)でなければならない。

応募者は、知的財産権に関する立場を表明しなければならない。その表明文書は必要に応じて更新されなければならない。

8.2.3 ISO/IEC JTC1/SC27/WG2 について

関連 Web アドレス　http://www.din.de/ni/sc27/

ISO/IEC JTC1/SC27 (ISO は International Organization for Standardization [国際標準化機構]、IEC は International Electrotechnical Commission[国際電気標準会議]、JTC1 は ISO と IEC が共同で設置した情報処理関連技術の国際規格の作成を担当する委員会、SC27 はその下部組織) は、情報セキュリティ技術全般の国際標準を決定する機関である。

SC27 の傘下には、３つの WG (技術検討部会) が設けられており、WG2 は IT セキュリティ技術とメカニズムとして、機密性確保・エンティティ認証・否認防止・かぎ管理・データ完全性（メッセージ認証、ハッシュ関数、デジタル署名など）の標準化をおこなっている。

ISO/IEC JTC1 では、国際標準規格（IS）発行までに次のドラフト段階を経る。

(1)New Work Item Proposal(NP)

(2)Working Draft (WD)

(3)Committee Draft (CD)

(4)Final CD (FCD)

(5)Final Draft of IS(FDIS)

(6)International Standard (IS)

暗号アルゴリズムの標準化の活動として、18033 が 1999 年 10 月より審議が開始され、以下の４つのパートよりなり、2003 年頃の IS 化を目指している。

-Part1 一般モデル

-Part2 公開鍵暗号

-Part3 ブロック暗号

-Part4 ストリーム暗号

共通鍵暗号関連の ISO 標準規格一覧表

ISO#	内容	TITLE	STATUS
8372	64 ビットブロック暗号アルゴリズムの利用モード	Modes of operation for a 64-bit block cipher algorithm	IS
10116	nビットブロック暗号アルゴリズムの利用モード	Modes of operation for an n-bit block cipher algorithm	IS
9798-1 9798-2	エンティティ認証	Entity authentication -Part1:General -Part2:Mechanisms using symmetric encipherment algorithms	IS
9797-1	メッセージ認証符号	Message authentication codes (MACs) - Part1: Mechanisms using a block cipher Part2: Mechanisms using a hash-functions	IS
13888-2	否認防止	Non-repudiation -Part2: Using symmetric techniques	IS
10118-2	ハッシュ関数	Hash-functions -Part2: Hash-functions using an n-bit block cipher algorithm	IS
11770-2	かぎ管理	Key management -Part2: Mechanisms using symmetric techniques(confirmed 1999)	IS
18033-3	ブロック暗号	Encryption algorithms -Part 3: Block ciphers	WD 準備中
18033-4	ストリーム暗号	Encryption algorithms -Part 4: Stream ciphers	WD 準備中

公開鍵暗号関連の ISO 標準規格一覧表

ISO#	内容	TITLE	STATUS
9796-1	メッセージ復元	Digital signature schemes giving message recovery	
9796-2	型ディジタル署	-Part1:廃止	-
9796-3	名	-Part2: Mechanisms using a hash-function	IS
		-Part3: Discrete logarithm based mechanisms	IS
9798-1	エンティティ認	Entity authentication	IS
9798-3	証	-Part1:General	IS
		-Part3:Mechanisms using asymmetric signature techniques	
11770-1	かぎ鍵管理	Key management	
11770-2		-Part1: Framework	IS
11770-3		-Part2: Mechanisms using symmetric techniques	IS
		-Part3: Mechanisms using asymmetric techniques	IS
13888-3	否認防止	Non-repudiation	IS
		-Part3:Using asymmetric techniques	
14888-1	添付型ディジタ	Digital signatures with appendix	
14888-2	ル署名	-Part1: General	IS
14888-3		-Part2: Identity-based mechanisms	IS
		-Part3: Certificate-based mechanisms	IS
15946-1	楕円曲線暗号	Cryptographic techniques on elliptic curves	
15946-2		-Part1: General	FDIS
15946-3		-Part2: Digital signatures	FDIS
15946-4		-Part3: Key establishment	FDIS
		-Part4: Digital signatures giving message recovery	CD
18033-1	公開鍵暗号方式	Encryption algorithms	WD 準備中
18033-2		-Part1: General	
		-Part2: Asymmetric ciphers	

8.2.4 IEEE について

関連 Web アドレス　http://grouper.ieee.org/groups/1363/

IEEE(Institute of Electrical and Electronics Engineers) は、ニューヨークに本部を持つ“米国電気電子学会”で、1963 年に創設され、会員は世界 130 カ国・32 万人以上を誇る学会である。IEEE は、アメリカの学会であるが、アメリカの国内標準を規定する ANSI(American National Standards Institute)から信任されたアメリカの国内標準化組織の 1 つでもあり、さまざまな国際的規格を定めている。

この IEEE において、IEEE P1363 (“Standard Specifications For Public Key Cryptography”)では、公開鍵暗号系の標準化を鍵共有、公開鍵暗号、デジタル署名という機能分類において、離散対数問題、楕円曲線上の離散対数問題、素因数分解問題ベースのスキームを対象にして行われている。

8.2.5 IETF について

関連 Web アドレス　http://www.ietf.org/

IETF (Internet Engineering Task Force)は、インターネット関連技術の標準化を推進している団体であり、メーリングリストや年 3 回のオフラインミーティングを中心に活動している。

8 つのエリアの一つである SecurityArea には、IPSec(IP Security Protocol WG)、PKIX(Public-Key Infrastructure (X.509) WG) TLS(Transport Layer Security WG)などの WG があり、暗号技術実装に関する議論が盛んに行われている。

S/MIME Mail Security WG のプリミティブとして、署名方式は DSA と RSA(PKCS#1v1.5)、ハッシュ関数は SHA-1、鍵管理は RSA(PKCS#1v1.5)、暗号方式は Triple DES がほぼ合意事項となっている。