

**暗号技術評価報告書 (2001 年度版)**  
**CRYPTREC Report 2001**

平成 14 年 3 月  
情報処理振興事業協会  
通信・放送機構



# 目次

	はじめに	1
	本報告書作成にあたって	3
	本報告書の利用にあたって	5
<b>第 1 章</b>	<b>暗号技術評価の概要</b>	<b>7</b>
1.1	暗号技術評価の体制とスケジュール	7
1.2	暗号技術評価の進め方	9
1.3	用語解説	10
1.4	評価委員会 委員名簿	11
<b>第 2 章</b>	<b>公開鍵暗号技術の評価</b>	<b>13</b>
2.1	評価対象と評価方法	13
2.1.1	評価対象暗号技術	13
2.1.2	評価方針	14
2.1.3	評価の実施方法	15
2.2	評価結果	17
2.2.1	評価結果の全体像	17
2.2.2	数論的問題の困難さに関する総評	18
2.2.3	詳細評価対象暗号技術の総評	19
2.2.4	監視状態の暗号技術の総評	21
2.2.5	2001 年度スクリーニング評価対象暗号の総評	22
2.3	数論的問題の困難さに関する評価	23
2.3.1	素因数分解問題	23
2.3.2	離散対数問題	26
2.3.3	楕円曲線上の離散対数問題	30
2.4	詳細評価対象暗号 (個別暗号) の評価	33
2.4.1	DSA	33
2.4.2	ECDSA	36
2.4.3	ESIGN 署名	40
2.4.4	RSA (RSA-OAEP, RSA-PSS, RSA 署名)	47
2.4.5	EPOC-2	56
2.5	監視状態の暗号の評価	60
2.5.1	ECIES in SEC1	60
2.5.2	ECDH in SEC1	62
2.5.3	DH	63
2.6	スクリーニング評価対象暗号の評価	65
2.6.1	OK-ECDSA	65
2.6.2	NTRU	67
2.6.3	HIME(R)	69
2.6.4	OK-ECDH	70

2.6.5	PSEC-KEM	71
<b>第 3 章</b>	<b>共通鍵暗号技術の評価</b>	<b>75</b>
3.1	評価方法	75
3.1.1	共通鍵暗号の評価方法	75
3.2	総評	80
3.2.1	64 ビットブロック暗号	80
3.2.2	安全性の総評	81
3.2.3	128 ビットブロック暗号	88
3.2.4	安全性の総評	90
3.2.5	ストリーム暗号	97
3.3	詳細評価対象暗号 (個別暗号) の評価	100
3.3.1	CIPHERUNICORN-E	100
3.3.2	Advanced Encryption Standard (AES)	106
3.3.3	CIPHERUNICORN-A	111
3.3.4	SEED	118
3.3.5	MULTI-S01	124
3.4	監視状態の暗号の評価	134
3.4.1	Hierocrypt-L1	134
3.4.2	MISTY1	139
3.4.3	Triple DES	144
3.4.4	Camellia	144
3.4.5	Hierocrypt-3	149
3.4.6	RC6	154
3.4.7	SC2000	158
3.5	スクリーニング評価対象暗号の評価	162
3.5.1	MUGI	162
<b>第 4 章</b>	<b>ハッシュ関数の評価</b>	<b>165</b>
4.1	評価方法	165
4.1.1	ハッシュ関数の評価方法	165
4.2	総評	165
4.2.1	ハッシュ関数の評価	165
4.3	詳細評価対象暗号 (個別暗号) の評価	166
4.3.1	draft SHA-256/384/512	166
4.4	監視状態の暗号の評価	175
4.4.1	RIPEMD-160	175
4.4.2	SHA-1	179
<b>第 5 章</b>	<b>擬似乱数生成系の評価</b>	<b>181</b>
5.1	評価方法	181
5.1.1	擬似乱数生成の評価方法	181
5.2	総評	181
5.2.1	擬似乱数生成系	181
5.3	監視状態の暗号の評価	182
5.3.1	PRNG based on SHA-1	182
5.4	スクリーニング評価対象暗号の評価	189
5.4.1	TAO TIME	189
<b>第 6 章</b>	<b>SSL プロトコルに関する暗号技術の評価</b>	<b>191</b>
6.1	総評	191

6.1.1	調査の目的	191
6.1.2	調査の対象と範囲	191
6.1.3	調査の方法	192
6.1.4	調査結果	192
6.1.5	SSL/TLS の運用と利用についての注意点	194
6.2	SSL/TLS プロトコルの実装と運用方法の評価	195
6.2.1	暗号方式に関わる安全性について	195
6.2.2	プロトコルメカニズムに関わる安全性について	195
6.2.3	実装に関わる安全性について	196
6.2.4	運用に関わる安全性について	196
6.2.5	SSL/TLS の比較調査	196
6.2.6	TLS の拡張作業	197
6.2.7	総括	197
6.3	SSL/TLS で利用されている暗号技術の評価	197
6.3.1	RSA(1024,2048) を用いた鍵共有法および署名法の脆弱性に関する調査	197
6.3.2	DES (40bit/56bit-key DES, 168bit-key Triple DES)	201
6.3.3	RC2(40,128)	206
6.3.4	RC4(40,128) および Arcfour(128)	207
<b>第 7 章</b>	<b>2002 年度評価予定暗号の問い合わせ先一覧</b>	<b>209</b>
7.1	監視状態の暗号	209
7.1.1	公開鍵暗号技術	209
7.1.2	共通鍵暗号技術	212
7.2	2002 年度詳細評価対象暗号候補	219
7.2.1	公開鍵暗号技術	219
7.2.2	共通鍵暗号技術	221
<b>第 8 章</b>	<b>評価暗号一覧</b>	<b>223</b>



# はじめに

本報告書は、暗号技術評価委員会の 2001 年度評価結果をまとめたものであります。同委員会は、2003 年度までにその基盤構築が予定されている我が国の電子政府で利用可能な暗号技術のリストアップを目的とした暗号技術評価プロジェクト (CRYPTREC プロジェクト) の一環として、これまで積極的な暗号技術評価活動を進めて参りました。

近年、申請届出手続きや政府調達など行政手続きの電子化を実現する電子政府の構築が進められておりますが、電子政府のサービスをより安心して利用できるようにするためには、暗号技術の利用が不可欠です。現在、様々な暗号技術が開発され、多くの暗号技術を組み込んだ製品・ソフトウェアが市場に提供されておりますが、電子政府において、このような暗号技術を利用していくためには、その安全性を判断する情報が極めて重要なものとなります。このような状況を背景に、2000 年度、経済産業省 (旧通商産業省) は、情報処理振興事業協会 (IPA) に電子政府において利用可能な暗号技術の評価の業務を委託し、CRYPTREC Report 2000(暗号技術評価報告書 2000 年度版) がまとめられました。(CRYPTREC Report 2000 は 2001 年 11 月 1 日 JIS-TR X0050 として発行)

2001 年度からは、IPA と通信・放送機構 (TAO) が暗号技術評価委員会の共同事務局となり、新設された暗号技術検討会 (総務省・経済産業省が事務局) との連携を図るとともに、本評価委員会オブザーバーに、警察庁、防衛庁、総務省、外務省、経済産業省、通信総合研究所に参加頂きました。具体的評価においては国内外の多数の専門家に評価を依頼するとともに、2002 年 1 月に開催した暗号技術評価ワークショップや Call for comment などを通じて広く意見を求めるなど、公平性、透明性の観点でもその充実を図りました。

暗号技術をめぐる世界の動きを見ると、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、欧州における暗号評価プロジェクト (NESSIE)、さらに ISO/IEC の暗号技術国際標準活動が活発に進められております。こうした状況にあって、本暗号技術評価プロジェクトは、我が国のみならず、世界の暗号技術の潮流にのった重要な活動であったと認識しております。そして、これらの動きの中で、本報告書により、我が国の電子政府構築における暗号技術に関して有益な情報を提供でき、関係者の皆様のお役に立つことができましたなら、これに勝る喜びはありません。

最後に、昨年度に引き続き暗号技術評価委員会の推進に委員長としてご尽力いただいた東京大学今井教授、顧問として大所高所から貴重な意見を頂いた中央大学辻井教授、共通鍵暗号評価小委員会委員長である東京理科大学金子教授、公開鍵暗号評価小委員会委員長である横浜国立大学松本教授をはじめ、多大なご尽力を頂きました各委員の皆様、関係省のオブザーバの皆様へ感謝するものであります。

2002 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 内藤 理  
通信・放送機構 研究企画管理部 部長 鈴木 薫





# 本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の 2001 年度の活動結果をまとめたものである。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、1999 年度 (平成 11 年度) には情報処理振興事業協会 (IPA) の「政府調達情報セキュリティ標準 (基準) に関する調査研究」と郵政省 (現総務省) 「暗号通信の普及・高度化に関する研究会 (委員長: 辻井重男中央大学教授)」との両者で、「暗号技術の評価」を実施すべきであるとの提言がなされた。この両者の報告における共通点は、情報セキュリティの基盤技術である暗号技術について、その信頼性等を技術的・専門的見地から客観的に検証する必要性が強調されていたことである。

この報告を踏まえ、2000 年 5 月に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、暗号技術の安全性等を評価することになった。

さらに 2001 年度は、暗号技術評価をより政府横断的な事業とすべく、総務省、経済産業省を事務局とした暗号技術評価検討会と通信・放送機構 (TAO) と IPA とを事務局とした暗号評価技術委員会が連携を取り、暗号技術評価を実施してきた。

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、暗号技術の評価することは、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の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、2002年度に予定されている電子政府推奨暗号の選定対象暗号に関し、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。本暗号技術評価は、我が国では初めての事業であるため、米国 AES の公募選定事業を参考にし、さらに 2000 年度の活動を踏まえ、2001 年 8 月の暗号技術公募、10 月の応募暗号説明会および 2002 年 1 月に暗号技術評価ワークショップを実施し、評価の公平性・透明性を考慮しつつ進めてきた。しかし、今回の暗号技術評価の成果は現時点で利用可能な技術による評価であり、全てが完了したという訳ではない。政府レベルの暗号技術検討会における検討結果を踏まえて、評価を継続する予定である。

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、暗号技術の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。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が、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。

2002 年 3 月

暗号技術評価委員会 委員長 今井 秀樹

# 本報告書の利用にあたって

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

本評価報告書の 1 章には、本暗号技術評価の概要を、2 章から 5 章には、各暗号技術の評価結果をまとめた。公開鍵暗号技術は 2 章、ブロック暗号やストリーム暗号の共通鍵暗号技術は 3 章、ハッシュ関数は 4 章、擬似乱数生成系は 5 章に記述した。また、SSL プロトコルに関する調査を 6 章に記述した。

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える。

今回の暗号技術評価は、2001 年 8 月から 9 月にかけて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ため、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または通信・放送機構までご連絡していただくと幸いです。

(問い合わせ先 e-mail: cryptrec-call@ipa.go.jp)



# 第1章

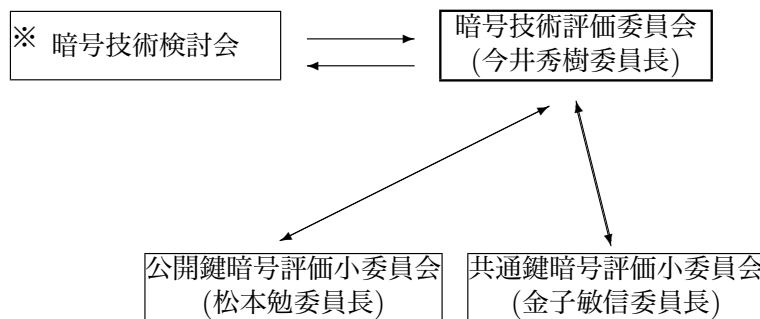
## 暗号技術評価の概要

### 1.1 暗号技術評価の体制とスケジュール

2003年度を目途として構築中の電子政府においてセキュリティの共通基盤となる暗号技術の評価を2000年度の活動に引き続き、2001年度(2001年4月から2002年3月)にも行った。電子政府におけるセキュリティ共通基盤の確保は重要な課題とされており、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。

2001年度は暗号技術評価体制をより強力にするため、総務省技術総括審議官と経済産業省商務情報局長が連名で主催する「暗号技術検討会」が設置された。この「暗号技術検討会」は政策的な判断を行ない、情報処理振興事業協会及び通信・放送機構が共同で設置した「暗号技術評価委員会」が技術的評価作業を実施することになった。(図1.1参照)従って、本報告書のみならず、「暗号技術検討会」の報告書も参照いただきたい。



※事務局: 総務省、経済産業省

図 1.1: CRYPTREC 体制

2001年度は、2000年度に引続き「暗号技術の公募」を実施した。公募対象としては、2000年度と同様、公開鍵暗号技術(守秘、署名、鍵共有、認証)と共通鍵暗号技術(64ビット)

トブロック暗号、128ビットブロック暗号、ストリーム暗号、ハッシュ関数、擬似乱数生成系)である。ただし、2000年度に応募いただき、詳細評価まで実施した暗号技術に関しても、応募時点以降の評価情報の追加や応募書類のエディトリアルな修正を施していただくため、再度応募を行っていただくこととした。更に、新規に応募いただく場合には、2000年度に詳細評価対象となった暗号技術と同等以上の特長を持つ暗号技術であることを技術的な条件とした。

2001年度の暗号技術の公募は、2001年8月1日から9月27日に実施し、その結果、31件の応募があった。2001年度の暗号技術の評価の実施にあたっては、2001年10月8日、9日に「応募暗号説明会」を開催し、すべての応募者から応募暗号技術に関する説明を行っていただいた。

2001年度に評価した暗号技術については、第8章の評価暗号一覧を参照していただきたい。

評価にあたっては、Call for Comment や「暗号技術評価ワークショップ」(2002年1月28日開催)を通して、評価検討状況の内容等について広く意見・コメントを求めた。

表 1.1: 暗号技術評価委員会の主な活動	
2000年5月	暗号技術評価委員会の設置
2000年6-7月	2000年度暗号技術の公募
2000年8-10月	2000年度暗号技術スクリーニング評価
2000年10月	暗号技術シンポジウム (CRYPTREC 活動の目的紹介)
2000年10-2001年3月	2000年度暗号技術の詳細評価
2001年3月	CRYPTREC Report 2000 の発行
2001年4月	暗号技術評価報告会 (2000年度 CRYPTREC 活動報告)
2001年8月-9月	2001年度暗号技術の公募
2001年10月	応募暗号説明会
2001年10-2002年3月	2001年度継続して応募された暗号技術 およびその他評価が必要と判断された暗号技術の詳細評価 2001年度新規に応募された暗号技術のスクリーニング評価
2002年1月	暗号技術評価ワークショップ (スクリーニング評価・詳細評価の状況報告)
2002年3月	CRYPTREC Report 2001 の発行

「暗号技術検討会」から評価依頼された暗号技術、「電子署名及び認証業務に関する法律施行規則」、及び「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」で取り上げられた暗号技術などを、暗号技術評価委員会で評価が必要と判断した暗号技術に加え、合計44個の暗号技術を評価の対象とした。(第8章参照)

上記「電子署名及び認証業務に関する法律施行規則」、及び「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」での規定は以下の通りである。

- (a) 電子署名及び認証業務に関する法律第2条第3項に基づいて、2001年4月1日に施行された電子署名及び認証業務に関する法律施行規則から抜粋:

**(特定認証業務)**

第二条 法第二条第三項の主務省令で定める基準は、電子署名の安全性が次のいずれかの有する困難性に基づくものであることとする。

- 一 ほぼ同じ大きさの二つの素数の積である千二十四ビット以上の整数の素因数分解
- 二 大きさ千二十四ビット以上の有限体の乗法群における離散対数の計算
- 三 楕円曲線上の点がなす大きさ百六十ビット以上の群における離散対数の計算
- 四 前三号に掲げるものに相当する困難性を有するものとして主務大臣が認めるもの

(b) 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から抜粋:

**(特定認証業務に係る電子署名の基準)**

第三条 規則第二条の基準を満たす電子署名の方式は、次の各号のいずれかとする。

- 一 RSA方式 (オブジェクト識別子 一 二 八四〇 一一三五四九 一 一 五又は 一 二 八四〇 一一三五四九 一 一 四) であって、モジュラスとなる合成数が千二十四ビット以上のもの
- 二 ECDSA方式 (オブジェクト識別子 一 二 八四〇 一〇〇四五 四 一) であって、楕円曲線の定義体及び位数が百六十ビット以上のもの
- 三 DSA方式 (オブジェクト識別子 一 二 八四〇 一〇〇四〇 四 三) であって、モジュラスとなる素数が千二十四ビットのもの
- 四 ESIGN方式 (オブジェクト識別子 〇 二 四四〇 五 五 三 四又は 〇 二 四四〇 五 五 三 三) であって、モジュラスとなる合成数が千二十四ビット以上、検証に利用されるべき指数が八以上のもの

## 1.2 暗号技術評価の進め方

2001年度の暗号技術評価は、「暗号技術評価委員会」で評価方針を定め、具体的な評価作業を「共通鍵暗号評価小委員会」および「公開鍵暗号評価小委員会」で行った。共通鍵暗号技術、公開鍵暗号技術の性質上評価作業の進め方は、若干違いが見られる。共通の方針は、詳細評価に関しては、内外の主要な研究者に評価を依頼し、その評価結果をもとに「共通鍵暗号評価小委員会」、「公開鍵暗号評価小委員会」それぞれで「委員会」としての結論をとりまとめた。また、スクリーニング評価は、国内の主要な研究者に評価を依頼し、その評価結果をもとに「共通鍵暗号評価小委員会」、「公開鍵暗号評価小委員会」それぞれで「委員会」としての結論をとりまとめた。

(a) 共通鍵暗号評価小委員会での評価

2001年度に「共通鍵暗号評価小委員会」で実施した暗号技術評価は、

- (1) 2000年度の詳細評価実施暗号の詳細評価
- (2) 2001年度に新規に提案された暗号のスクリーニング評価
- (3) 暗号技術評価委員会で評価が必要と判断した暗号技術の評価に大別できる。

2000年度に詳細評価対象となった暗号技術に対する詳細評価は、主としては2000年度の詳細評価においては、共通鍵暗号評価小委員会としての結論をまとめきれなかった暗号技術を中心に評価を行った。また、暗号技術検討会からの依頼に基づき、韓国がISO/IEC JTC1 SC27に提案したSEED及びInternetにおける標準的なセキュ

アプロトコルである SSL/TLS で用いられている、RC2、RC4、DES、Triple DES の評価を実施した。これら暗号技術検討会からの依頼で評価した暗号の内、SEED については国際貢献/国際協力という観点から評価を依頼された。

(b) 公開鍵暗号評価小委員会での評価

2001 年度に「公開鍵暗号評価小委員会」で実施した暗号技術評価は、

- (1) 2000 年度の詳細評価実施暗号の詳細評価
- (2) 2001 年度に新規に提案された暗号のスクリーニング評価
- (3) 暗号技術評価委員会で評価が必要と判断した暗号技術の評価

に大別できる。2000 年度の詳細評価対象となった暗号、「電子署名及び認証業務に関する法律に基づく特定認証業務の認定に係る指針」に記載された暗号技術、及び、SSL/TLS で用いられている暗号技術 (RSA 暗号) に対する詳細評価に関しては、SSL/TLS での使われ方に着目して評価を実施した。

また、2001 年度にはプリミティブの安全性評価に資する目的で、素因数分解問題に関する調査を開始した。素因数分解に関する調査は、主に計算機実験による具体的な計算量の評価を行ない、現在困難とされている合成数の素因数分解がどの程度難しいかの見積りの精度を高めることを目的としている。

さらに、SSL/TLS に関しては電子政府システムにおいても、ユーザーと電子政府システムで使用される主要な暗号プロトコルと想定されるため、暗号プロトコルとしての現状明らかになっている問題点等を調査した。

### 1.3 用語解説

この節では、本報告書で用いられる暗号技術評価委員会特有の用語について、定義と若干の解説を加える。

(1) 応募暗号

暗号技術評価委員会が行った暗号技術の公募に対して、応募された暗号技術。

(2) その他評価が必要な暗号技術

暗号技術評価委員会が公募の有無にかかわらず評価が必要な暗号と判断した暗号技術。

(3) スクリーニング評価

応募暗号技術に対して、安全性に明らかな問題がないか、第三者実装上問題がないかの一次的評価。

(4) 詳細評価

電子政府で利用可能かどうかの観点から、既知の攻撃法に対する耐性、パラメータや鍵の設定基準および実装性に関する評価。

(5) 継続評価

暗号技術評価委員会においてさらに評価の継続が必要と判断された暗号技術の評価。

(6) 特定評価

電子署名法等で利用される暗号技術の評価など、暗号技術の要件が整理された特定分野の暗号技術の評価

(7) 監視状態の暗号

詳細評価の結果、安全性について今のところ特に問題がないと判断され、新たな技術的な進展による脅威を監視する状態にある暗号技術。その動向により再評価を実施することがある。



## 1.4 評価委員会 委員名簿

### 暗号技術評価委員会 (肩書等は 2002 年 3 月現在)

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

### 公開鍵暗号評価小委員会 (肩書等は 2002 年 3 月現在)

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

### 共通鍵暗号評価小委員会 (肩書等は 2002 年 3 月現在)

委員長	金子 敏信	東京理科大学 教授
委員	荒木 純道	東京工業大学 大学院 教授
委員	川村 信一	株式会社東芝 主任研究員
委員	神田 雅透	日本電信電話株式会社 研究主任
委員	香田 徹	九州大学 大学院 教授
委員	古原 和邦	東京大学 助手
委員	櫻井 幸一	九州大学 大学院 助教授
委員	下山 武司	株式会社富士通研究所 研究員
委員	宝木 和夫	株式会社日立製作所 部長
委員	館林 誠	松下電器産業株式会社 主席研究員
委員	角尾 幸保	日本電気株式会社 主任研究員
委員	時田 俊雄	三菱電機株式会社 主席研究員
委員	森井 昌克	徳島大学 教授

## オブザーバー (肩書等は 2002 年 3 月現在)

赤岩 司	警察庁 情報通信局
鳥居 秀行	警察庁 情報通信局
榊賀 政浩	防衛庁 運用局
松井 教安	防衛庁 陸上幕僚監部
喜安 拓	総務省 情報通信政策局
丹代 武	総務省 情報通信政策局 (2001 年 6 月まで)
門馬 弘	総務省 情報通信政策局
今井 清春	総務省 情報通信政策局 (2001 年 6 月まで)
百々 浩樹	総務省 情報通信政策局 (2001 年 7 月まで)
福岡 晃	総務省 情報通信政策局
浦谷 真人	総務省 行政管理局
山本 寛繁	総務省 行政管理局
奥村 定夫	外務省 外務省大臣官房
東井 芳隆	経済産業省 商務情報政策局 (2001 年 5 月まで)
大野 秀敏	経済産業省 商務情報政策局
山本 文土	経済産業省 商務情報政策局
田辺 雄史	経済産業省 商務情報政策局
町田 昇	経済産業省 商務情報政策局
木戸 達雄	経済産業省 産業技術環境局
勝亦 真人	経済産業省 産業技術環境局
深沢 太郎	経済産業省 産業技術環境局
滝澤 修	通信総合研究所 情報通信部門

## 事務局

情報処理振興事業協会セキュリティセンター

小林正彦 (2001 年 6 月まで)、内藤理、戸叶秀晴、網島和博、黒川貴司、  
四方順司、武田仁己、竹谷清康、田中公明、田中秀磨、矢田健一、山岸篤弘、  
黒川恭一、中澤昭彦

通信・放送機構

鈴木薫、山田和晴、青木和麻呂、天野滋、笠井祥、中嶋香代子、山村明弘

## 第2章

# 公開鍵暗号技術の評価

## 2.1 評価対象と評価方法

### 2.1.1 評価対象暗号技術

2001年度評価対象となった暗号技術は、

1. 応募暗号技術 (新規応募および継続評価対象)  
ESIGN(電子署名法に係る指針), ECDSA in SEC1, EPOC-2, RSA-OAEP, RSA-PSS,  
HIME(R), OK-ECDSA, NTRU, OK-ECDH, PSEC-KEM
2. 電子署名法に係る指針<sup>\*1</sup>に記載された暗号技術  
DSA, ECDSA(ANSI X9.62), ESIGN(電子署名法に係る指針),  
RSA 署名 (PKCS#1 v1.5<sup>\*2</sup>)
3. 監視状態にあった暗号技術  
ECIES in SEC1, DH, ECDH in SEC1

の3つに分類される。

2001年度は、暗号技術公募へ新規に応募された暗号技術に対してスクリーニング評価を実施し、また、2000年度詳細評価対象となった暗号技術のうち、詳細評価結果に基づき暗号技術評価委員会が継続して評価を必要とした暗号技術であって、応募者の継続的な評価の意思および調達可能な状況が確認できた暗号技術に対して詳細評価を行った。ここで、前者に対しては、2000年度詳細評価対象となった暗号技術と同等以上の特長（安全性・実装性等）を持つ技術を期待し、後者に対しては、2000年度の詳細評価結果のうち、更なる検討が必要とされた評価項目について重点的に評価を実施した。

---

<sup>\*1</sup> 第1.1節 (a),(b)を参照のこと。

<sup>\*2</sup> 本方式は、規格書 RSA PKCS#1v1.5で規定され、規格書 RSA PKCS#1v2.0以降にも引き継がれているため、本報告書では、方式名として RSA-PKCS#1v1.5と略記する。

### 2.1.2 評価方針

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

本年度の評価は、次の方針によって行った。

1. 比較的長い期間にわたる使用実績・評価実績があり、インターオペラビリティの観点から仕様の変更を簡単には求められない公開鍵暗号技術については、必ずしも証明可能安全性が示されていることを要件とはしない。
2. 仕様実績のあまりない新しい公開鍵暗号技術については、既存暗号技術とは独立に仕様を定めることができることから、証明可能安全性がきちんと示されていることを必須とする。
3. 2001年9月末の2001年度応募締切時点までは、2000年度応募暗号については2000年度の仕様からの変更は同一暗号として認める場合もあるが、2001年度の評価は応募時点での暗号技術を対象として行う。

なお、ここで参照する証明可能安全性とは、暗号が安全であることが証明されているということを示すものではないことに注意が必要である。本章では、「ある仮定の下での証明可能安全性を有する」という表現を用いて次の状況を示す。すなわち、ある公開鍵暗号が証明可能安全性を有するとは、その暗号またはその暗号の理想化暗号に対して、その暗号で守りたい安全性を脅かす攻撃方法があれば、それを使って、別の数学的問題を低い計算量で解く方法が導けることを、何らかの前提のもとで、厳密に証明できることを指すことにする。ただし、ある暗号の理想化暗号とは、その暗号スキームが用いる補助関数（ハッシュ関数など）を仮想的なもの（ランダム関数など）に置き換えた以外はそ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。

### 2.1.3 評価の実施方法

評価作業は評価対象別にスクリーニング評価、詳細評価および関連調査の3つに分かれる。公開鍵暗号評価小委員会では国内外の研究者に評価を委託し、その評価結果を検討し、まとめた。

#### 2.1.3.1 スクリーニング評価

スクリーニング評価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どうかを判断するため実施された。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や自己完結性）が整っていることの確認
- 書面上で容易に判明するような欠点（解読手法等）の検査
- 応募時点で提出された暗号技術仕様書、自己評価書の内容の点検と妥当性の確認

また、2000年度詳細評価対象となった暗号技術と同等以上の特長（安全性・実装性等）を持つ技術を期待した点が2001年度の特徴である。PSEC-KEMに関しては、仕様変更が大きかったため、新規応募扱いとなった。スクリーニング評価においては、国内外の暗号研究者に委託して評価を行った（表2.1を参照）。

表 2.1: スクリーニング評価に関する外部評価依頼数

評価方法	評価対象	海外評価数	国内評価数	合計
スクリーニング評価	OK-ECDSA	-	3	3
	NTRU	-	3	3
	HIME(R)	-	3	3
	OK-ECDH	-	3	3
	PSEC-KEM	1	2	3

#### 2.1.3.2 詳細評価

2001年度は安全性評価だけを実施した。2000年度に既に詳細評価済みのものに関して更なる検討が必要とされた項目を重点的に評価したのが2001年度の特徴である。安全性評価においては、国内外の暗号研究者に委託して評価を行った（表2.2を参照）。実装評価は、詳細評価対象暗号の多くが2000年度に計測済みかあるいは多くの使用実績を有しているものと考えられるため実施しなかった。

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

- 数論的問題の困難さに関する安全性評価項目
  - i) 素因数分解問題

- 既知の解法アルゴリズムの調査とそれらの効率の比較
- $pq$  型と  $p^d q$  型 ( $d \geq 2$ ) の比較
- ii) 離散対数問題
  - 既知の解法アルゴリズムの調査とそれらの効率の比較
- iii) 楕円曲線上の離散対数問題
  - 既知の解法アルゴリズムの調査とそれらの効率の比較
  - 限定された曲線 (Koblitz 曲線等) の場合の問題点の調査
- 暗号スキームに関する安全性評価項目
  - i) DSA
    - プリミティブ及びスキームの安全性評価
    - FIPS186-2 Appendix 3 で与えられている乱数生成法
  - ii) ECDSA
    - generic group model における存在的偽造不可の証明可能安全性
    - Koblitz 曲線の安全性評価
  - iii) ESIGN
    - 推奨パラメータのサイズの妥当性
    - $e$  乗根近似問題
  - iv) RSA
    - RSA 署名の安全性評価
    - RSA-PSS, RSA-OAEP の証明可能安全性
    - RSA-OAEP に対する Manger の攻撃法
  - v) EPOC-2
    - コンバージョンの評価
    - 推奨パラメータの選択の妥当性

表 2.2: 詳細評価に関する外部評価依頼数

評価方法	評価対象	海外評価数	国内評価数	合計
詳細評価 (数論的問題 の困難さ)	素因数分解問題 (実験)	-	1	1
	素因数分解問題 (調査)	-	1	1
	特殊な形の素因数分解問題	3	1	4
	離散対数問題	2	1	3
	楕円曲線離散対数問題	2	-	2
詳細評価 (スキーム)	EPOC-2	2	2	4
	RSA-OAEP, RSA-PSS 等	2	2	4
	ESIGN	3	1	4
	DSA	3	2	5
	ECDSA	3	1	4

### 2.1.3.3 関連調査

電子政府等の政府利用における暗号技術に対する要求条件を明確にするため、暗号技術検討会から SSL に関する調査依頼があった。公開鍵暗号評価小委員会では、RSA(1024, 2048bit) の評価と SSL/TLS プロトコルに関する調査を実施した。以下の点に配慮して、国内の研究者に委託して評価を行った (表 2.3 を参照)。

- RSA 暗号を用いた署名法や鍵共有法について、プロトコルの仕様や動作を利用した攻撃法に対する安全性

- 既知のセキュリティホール（実装によるものも含む）とその対策法や運用上の注意点
- SSL 3.0 と TLS 1.0 の相違点や TLS1.0 の改訂プロジェクトの内容

なお、評価結果については第 6 章を参照のこと。

表 2.3: SSL に関する外部調査依頼数

評価方法	評価対象	海外評価数	国内評価数	合計
関連調査 (SSL/TLS)	RSA に係る脆弱性	-	1	1
	プロトコル調査	-	2	2

## 2.2 評価結果

### 2.2.1 評価結果の全体像

2001 年度に評価を行った公開鍵暗号技術は、機能と関連する数論的問題とから表 2.4 のように分類できる。

表 2.4: 公開鍵暗号技術の評価結果

	素因数分解問題	(楕円曲線) 離散対数問題	格子問題
署名	ESIGN <sup>(2)</sup> RSA 署名 <sup>(1)</sup> RSA-PSS <sup>(1)</sup>	DSA <sup>(1)</sup> ECDSA <sup>(1)</sup> ECDSA in SEC1 <sup>(1)</sup> OK-ECDSA <sup>(7)</sup>	
守秘	EPOC-2 <sup>(4)</sup> HIME(R) <sup>(5)</sup> RSA-OAEP <sup>(1)</sup>	ECIES in SEC1 <sup>(3)</sup>	NTRU <sup>(7)</sup>
鍵共有		DH <sup>(1)</sup> ECDH in SEC1 <sup>(1)</sup> OK-ECDH <sup>(7)</sup> PSEC-KEM <sup>(6)</sup>	

第 2.1.2 節の方針に従って評価を行い、以下の結論 (1), ..., (7) を得た。表 2.4 中には暗号名にその暗号に対する結論を付記した。

- (1) 特定評価対象または継続評価対象の RSA 署名 (RSA-PKCS #1 v.1.5)、RSA-PSS、RSA-OAEP、DSA、ECDSA(ANSI X9.62)、ECDSA in SEC1、DH、ECDH in SEC1 は、電子政府での使用には問題がないと考えられる。もちろん、使用に際しては適切なパラメータ群を選択することが必要である。
- (2) 特定評価対象で、電子署名法に係る指針に記載されている署名方式である ESIGN はその指針に記載されている安全性パラメータの範囲に、無視できない確率で署名の偽造に成功するパラメータが含まれていることが判明した。よって ESIGN は詳細な評価なしに電子政府での使用は薦められない。
- (3) 監視状態にあった暗号の 1 つである ECIES in SEC1 は、安全性についての議論が新たに生じてきたので、詳細な評価なしに電子政府での使用は薦められない。

- (4) 継続評価対象暗号の EPOC-2(CRYPTREC2001 評価対象の仕様のもの) は新しい暗号であるが、自己評価書で与えられた証明可能安全性の議論に不備があることが詳細な評価の結果判明した。ゆえに、この EPOC-2 の電子政府での使用は薦められない。
- (5) スクリーニング評価対象暗号の HIME(R) は、自己評価書における証明可能安全性の証明に疑問があり、HIME(R) の電子政府での使用についての判断は、詳細な評価なしには行えない。
- (6) スクリーニング評価対象暗号の PSEC-KEM は、鍵カプセル化メカニズムとしての証明可能安全性を有するとされる。しかし、鍵カプセル化メカニズムは比較的新しい技術であるので、PSEC-KEM の電子政府での使用についての判断にはさらなる検討が必要である。
- (7) スクリーニング評価対象暗号の NTRU は新しい暗号であるが、自己評価書において証明可能安全性の証明が与えられていない。スクリーニング評価対象暗号の OK-ECDSA は新しい暗号であるが、証明可能安全性について ECDSA と同じである。スクリーニング評価対象暗号の OK-ECDH は新しい暗号であるが、証明可能安全性について ECDH と同じである。また、OK-ECDSA、OK-ECDH のサイドチャネル攻撃に対する耐性は自己評価書に記載されている内容だけでは十分に確認できない。よって、これら3つの暗号の電子政府での使用は薦められない。

## 2.2.2 数論的問題の困難さに関する総評

### 2.2.2.1 素因数分解問題

合成数  $n$  の素因数分解問題に関しては、2001年時点で、 $n = pq$  については  $|p| = |q|$  かつ  $|n| \geq 1024$  で、 $n = p^2q$  については  $|p| = |q|$  かつ  $|n| \geq 1024$  で、それぞれ安全と考えられる。素因数分解問題の安全性についての今後の見通しについては、2.3.1.3 節に詳述してある。

### 2.2.2.2 離散対数問題

素体  $\mathbb{F}_p$  の部分群 (位数  $q$ ) の離散対数問題に関しては、2001年時点で、 $|p| \geq 1024$  かつ  $|q| \geq 160$  で安全と考えられる。離散対数問題の安全性についての今後の見通しについては、2.3.2.3 節に詳述してある。

### 2.2.2.3 楕円曲線上の離散対数問題

楕円曲線上の離散対数問題に関しては、2001年時点で、例外的な楕円曲線を除けば、群位数 (より正確には、ベースポイントの位数) が160ビット以上の素因子をもてば安全と考えられる。楕円曲線上の離散対数問題の安全性についての今後の見通しについては、2.3.3.4 節に詳述してある。



### 2.2.3 詳細評価対象暗号技術の総評

#### 2.2.3.1 DSA (署名)

有限体上の離散対数問題の困難性に依存している。証明可能安全性は示されていない。DSA は米国 NIST(National Institute of Standards and Technology) によって提案、標準化された電子署名方式であり、電子署名法に係る指針に記載されている。FIPS 186-2 Appendix 3 の擬似乱数生成について疑問が提示されており、2001 年 10 月に NIST が FIPS 186-2 の Change Notice において提示した擬似乱数生成の手順の修正に従うことが望ましい。仕様上はパラメータ  $p$  のサイズを選択可能であるが、安全性の観点から 1024 ビットを選択することを強く推奨する。より大きなサイズのパラメータを選択可能とするために NIST では仕様の変更を検討していることに注意されたい。

#### 2.2.3.2 ECDSA (署名)

■**ECDSA (ANSI X9.62)** 署名のための公開鍵方式であり、その安全性は楕円曲線上の離散対数問題の困難性に依存している。特殊なモデルでの証明可能安全性は示されているが、そのモデルの妥当性に関しては決着がついていない。2001 年時点では安全性に大きな脅威を与えるような問題点は指摘されていない。電子署名法に係る指針にはパラメータの値が 160 ビット以上の ECDSA が記載されている。擬似乱数生成器に関して FIPS186-2 に記載された手法が規定されているが、ECDSA の原型である DSA において問題点が指摘されており、NIST が FIPS186-2 change notice に提示した擬似乱数生成器の動向に注意すべきである。

■**ECDSA in SEC1** 署名のための公開鍵方式であり、その安全性は楕円曲線上の離散対数問題の困難性に依存している。特殊なモデルでの証明可能安全性は示されているが、そのモデルの妥当性に関しては決着がついていない。2001 年時点では安全性に大きな脅威を与えるような問題点は指摘されていない。SEC2 に使用の推奨される具体的な楕円曲線が示されている。これらの楕円曲線については特段の問題点は指摘されていない。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（またはアノマラスバイナリ曲線）とよばれる曲線は、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある。

#### 2.2.3.3 ESIGN (署名)

プリミティブは  $n = p^2q$  型素因数分解問題と  $e$  乗根近似問題を安全性の根拠としている。ESIGN の署名生成速度は RSA 署名と比べて高速である。ESIGN 署名の仕様は複数存在する。その中で電子署名法に係る指針に記載された署名方式の 1 つである ESIGN (電子署名法に係る指針) の評価結果は次の通りである。

■**ESIGN (電子署名法に係る指針)** 証明可能安全性は示されていない。電子署名法に係る指針で指定された安全性パラメータの一部 (例えば SHA-1 を用いた場合、 $|n| = 2048$  かつ  $e = 8$  など) は、無視できない確率で署名の偽造に成功する。また、ハッシュ関数と

して指定された MD5 と SHA-1 のうち、MD5 の使用は推奨できない。電子署名法に係る指針の改定を検討すべきである。

#### 2.2.3.4 RSA (署名)

プリミティブの安全性は、 $n = pq$  型素因数分解問題の困難性に依存している。RSA に関しては長期間広く使われている実績、広範な観点からの安全性評価が行われてきていて経験的に安全であると考えられる。RSA プリミティブを利用した署名には多くの仕様が存在する。我々は RSA-PKCS #1 v1.5 と RSA-PSS (IEEE P1363a 版) を評価した。

■**RSA-PKCS #1 v1.5** 証明可能安全性は有しない。RSA-PKCS #1 v1.5 は電子署名法に係る指針に記載されている署名方式の 1 つである。安全性に対する疑問点の指摘は現在までに報告されていない。ただし、多くの署名法のエンコーディング手法について署名の偽造が提案されていることにより、本方式で採用されているエンコーディング手法の安全性について検討を継続する必要がある。電子署名法に係る指針では、MD5 と SHA-1 がハッシュ関数として指定されているが、CRYPTREC Report 2000 で指摘されているように、MD5 の使用は推奨できない。一方で、同程度の効率を実現でき、かつ安全性が示されている RSA-PSS の優位性を指摘する意見もある。

■**RSA-PSS (IEEE P1363a 版)** ランダムオラクルモデルのもとで証明可能安全性を有する。安全性が証明されているほかの署名法 (全域ハッシュ法等) に比べて緊密な帰着関係を証明できる特徴を有している。CRYPTREC への提案方式と論文で証明が与えられている方式に若干の相違があるため、対応するパラメータの関係を把握して設計パラメータを選択する必要がある。RSA-PSS を電子署名法に係る指針で指定することを検討することが必要である。

#### 2.2.3.5 RSA (守秘)

プリミティブの安全性は、 $n = pq$  型素因数分解問題の困難性に依存している。RSA に関しては長期間広く使われている実績、広範な観点からの安全性評価が行われてきていて経験的に安全であると考えられる。

■**RSA-OAEP** ランダムオラクルモデルのもとで証明可能安全性を有する。2000 年 Shoup の指摘を皮きりにして、RSA-OAEP の安全性が議論されたが、安全性の帰着の効率は低下するものの安全であることは示されている。CRYPTREC への提案方式と論文で証明が与えられている方式に若干の相違があるため、対応するパラメータの関係を把握して設計パラメータを選択する必要がある。

#### 2.2.3.6 EPOC-2 (守秘)

2001 年度の CRYPTREC に応募されている EPOC-2 は 2000 年度の CRYPTREC に応募された EPOC-2 においてエンコーディング手法を厳密に規定した仕様をもつ。安全性は、 $n = p^2q$  型の素因数分解問題の困難性に依存している。自己評価書で与えられた証明可能安全性の議論に不備があることが詳細な検討の結果判明した。 $n = p^2q$  型の素因

数分解問題の困難性は  $n = pq$  型の素因数分解問題の困難性と違いがあることに注意したい。共通鍵暗号でブロック暗号を不適切に利用することが安全性を損なうことにつながることもあるので、使用に際してはブロック暗号の利用モード (modes of operation) に注意すべきである。

## 2.2.4 監視状態の暗号技術の総評

### 2.2.4.1 ECIES in SEC1 (守秘)

本暗号技術は、2000 年度には ECAES in SEC1 として CRYPTREC に応募されていたが、2001 年度では暗号技術名を ECIES in SEC1 に変更して応募されている。ECIES in SEC1 の安全性は楕円曲線上の離散対数問題の困難性に依存している。最近、安全性について新たに議論が生じてきたので、このことに関しての今後の継続的な安全性評価が必要である。

### 2.2.4.2 DH (鍵共有)

安全性は離散対数問題の困難性に依存している。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、個々のプロトコ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 11770-3、Oakley、PGP)。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) に対しては問題点は指摘されていないが、能動的攻撃 (鍵共有のために通信されるデータに攻撃者が影響を与える可能性がある場合) に対して、最低限以下の 3 点に注意を払う必要がある。

- 公開鍵と Entity との結びつきを保証する手段を確保する。
- (更新を前提とする) セッション鍵共有方式として使用する場合は、交換する公開鍵は一時的なものとする。
- 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。

### 2.2.4.3 ECDH in SEC1 (鍵共有)

本暗号技術は、2000 年度には ECDHS in SEC1 として CRYPTREC に応募されていたが、2001 年度では暗号技術名を ECDH in SEC1 に変更して応募されている。ECDH in SEC1 の安全性は楕円曲線上の離散対数問題の困難性に依存している。現時点では、受動的攻撃に対して、大きな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である。

- 公開鍵と Entity との結びつきを保証する手段を確保する。
- (更新を前提とする) セッション鍵共有方式として使用する場合は、交換する公開鍵は一時的なものとする。

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 曲線とよばれる曲線は、限定されたクラスの楕円曲線であるため、そ

のクラス特有の攻撃法が出現する可能性に注意を払う必要がある。

## 2.2.5 2001年度スクリーニング評価対象暗号の総評

### 2.2.5.1 OK-ECDSA (署名)

署名のための公開鍵方式であり、モンゴメリ型楕円曲線を利用することに特徴がある。その安全性はモンゴメリ型楕円曲線上の離散対数問題の困難性に依存している。特殊なモデルでの証明可能安全性は示されているが、そのモデルの妥当性に関しては決着がついていない。サイドチャネル攻撃に対する耐性は自己評価書に記載されている内容だけでは十分確認できない。

### 2.2.5.2 NTRU (守秘)

安全性の根拠は格子における最短ベクトル問題の困難性に依存している。既存の公開鍵暗号に対して処理性能の高速性が特徴である。応募者は、

- (1) メッセージのランダムパディングにより IND-CPA になる、
- (2) IND-CPA に藤崎-岡本変換を施し、IND-CCA2 を達成できる、

と主張している。しかし、(1) の真偽が確認されていないため、現在では証明可能安全性は示されていない。また、安全性の根拠となる問題の特殊性も懸念される。

### 2.2.5.3 HIME(R) (守秘)

守秘のための公開鍵方式であり、その安全性は特殊な形の素因数分解問題の困難性に依存している。自己評価書における証明可能安全性の主張には疑問があり、詳細な評価なしに電子政府での使用は薦められない。処理が RSA-OAEP より高速である可能性がある。

### 2.2.5.4 OK-ECDH (鍵共有)

鍵共有のための公開鍵方式であり、モンゴメリ型楕円曲線を利用することに特徴がある。その安全性はモンゴメリ型楕円曲線上の離散対数問題の困難性に依存している。受動的攻撃に対する問題点は指摘されていないものの、能動的攻撃に対して脆弱なことが指摘されている。サイドチャネル攻撃に対する耐性は自己評価書に記載されている内容だけでは十分確認できない。

### 2.2.5.5 PSEC-KEM (鍵共有)

鍵カプセル化のための公開鍵方式であり、その安全性は楕円曲線上の離散対数問題の困難性に依存している。鍵カプセル化メカニズムとしての証明可能安全性を有するとされる鍵カプセル化メカニズムの電子政府のための暗号技術における位置付けが明確にされていない現状においては、PSEC-KEM の電子政府での使用についての判断にはさらなる検討

が必要である。

## 2.3 数論的問題の困難さに関する評価

### 2.3.1 素因数分解問題

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

#### 2.3.1.1 有力なアルゴリズム

暗号プリミティブの評価を主眼として素因数分解問題を考えるため、ここでは分解の対象とする合成数を  $n = p^r q$  ( $p, q$ : 素数,  $p < q, r \geq 1$ ) とする。 $r = 1$  のときは RSA 等で使われる合成数、 $r \geq 2$  のときは ESIGN 等で使われる合成数となる。

素因数分解問題を解くアルゴリズムはいくつか知られているが、それらは実行時間の関数を決める主要因に応じて 2 種類に大別される。ひとつは、 $n$  に含まれる最小素因数のサイズ  $|p|$  に依存して実行時間が決まるものであり、他方は  $n$  のサイズ  $|n|$  だけに依存して実行時間が決まるものである。前者に属するアルゴリズムのうち最高速のものは楕円曲線法 (ECM)[5] であり、後者に属するもので最高速は一般数体ふるい法 (GNFS)[6] である。どちらも準指数関数時間を要する。具体的には以下の表となる。

		素因数分解アルゴリズム	
		楕円曲線法 (ECM)	一般数体ふるい法 (GNFS)
入力	$n$ ( $n = p^r q, p < q, r \geq 1$ )		
実行時間	$O(\exp(c\sqrt{\log_e p \log_e \log_e p}) \cdot (\log_2 n)^2)$ (主に $ p $ の関数, $c = 1.414$ )	$O(\exp(c(\log_e n)^{1/3}(\log_e \log_e n)^{2/3}))$ ( $ n $ の関数, $c = 1.901$ )	
現実的な 実行条件	$ p  \leq \theta_e$ (2001 年現在 $\theta_e = 183$ )	$ n  \leq \theta_g$ (2001 年現在 $\theta_g = 512$ )	

上記の表で「現実的な実行条件」とは、現在のそのアルゴリズムと計算資源をもってすれば現実的な時間内に分解が実現できる範囲のことであり、現実的という曖昧な言葉で定

義されているうえに、利用可能な計算資源にも制限がないことから、当然のことながら  $\theta_e$  や  $\theta_g$  は規格化されていない不確定な値である。しかしながら、実際に分解に成功した例を目安にすることはそれほど不自然ではなく、その意味で 2001 年現在では  $\theta_e = 183$  (文献 [8])、 $\theta_g = 512$  (文献 [3]) と考えることができる\*<sup>3</sup>。  $\theta_e$  と  $\theta_g$  の将来予想については後述する。

なお ECM も GNFS も、アルゴリズムの基本方針を踏襲しながら実行時間を改良するための研究が常になされおり (例えば ECM については [9]、GNFS については [4])、その観点からは、両アルゴリズムとも 2001 年現在のものであることに注意を要する。

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

### 2.3.1.2 安全な合成数のサイズ

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

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

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

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

しかしながら、 $\theta_e$  と  $\theta_g$  はもともと規格化された量ではなく、また仮に規格化されたとしても、 $|p|$  や  $|n|$  を不必要に大きくとればスキームのパフォーマンス (暗号化・復号・署名生成・署名検証等の計算時間) に望ましくない副作用をもたらす。そこで、2001 年段階で目安となる  $\theta_e$  や  $\theta_g$  の値を勘案して、実際的な意味で  $|p|$  や  $|n|$  の安全な範囲を評価する必要がある。

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

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

ただし、細かな点でいくつか重要な補足意見があった。具体的にはマージンの問題であ

\*<sup>3</sup> 2002 年 1 月 21 日付のネットワーク上のアナウンスメントによれば、 $|n| = 524$  (10 進 158 桁) の  $n = pq$  型の合成数が GNFS により分解された。F. Bahr, J. Franke, T. Kleinjung, "Factorization of 158-digit cofactor of  $2^{953} + 1$ ," <http://www.cryptoworld.com/announcements/c158.txt>

る。直観的には、アルゴリズムの現実的な実行条件を決めるサイズ ( $\theta_e$  や  $\theta_g$ ) と実際のサイズ ( $|p|$  や  $|n|$ ) の差がマージンに関係していると考えてよい。マージンを考える趣旨は、それが小さいと、アルゴリズムや利用可能な計算資源が仮に劇的に改良されたときには  $\theta_e$  や  $\theta_g$  が上昇し、そのままの  $p$  や  $n$  では分解されるおそれがあるからである。問題にされたのは ECM による攻撃に対するマージン (対 ECM マージン) である。例えば、 $n = pq$  と  $n = p^2q$  の場合、どちらも  $|n| = 1024$  ならば最小素因数サイズ  $|p|$  に違いが現れ、明らかに  $n = p^2q$  の場合のほうがマージンが小さい。もし、 $n = pq$  型で  $|n| = 1024$  の場合と同様の対 ECM マージンを確保するならば、 $n = p^2q$  型の合成数は  $|n| = 1280$  とすべきという見解が評価者の 1 人から提出されている (同じ評価者は  $n = p^3q$  で  $|n| = 1024$  の場合は安全ではないとも主張している)。また別の評価者の 1 人は、 $n = pq$  で  $|n| = 1024$  と対等の対 ECM マージンを確保するために、 $n = p^2q$  の  $|n|$  としてさらに大きな値を主張している。マージンの評価は、より安全な合成数サイズを評価するうえで重要ではあるが、少なくともこのような議論は、上記の (a) 及び (b) の主張を 2001 年時点で覆すものではない。

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

一方、文献 [7] では、その時点 (西暦年) で推奨される  $|n|$  の下限を与えている。上述の文献 [2] では、その時点で分解可能と予測される  $|n|$  を示しているのに対して、文献 [7] では、その時点で確保すべきマージンまで推定して推奨値の下限を提示しているため、同一時点で比較すると、文献 [2] より文献 [7] のほうが当然ながら大きな値となる。なお、文献 [7] では Moore の法則を若干拡張した独自版の仮定をもとに予測を行っている。

### 2.3.1.3 補足

一般に、既存のアルゴリズムは研究され改良される可能性があるため、最高速の楕円曲線法 (ECM) や一般数体ふるい法 (GNFS) がさらに高速になる可能性は否定できない。また格子利用法 (LFM) は、合成数の型によっては多項式時間で分解できる能力を有しており、考案されてから時間が経っていないことを考えれば、その潜在的な可能性は相対的に大きい。これらを踏まえたうえで、「2001 年時点では」(a)(b) の主張が成立していることを銘記すべきである。また、マージンの議論で述べたように、 $n = pq$  と  $n = p^2q$  がともに  $|p| = |q|$  かつ  $|n| = 1024$  の場合、 $n = pq$  と  $n = p^2q$  が ECM に対して同一のマージンを有しているわけではないことにも留意する必要がある。

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

- 量子計算機が実用化レベルに達したならば、素因数分解問題は効率的に解けるように

なり、素因数分解問題は、少なくとも暗号プリミティブとしての役割は終えることになる。

- 素因数分解問題と帰着関係にある数論的な問題に対する効率的なアルゴリズムにも注意を要する。例えば、 $n = p^r q$  ( $r > 1$ ) の分解問題は平方無縁部分 (squarefree part) 抽出問題に多項式時間で帰着する。平方無縁部分抽出問題とは、入力  $n$  に対して、 $n = u^2 v$  かつ  $v$  が平方無縁となる  $\{u, v\}$  を出力する問題である ( $v$  が平方無縁とは、 $v$  が  $a^2$  型の因数 ( $a > 1$ ) をもたないことである)。もし、これに対する効率的なアルゴリズムが発見されれば、結果として  $n = p^r q$  ( $r > 1$ ) 型の分解問題は効率的に解けることになる。ただし、 $n = pq$  ( $r = 1$ ) の分解問題が平方無縁部分抽出問題に帰着するとは知られていない。
- 構造的計算量理論において、計算量のクラスの包含関係に関する劇的な結果が証明されれば、素因数分解問題を含む数論的問題の多くが、その難しさに劇的な影響を受ける可能性がある。
- Moore の法則がいつ破綻するかにより、現実的に分解可能なサイズは影響を受ける。Moore の法則の寿命は 2005 年以降に延びたと考えられているが、もしそれが破綻すれば、少なくとも単体の CPU の高速化による脅威は緩和され、現実的に分解可能なサイズは頭打ちになる可能性がある。ただしその場合でも、莫大な数の CPU が参加する分散処理による分解が脅威として依然残る。

## 参考文献

- [1] D. Boneh, G. Durfee, N. Howgrave-Graham, "Factoring  $N = p^r q$  for large  $r$ ," Proc. Crypto'99, LNCS 1666, Springer-Verlag, pp.326–337, 1999.
- [2] R. P. Brent, "Recent progress and prospects for integer factorisation algorithms," Proc. COCOON 2000, LNCS 1858, Springer-Verlag, pp.3–22, 2000.
- [3] S. Cavallar, W. Lioen, H. te Riele, B. Dodson, A. K. Lenstra, P. L. Montgomery, B. Murphy, K. Aardal, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C. C. Putnam, P. Zimmerman, "Factorization of a 512-bit RSA modulus," Report MAS-R0007, CWI, Feb. 29, 2000.
- [4] D. Coppersmith, "Modifications to the number field sieve," J. Cryptology, vol.6, pp.169–180, 1993.
- [5] H. W. Lenstra, Jr., "Factoring integers with elliptic curves," Annals of Mathematics, vol.126, pp.649–673, 1987.
- [6] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, "The number field sieve," Proc. 22nd STOC, pp.564–572, 1990.
- [7] A. K. Lenstra, E. Verheul, "Selecting cryptographic key sizes," Proc. PKC 2000, LNCS 1751, Springer-Verlag, pp.446–465, 2000.
- [8] I. Miyamoto, Report on ECM-net, Oct. 2001.  
<http://www.loria.fr/~zimmerma/records/ecmnet.html>
- [9] E. Okamoto, R. Peralta, "Faster factoring of integers of a special form," IEICE Trans. Fundamentals, vol.E79-A, pp.489–493, 1996.

### 2.3.2 離散対数問題

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



る。また、問題そのものの難しさに影響を与えうる事項に関しても最後に補足する。

### 2.3.2.1 離散対数問題の定義

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

### 2.3.2.2 攻撃法

DLP に対する攻撃法は、一般の有限群に適用できる一般法と、有限体乗法群の性質を用いる index calculus 法 [6] に大別される。各々について知られている攻撃法およびその計算量を表にまとめる。一般法の計算量は、群位数ビット長の指数時間オーダーとなるが、index calculus 法の計算量は、準指数時間オーダーとなり、index calculus 法の方が高速であるといえる。

#### 一般法

$N$  を群の位数とすると、一般法の計算量は  $N$  に依存する。下記のうち、Pollard 法は並列化可能であり、 $m$  台の計算機を用いれば、DLP を解くのに要する計算量は 1 台あたり、 $\sqrt{\pi N/2}/m$  と見積もられる [7]。ECDLP の場合の解読記録に関しては、次節を参照されたい。

攻撃法の名称	計算量	文献
全数探索法	$O(N)$	
Pohlig-Hellman 法	素位数部分群上の DLP に帰着	[8]
Baby-Step/Giant-Step 法	$O(\sqrt{N})$	[10]
Pollard 法	$\sqrt{\pi N/2}$	[9]

#### index calculus 法

index calculus 法は、素体の乗法群に対して適用できる数体ふるい法と、小さな標数の拡大体の乗法群に対して適用できる関数体ふるい法とに大別される。数体ふるい法は更に、特殊数体ふるい法と一般数体ふるい法とに類別される。各々についてその計算量および 2001 年における解読記録を表にまとめる。表中、 $q$  は有限体の位数を表し、記号  $L_q[a, b]$  は、 $L_q[a, b] = e^{b(\log q)^a (\log \log q)^{1-a}}$  を表すものとする。

なお、2002 年になって、 $\mathbb{F}_{2^{607}}$  における DLP を解読したという報告 [13][12] もある。

攻撃法の名称	計算量	解読記録	文献
一般数体ふるい法	$L_q[1/3, c + o(1)], c = (64/9)^{1/3} = 1.9229\dots$	120 桁	[3, 11]
特殊数体ふるい法	$L_q[1/3, c + o(1)], c = 1.5262\dots$	129 桁	[14, 11]
関数体ふるい法	$L_q[1/3, c + o(1)], c = (32/9)^{1/3}$	$\mathbb{F}_{2^{521}}$	[4, 1]

### 2.3.2.3 安全な鍵のサイズ

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

#### Brent の予測式 [2]

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

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

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

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

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

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

年	ビット長
2019	1024
2042	2048
2070	4096

#### Lenstra-Verheul の表 [5]

Lenstra-Verheul の表 (以下、LV と記述) は、1982 年の DES と同程度の強度をその年に持つための鍵長を表している。Brent の予測式が、その年に解かれ得る鍵長を表しているのに対し、LV はその年に安全と思われるかなり大きなマージンをとった鍵長になっていることに注意されたい。

DLP 攻撃の最速アルゴリズム index calculus 法に対しては、基礎となる体のビット長を考慮する必要があるが、DSA など位数が比較的小さい部分群を用いるものは、部分群に対する一般攻撃法が index calculus 法よりも効率的にならないことも考慮に入れる必要がある。部分群 DLP を用いるものは、基礎となる体のビット長とともに部分群位数の

項も表の値以上に大きく取る必要がある。以下に LV の表から 10 年おきの値を抜粋する。

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

### 2.3.2.4 その他

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

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

## 参考文献

- [1] L. Adleman, A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In ANTS I(1994), L.Adleman and M.-D. Huang, Eds., vol. 877 of Lecture Notes in Computer Science.
- [2] R. P. Brent, “Recent progress and prospects for integer factorisation algorithms”, Proc. COCOON 2000, LNCS 1858, Springer-Verlag, pp.3–22, 2000.
- [3] A. Joux and R. Lercier, Discrete logarithms in  $GF(p)$ , Announcement on the NMBRTHRY Mailing List, 17. April 2001.
- [4] A. Joux and R. Lercier, Discrete logarithms in  $GF(2^n)$ , Announcement on the NMBRTHRY Mailing List, 25. September 2001.
- [5] A. K. Lenstra, E. Verheul, “Selecting cryptographic key sizes”, Proc. PKC 2000, LNCS 1751, Springer-Verlag, pp.446–465, 2000.  
<http://www.cryptosavvy.com/table.htm>
- [6] K. McCurley, The discrete logarithm problem. In Cryptography and computational number theory, Proc. Symp. Appl. Math(1990), C. Pomerance, Ed., vol.42 of Amer. Math. Soc., pp.49–74.
- [7] P. van Oorschot and M. Wiener, Parallel collision search with applications to hash functions and discrete logarithms, 2nd ACM Conference on Computer and Communications Security, 210-218, ACM Press 1994.
- [8] S. Pohlig and M. Hellman, An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance, IEEE Trans. on Infor. Th., 24, 106-110, 1978.
- [9] J. Pollard, Monte Carlo methods for index computations mod  $p$ , Math. Comp., 32, 918-924, 1978.
- [10] D. Shanks, Class number, a theory of factorization and genera, In Proc. Symp. Pure Math. 20(1971), AMS, Providence, R.I., pp.415–440.

- [11] O. Schirokauer, Discrete logarithms and local units, Phil. Trans. R. Soc. London A 345(1993), pp.409–423.
- [12] E. Thomé, Computation of discrete logarithms in  $\text{GF}(2^{607})$ , In Proc. ASI-ACRYPT 2001, LNCS 2248, pp. 107–124.
- [13] E. Thomé, Discrete Logarithms in  $\text{GF}(2^{607})$ .  
<http://www.lix.polytechnique.fr/Labo/Emmanuel.Thome/announcement/announcement.html>
- [14] D. Weber and T. Denny, The solution of mcurleys discrete logarithm challenge, In Advances in Cryptology - Crypto '98, vol. 1462 of LNCS, pp. 458–471.

### 2.3.3 楕円曲線上の離散対数問題

#### 2.3.3.1 定義

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

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

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

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

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

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

#### 2.3.3.2 攻撃法

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

##### 一般法

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

- 注1  $N$  がほぼ素数であるとは、 $N$  が  $N$  と同程度の大きさの素数と小さな整数 (1,2,4 等) の積であることを意味する。
- 注2 Baby-Step/Giant-Step 法や Pollard 法は、2001 年現在で  $N > 2^{160}$  であれば実行不可能であると思われる。
- 注3 Pollard 法は “Las Vegas” タイプのアルゴリズムで、その計算量評価は必要となる楕円曲線上の加算回数の統計的な見積りである。また、Pollard 法は並列化可能であり、 $m$  台の計算機を用いれば、ECDLP を解くのに要する計算量は 1 台あたり、 $\sqrt{\pi N/2}/m$  と見積もられる [3]。

### 特殊法

$N$  はほぼ素数であり、その最大素因子を  $l$  とする。

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

- 注1 自己同型法では楕円曲線の任意の自己同型が使用できるわけではない。Koblitz 曲線がもつ自己同型が自己同型法の対象となる典型である。ただし、その場合でも、自己同型法の効果は 163 ビット楕円曲線の場合に、ECDLP を解く計算量が最大で 5 ビット少なくなる程度である。Koblitz 曲線については第 2.4.2 節 ECDSA 参照。
- 注2 最近、Diem[12] によって奇標数の場合にも  $k = [\mathbb{F}_q : \mathbb{F}_p] = 5, 7$  のときには、Weil descent 法が成立する場合があることを示唆する結果が報告されている。OEF(Optimal Extension Field) を用いる場合注意が必要である。

#### 2.3.3.3 実験結果

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

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

また、Harley ら [14] は自己同型法を援用した並列化 Pollard 法を用いて、ECC challenge の一つである ECC2K-108 を解いている。ECC2K-108 は標数 2 の有限体上の位数 108 ビットの Koblitz 曲線である。ECC2K-108 を解くために、約 9500 台の計算機を用い、4 ヶ月をかけて、 $2.3 \times 10^{15}$  回の楕円曲線上の加算を実行したと報告されている。

### 2.3.3.4 安全な群位数サイズ

2001 年現在、楕円曲線上の離散対数問題は、特殊法にあげた特殊な楕円曲線を除けば、群位数 (より正確には、ベースポイントの位数) が 160 ビット以上の素因子を含めば十分安全であるとされている。将来のある時点における、安全な群位数サイズを見積もるには、解読アルゴリズムの実行に必要な実際的な計算量、インターネット資源等を活用して実行可能な最大計算量の増加、解読アルゴリズムの進歩等を見極める必要があり、その正確な数字を算出するのは困難である。ここでは、参考情報として、Lenstra と Verheul による結果 ([15]) を紹介する。

#### Lenstra と Verheul による安全な群位数サイズの見積もり [15]

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

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

### 参考文献

- [1] S. Pohlig and M. Hellman, An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance, *IEEE Trans. on Infor. Th.*, 24, 106-110, 1978.
- [2] J. Pollard, Monte Carlo methods for index computations mod  $p$ , *Math. Comp.*, 32, 918-924, 1978.
- [3] P. van Oorschot and M. Wiener, Parallel collision search with applications to hash functions and discrete logarithms, 2nd ACM Conference on Computer and Communications Security, 210-218, ACM Press 1994.
- [4] R. Gallant, R. Lambert and S. Vanstone, Improving the parallelised Pollard lambda search on binary anomalous curves, *Math. Comp.*, 69, 1699-1705, 2000.
- [5] M. J. Wiener and R. J. Zuccherato, Faster attacks on elliptic curve cryptosystems, *Selected Areas in Cryptography - SAC 1999*, Springer-Verlag LNCS 1556, 190-200, 1999.
- [6] A. Menezes, T. Okamoto and S. Vansone, Reducing elliptic curve logarithms to logarithms in finite fields, *IEEE Trans. on Infor. Th.*, 39, 1639-1646, 1993.
- [7] G. Frey and H. -G. Rück, A remark concerning  $m$ -divisibility and the discrete logarithm in the divisor class group of curves, *Mathematics of Computation*,

- 62, 865-874, 1994.
- [8] P. N. Smart, The discrete logarithm problem on elliptic curves of trace one, J. Cryptology 12, 193-196, 1999.
  - [9] T. Satoh and K. Araki, Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves, COMMENTARII MATHEMATICI UNIVERSITATIS SANCTI PAULI, vol. 47, No. 1, 81-92, 1998.
  - [10] I. A. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curves in characteristic p, Math. Comp. 67, 353-356, 1998.
  - [11] P. Gaudry, F. Hess and N. Smart, Constructive and destructive facets of Weil descent on elliptic curves, HP Labs Tech. Report, HPL-2000-10, To appear in J. Cryptology.
  - [12] C. Diem, The GHS-attack in odd characteristic, preprint, 2001. Available at <http://www.exp-math.uni-essen.de/~diem/english.html>
  - [13] A. Escott, J. Sager, A. Selkirk and D. Tsapakidis, Attacking elliptic curve cryptosystems using the parallel Pollard rho method, CryptoBytes - The Technical Newsletter of RSA Laboratories, volume 4, number 2, Winter 1999, 15-19. Also available at <http://www.rsasecurity.com>
  - [14] R. Harley, Elliptic Curve Discrete Logarithms: ECC2K-108. <http://crystal.inria.fr/~harley/ecdl17/>
  - [15] A. K. Lenstra, E. Verheul, Selecting cryptographic key sizes, Proc. PKC 2000, LNCS 1751, Springer-Verlag, pp.446-465, 2000.

## 2.4 詳細評価対象暗号 (個別暗号) の評価

### 2.4.1 DSA

#### 2.4.1.1 技術概要

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

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

#### 2.4.1.2 技術仕様

##### 鍵生成処理

DSA における鍵生成処理は以下の手順で行う。

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

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

### 署名生成処理

DSA において、平文  $M$  に対する署名生成処理は以下の手順で行う。

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

上記手順によって生成された  $(M, r, s)$  が署名文となる。

### 署名検証処理

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

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

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

#### 2.4.1.3 安全性評価

##### FIPS 186-2 Appendix 3 の乱数生成について

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

$$x = G(t, XVAL) \bmod q \quad \text{ただし、} t, XVAL \text{ は乱数種}$$

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

D. Bleichenbacher は、このような問題点を利用した攻撃法を指摘しており [3]、2001 年 10 月に、NIST は、FIPS 186-2 に Change Notice を追加して乱数生成の手順を修正した。Bleichenbacher の攻撃法の詳細は未だ明らかにされていないが、修正された手順によって生成した乱数を使用すれば元々の問題点を回避できることから、FIPS186-2 change notice に示された手順に従うことが望ましい。

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



られる。

### パラメータの選択について

上記技術仕様にも記載したように、DSA の元々の仕様では、パラメータ  $p$  の大きさを 512 ビットから 1024 ビットまで 64 ビット単位で選択可能である。これに対し、電子署名法に係る指針や、FIPS 186-2 Change Notice では、パラメータ  $p$  の大きさを 1024 ビットに限定している。さらに、NIST は、より大きなサイズを選択可能 (パラメータ  $q$  も含む) とするためにすでに仕様の改定を検討している [4]。現時点で安全なパラメータサイズがどれくらいであるかということに関しては様々な意見があり、正確な値を示すことは困難であるが、現在の計算機の能力でも 512 ビット程度では安全性を保つことが難しいというのは、ある程度一致した見解だと考えられる。したがって、我々は、パラメータ  $p$  の大きさとして、現在の仕様において最も安全なパラメータサイズ、すなわち 1024 ビットを選択することを強く推奨する。加えて、上記仕様変更も含めた今後の世の中の動向に十分注意を払うことが必要だと考える。なお、安全なパラメータサイズの問題に関しては、「2.3.2 離散対数問題」を参照されたい。

署名生成時に使用する乱数  $k$  に関しては、複数の平文を同じ乱数  $k$  を用いて署名すると秘密鍵  $x$  が算出されてしまう恐れがあるため、署名生成毎に異なる乱数  $k$  を選択することが必要である。

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

上記以外にも、いくつかの特殊なパラメータ (例えば、 $g = 0$ ) における攻撃法が報告されており、DSA の使用に際しては適正なパラメータを選択することが望まれる。

### 証明可能安全性について

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

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

### 参考文献

- [1] FIPS PUB(Federal Information Processing Standards publication) 186-2: DIGITAL SIGNATURE STANDARD (DSS).
- [2] ANSI X9.30 Public Key Cryptography for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA).
- [3] Lucent Technologies, Press releases: Scientist discovers significant flaw that would have threatened the integrity of on-line transactions, <http://www.lucent.com/press/0201/010205.bla.html>

- [4] NIST Second Key Management Workshop: Key Management Guideline (Draft), <http://csrc.nist.gov/encryption/kms/workshop2-page.html>
- [5] P. Q. Nguyen and I. E. Shparlinski, The insecurity of the Digital Signature Algorithm with partially known nonces, Journal of cryptology, to appear.

## 2.4.2 ECDSA

### 2.4.2.1 技術概要

ECDSA は楕円曲線を用いた署名方式である。ECDSA には複数の仕様があるが、ここでは 2000 年度暗号技術評価委員会に応募された ECDSA in SEC1[8][9] と電子署名法に係る指針に記載されている ANSI X9.62[1] を評価対象とする。ECDSA in SEC1 は SECG(Standards for Efficient Cryptography Group) によって策定された仕様である。

### 2.4.2.2 技術仕様

ECDSA による署名生成、署名検証は以下のように行われる。

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

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

**署名生成:** 平文  $M$  に対する署名生成を以下の手順で行う。

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

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

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

### 2.4.2.3 評価結果

2000 年度暗号技術評価結果を受けて今年度は特に以下の 2 つの観点で詳細に評価した。

- ECDSA の証明可能安全性、特に generic group model における証明可能安全性の検証
- Koblitz 曲線の安全性検証 (ECDSA in SEC1 に対して)

以下、上記の観点を中心に評価結果を述べるが、まず最初に 2 つの仕様の差異をまとめる。

■**ECDSA in SEC1 と ANSI X9.62 の差異** ECDSA in SEC1 と ANSI X9.62 の間で ECDSA の署名スキームは同一である。2つの仕様の差は主に以下の点にある。

- 推奨する具体的な楕円曲線パラメータ
- 擬似乱数生成器

ECDSA in SEC1 では、SEC2 ドキュメントに推奨される具体的な楕円曲線パラメータが挙げられている。標数  $p$  と標数 2 の楕円曲線に関し、複数の定義体サイズのもとで、検証可能な形式でランダムに選択された曲線と Koblitz 曲線とが推奨されている。検証可能な形式でランダムに曲線を選択する手法は ANSI X9.62 に記載されたものであり、これに関しては後で触れる。また、擬似乱数生成器に関しては具体的なアルゴリズムの記載がない。

一方、ANSI X9.62 では、付録にサンプルとして楕円曲線パラメータが挙げられているのみで、推奨される具体的な曲線パラメータはない。ただし、楕円曲線パラメータの選択手順が付録に示されており、検証可能な形式でランダムに選択する手法と、その他の手法 (Weil 法と CM 法の名前が挙げられている) が記載されている。また、擬似乱数生成器に関しては FIPS186 に記載された手法が規定されている。

■**プリミティブの安全性** ECDSA のプリミティブは楕円曲線上の離散対数問題に安全性の根拠を置いている。この問題の評価結果は「第 2.3.3 節 楕円曲線上の離散対数問題」を参照されたい。

■**証明可能安全性** ECDSA にはランダムオラクルモデルでの証明可能安全性は確認されていない。一方、ランダムオラクルモデルとは異なる generic group model (generic model とも呼ばれるが、ここでは generic group model で統一する) での安全性証明が Brown によってまとめられており [2]、今年度はこのレポートの検証や generic group model での証明の意義を検討した。

Brown は、ECDSA を抽象化した generic DSA に対して generic group model での安全性証明を議論している。

**定義 (Generic DSA)** 位数が素数  $n$  である加法群のベースポイント (原始元) を  $G$  とする。また、還元関数  $f: \langle G \rangle \rightarrow [0, n-1]$ 、ハッシュ関数  $h: \{0, 1\}^* \rightarrow [0, n-1]$  を定義する。

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

**署名生成:** 平文  $M$  に対する署名生成を以下の手順で行う。

- 1 ランダムな整数  $k \in [1, n-1]$  を選択。
- 2  $R = kG$  を計算。
- 3  $r = f(R)$  を計算。
- 4  $e = h(M)$  を計算。
- 5  $s = k^{-1}(e + dr) \bmod n$  を計算。
- 6  $(r, s)$  を平文  $M$  の署名とする。

**署名検証:** 平文  $M$ 、署名  $(r, s)$ 、公開鍵  $Q$  により下式が成り立つことを検証する。

$$r = f(s^{-1}h(M)G + s^{-1}rQ)。$$

generic DSA の具体的なインスタンスとして、楕円曲線上の群を利用し、還元関数  $f$  およびハッシュ関数  $h$  を具体的に定めた方式が ECDSA に相当する。

generic group model とは群要素の表現がランダムに与えられると仮定した仮想的なモデルである。すなわち、加法群  $\mathbb{Z}_n$  からビット列集合  $S \subset \{0, 1\}^*$  への全単射  $\sigma$  をランダムに定める generic group オラクルを仮定し、群要素の演算は generic group オラクルへの問い合わせにより実行するものとしたモデルである。

Brown はこのモデルの下で以下の定理を導出しており、ハッシュ関数の衝突困難性を前提とすれば generic DSA の (最強の意味での) 安全性を示すことができると主張している。

**定理** 適応的選択平文攻撃 (選択平文数を  $q$  とする) により、偽造者  $F_h$  が、実行時間  $\tau$  以下、確率  $\epsilon$  以上で存在的偽造に成功すると仮定すると、実行時間  $\tau'$  以下、確率  $\epsilon'$  以上でハッシュ関数  $h$  の衝突を求めるアルゴリズム  $C_h$  が存在する。ただし、 $\tau', \epsilon'$  は以下の通り:

$$\epsilon' \geq \epsilon - 3 \binom{\tau'}{2} / n, \quad \tau' \leq 2 \log n(\tau + q).$$

上記の議論を詳細に検討した結果、文献 [2] では証明の記載が簡略化されすぎてはいるものの、定理の正当性は概ね確認された。ここで「概ね」と言っているのは、外部評価者の中に独自に証明を構成した結果があり、確率や実行時間の評価式には違いが見られるものの、定理の主張が大筋として確認されたことを指している。

一方、generic group model での安全性証明の意義に関しては、以下のとおりである。

generic group model は、Nechaev[4] や Shoup[7] によって一般法での離散対数問題の計算量の下界を導出するのに利用され、その後、Schnorr-Jakobsson[5][6] によって generic group model とランダムオラクルモデルを組み合わせたモデルの下で Schnorr 署名や Signed ElGamal 暗号の安全性証明に利用されている。これら以外には、generic group model を利用した署名方式の安全性証明の議論はほとんど例が無く、現時点で安全性証明の手法として定着した技法とはいえない。証明可能安全性の議論で利用されることの多いランダムオラクルモデルと比較して、現状では研究の歴史に差があると考えられる。また、ECDSA と generic group model とのギャップが小さくないとする意見もある。すなわち、ECDSA の場合には群要素の表現を定める関数  $\sigma$  がランダムとはみなせないという主張であり、例えば次の具体例が挙げられる。適応的選択文書攻撃を拡張して、偽造者が署名オラクルから文書  $m$  に対する署名  $(r, s)$  を受け取り、同じ文書  $m$  に対する別の署名  $(r', s')$  を求める偽造を正当な攻撃とみなす場合、ECDSA は存在的偽造が可能である。文書  $m$  に対する署名  $(r, s)$  に対して、 $(r, -s)$  も文書  $m$  に対する署名となる性質を利用すれば偽造手法を簡単に示すことができる。一方、generic group model ではこのように選択文書攻撃を拡張した場合でも上記定理の証明が成立するため、ECDSA での実際と矛盾がある。

逆に、generic group model による安全性証明を評価する立場からは、現在の楕円曲線上の離散対数問題の解法アルゴリズムが群演算をブラックボックス化した generic group model 型の手法であることから、こうした攻撃者に対する安全性の一指標と考えられるという意見もある。

以上のように、generic group model での安全性証明は、その妥当性や現実的な意味合いが十分かどうかに関して決着がつかない。

■ **Koblitz 曲線の安全性** Koblitz 曲線とは 2 の拡大体上の曲線で次式によって定義されるもので、anomalous binary curve(ABC 曲線) とも呼ばれている:

$$y^2 + xy = x^3 + ax^2 + 1(a \in \{0, 1\})$$

ただし、この曲線は定義体が  $\mathbb{F}_{2^m}$  である  $\mathbb{F}_2$  上の曲線とする。この曲線の特徴として Frobenius 写像を用いることによって点のスカラー倍演算を高速に計算できることがあげられる。なお、素体  $\mathbb{F}_p$  上でも自己準同形写像が高速に計算できる曲線が存在し、ECDSA in SEC1 は、こうした曲線も含めて「Koblitz 曲線」として総称している。

Koblitz 曲線に固有の攻撃として、Wiener-Zuccherato[10] および Gallant-Lambert-Vanstone[3] が Koblitz 曲線上の離散対数問題に対して、 $\rho$  法での並列衝突探索法の若干の高速化手法を示している。この手法では、Koblitz 曲線の特長である点演算の高速性を利用しており、 $\mathbb{F}_{2^m}$  上の Koblitz 曲線に対し、 $\sqrt{2m}$  倍高速に離散対数問題が解ける。具体的には  $m$  が 160 のときに離散対数問題の計算ステップ数が約  $2^{76}$  に相当するため、通常の楕円曲線の場合と比べて約 16 倍高速に解けるが、計算量として大きな改善ではない。

上記の  $\rho$  法の高速化手法以外には現状では固有の攻撃法は発見されていないものの、Koblitz 曲線はかなり限定された曲線のクラスであり、そのクラス特有の攻撃が発見される可能性には注意すべきである。

■ **擬似乱数生成器** ANSI X9.62 の仕様には擬似乱数生成器として FIPS186-2 (DSA) の手法が記載されている。この擬似乱数生成器によって生成された  $k = rand \bmod n$  は  $[1, n - 1]$  で一様に分布しないため、この性質を利用した DSA に対する攻撃が Bleichenbacher によって指摘されている。NIST はこの攻撃の対策として擬似乱数生成器の改訂を FIPS186-2 change notice に示している。具体的には、2 つの乱数  $rand, rand'$  を用いて  $k = (rand || rand') \bmod n$  とするものである。現状では、擬似乱数生成器の仕様変更は ECDSA に対しては提唱されていないが、ECDSA での擬似乱数生成器にも同様の攻撃が適用される可能性があり、動向に注意すべきである。

この件に関しては「第 2.4.1 節 DSA」も参照されたい。

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

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

楕円曲線上の離散対数問題は「第 2.3.3 節」も参照されたい。

## 参考文献

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

- [2] D. Brown, “The exact security of ECDSA”, Technical Report CORR 200-34, Dept. of C&O, University of Waterloo, 2000. Available at <http://www.cacr.math.uwaterloo.ca>
- [3] R. Gallant, R. Lambert and S. Vanstone, “Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms”, Advances in Cryptology – CRYPTO2001, Lecture Notes in Computer Science **2139**, Springer-Verlag, pp.190–200, 2001.
- [4] V.I. Nechaev, “Complexity of a determinate algorithm for the discrete logarithm”, Math. Notes 55, pp.165–172, 1994.
- [5] C.P. Schnorr and M. Jakobsson, “Security of discrete log cryptosystems in random oracle + generic model”, Conference on the Mathematics of Public-Key Cryptography, The Fields Institute, Tronto, Canada, 2000. Available at <http://www.mi.informatik.uni-frankfurt.de/research/papers.html>
- [6] C.P. Schnorr and M. Jakobsson, “Security of Signed ElGamal Encryption”, Advances in Cryptology – ASISCRYPT2000, Lecture Notes in Computer Science **1976**, Springer-Verlag, pp.73–89, 2001.
- [7] V. Shoup, “Lower bounds for discrete logarithms and related problems”, Advanced in Cryptology – EUROCRYPT’97, Lecture Notes in Computer Science **1233**, Springer-Verlag, pp.256–266, 1997.
- [8] Standards for Efficient Cryptography, “SEC1:Elliptic Curve Cryptography”, Certicom Research, Ver.1.0, September 2000.
- [9] Standards for Efficient Cryptography, “SEC2:Recommended Elliptic Curve Domain Parameters”, Certicom Research, Ver.1.0, September 2000.
- [10] M. Wiener and R. Zuccherato, “Faster attacks on elliptic curve cryptosystems”, Selected Areas in Cryptography, Lecture Notes in Computer Science **1556**, pp.190–200, 1999.

## 2.4.3 ESIGN 署名

### 2.4.3.1 技術概要

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

電子署名法に係る指針に記載された署名方式は前者に属する。これを ESIGN(電子署名法に係る指針)と呼ぶ。IEEE P1363a[4] に提案された TSH-ESIGN 署名 [8] は後者に属する。

2001 年度、これらの 2 つの ESIGN 署名に対して評価を行い、次の評価結果を得た。

- ESIGN(電子署名法に係る指針)：電子署名法に係る指針に記載された安全性パラメータの一部(例えば SHA-1 を使い、かつ  $n$  が 2048 ビットの時、 $e$  が 8 以下の場合など)は、無視できない確率で署名の偽造に成功する。ハッシュ関数として指定された MD5 と SHA-1 のうち、MD5 の使用は推奨できない。
- TSH-ESIGN：証明される安全性は、従来知られていた安全性よりも少し弱い安全性である。

また、ESIGN 署名を電子政府で使用する場合は本章の最後で述べる。

本章の以下の節において、特に断わりのない場合は、2つの仕様の ESIGN 署名に対する共通の評価であると考えていただきたい。

### 2.4.3.2 技術仕様

ESIGN 署名は、何度かの仕様の変更が行われており、CRYPTREC 投稿版以外にも様々なバージョンがある [6]。

それらの仕様の違いを表 2.5 にまとめる。年数経過とともに、推奨パラメータはより大きくなっており、また安全性理論研究の進展によって、署名方式における最強の意味での安全性を (ある仮定のもとで) 証明できるスキームに変更されている。

表 2.5: 各種の ESIGN 署名

	推奨パラメータ	証明可能安全性
電子署名法に係る指針	$ n  \geq 1024, e \geq 8$	現時点では無い
CRYPTREC 2001	$ n  = 1152, e = 1024$	
CRYPTREC 2000	$ n  \geq 960, e \geq 8$	有 ( $n = p^2q$ 型素因数分解仮定, $e$ 乗根近似仮定とランダムオラクル モデルのもとで、適応的選択文書 攻撃に対して存在的偽造不可)
IEEE P1363a	(IEEE の方針により 規定無し)	
NESSIE†	$ n  = 1152, e = 1024$	

†NESSIE ではこの推奨パラメータに変更予定と提案者は述べている [6]

ESIGN(電子署名法に係る指針) と TSH-ESIGN のプリミティブ部分の仕様は共通であり、その概要は以下の通りである。

#### 鍵生成

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

#### 署名生成プリミティブ:SP-ESIGN

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

**署名検証プリミティブ:VP-ESIGN**

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

**注意:** ESIGN 署名が安全であるためには、2つの素数  $p, q$  は異なるものであることが必要である。しかし、ESIGN 仕様書 [6] には明示的にこのことが記述されていない。我々は、 $p$  と  $q$  は異なるものとして評価を行っている。

**2.4.3.3 プリミティブの安全性**

ESIGN 署名のプリミティブは、

- $e$  乗根近似問題
- $n = p^2q$  型素因数分解問題

の 2 つに安全性の根拠をおいている。この 2 つの問題のいずれかが解ければ、ESIGN 署名の秘密鍵が第 3 者に露呈するか、あるいは署名の偽造に成功する。我々はこの 2 つの問題について評価を行った。

■ **$e$  乗根近似問題** ESIGN 署名が安全性の根拠とする  $e$  乗根近似問題とは、次のような問題である [7]。

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

また、AER 問題が難しいという仮定は次のように定義される [7]。

**定義 2 (AER 仮定)** どのような確率的多項式時間アルゴリズム  $Adv$  に対しても、全ての定数  $c$ 、十分大きな値  $k$  に対して、

$$Pr[Adv(k, n, e, y) \rightarrow x] < 1/k^c$$

が成立するとき、 $e$  乗根近似問題は難しいという。ここで、 $0 \parallel y = [x^e \bmod n]^k$  であり、確率は  $\mathcal{G}$  と  $Adv$  の確率空間上で取られる。 $e$  乗根近似問題が難しいという仮定は、 $e$  乗根近似仮定 (AER 仮定) と呼ばれる。

- **$e = 2$  と  $e = 3$  の場合**

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

$x$  を  $n^{1/2}$  に近い整数とする。この時、 $x^2 \bmod n$  は  $O(n^{1/2})$  であり、メッセージ  $m = 0$  の場合の ESIGN 署名の検証式を満足する。この原理を任意の  $m$  に対して



適用できるように、連分数展開を用いて平方根の近似値を求めるようにした方法である。

Brickell と DeLaurentis の方法は、 $e = 3$  の場合にも容易に拡張できる。

また、 $e = 2$  の場合に、Vallée と Girault と Toffin は LLL アルゴリズムのような格子基底縮小アルゴリズムを利用した ESIGN 署名に対する署名偽造方法を発表した [12, 13]。格子基底縮小アルゴリズムを用いて有限体上の多変数多項式を解くことに関しては、Coppersmith による改良 [2, 3] が知られている。

- $e \geq 4$  の場合

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

■  $n = p^2q$  型素因数分解問題 法  $n$  の素因数分解が与えられれば  $e$  乗根近似問題を解くことができる。ESIGN 署名の法は、RSA 暗号で用いられる  $n = pq$  ( $p$  と  $q$  は同じ大きさ) という型とは異なり、 $n = p^2q$  ( $p$  と  $q$  は同じ大きさ) という型をしている。この型の素因数分解問題の困難さを考察することが必要である。素因数分解問題の困難さに関しては、第 2.3.1 節を参照のこと。

#### 2.4.3.4 スキームの安全性

前節で述べたように、ESIGN 署名には複数の仕様が存在する。メッセージエンコーディングの観点では、証明可能安全性を今のところ有しない ESIGN (電子署名法に係る指針) と、 $n = p^2q$  型素因数分解仮定と  $e$  乗根近似仮定と、ランダムオラクルモデルのもとで、適応的選択文書攻撃に対して存在的偽造不可証明可能安全性を有するとされる TSH-ESIGN の 2 つの仕様に分類される。それぞれの場合についての安全性評価を以下に述べる。

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

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

の 2 つを行う。前節では 1 の評価を行った。本節では 2 の評価を行う。まず、署名方式に対する攻撃の種類と、偽造の種類を表 2.6 と 2.7 に示す [11]。署名方式における最強の安全性は、

“適応的選択文書攻撃 (CMA) に対して存在的偽造不可”

ということである。

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

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

表 2.6: 署名方式への攻撃の種類

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

表 2.7: 署名の偽造の種類

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

という攻撃モデルを提案している [10]。SO-CMA は、1つのメッセージにつき、署名オラクルへの問い合わせは1回しか許されない (署名は1つしか入手できない) というものである。

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

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

Stern は TSH-ESIGN 署名に対し、ランダムオラクルモデルのもとで次の安全性を証明した [10]。

**定理 3 (Stern[10])** TSH-ESIGN 署名スキームに対して存在的偽造を生成する SO-CMA 攻撃者を  $\mathcal{A}$  とする。 $\mathcal{A}$  の攻撃成功確率を  $\varepsilon$ 、攻撃時間を  $\tau$ 、ハッシュ関数への問い合わせ回数を  $q_H$ 、署名オラクルへの問い合わせ回数を  $q_s$  とすると、 $e$  乗根近似問題は次式を満たす確率  $\varepsilon'$ 、時間  $\tau'$  で解ける。

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

$$\tau' \leq \tau + k(q_s + q_H) \cdot T_{exp}(k)$$

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

この定理の導出のアプローチおよび証明は、Shoup が OAEP に関して議論したこと [9] に関連している。この定理が証明していることは、一般の適応的選択文書攻撃 (CMA) に対する存在的偽造不可ではなく、SO-CMA に対する存在的偽造不可であることに注意。Stern はまた、次の2つも指摘している。

- 提案者による証明 [8] は、攻撃モデルとして SO-CMA を暗黙に仮定している
- 定理 3 を CMA へ拡張する方法は現在のところ知られていない

■ **ESIGN(電子署名法に係る指針)** 電子署名法に係る指針に記載された ESIGN 署名、すなわち 2001 度の CRYPTREC へ応募された ESIGN 署名では、EMSA と呼ばれるメッセージエンコーディング手法が適用されている [6]。ESIGN(電子署名法に係る指針) は、無視できない確率で署名の偽造に成功することが Stern により報告された [10]。この概要を以下に述べる。

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

$$00||PS||FF||H(m)$$

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

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

**定義 4 (変形版  $e$  乗根近似問題 [10])** 与えられた  $3k$  ビットの  $n$  と  $hLen$  ビットの  $v$  に対して、 $x^e \bmod n$  を 2 進表現した時に、ビット位置  $2k + 1, \dots, 2k + hLen$  に  $v$  が現れるような  $x$  を求めよ。

この変形版  $e$  乗根近似問題は、 $e$  が小さい時は容易に解ける。次の条件を満たす時に、署名の偽造が可能となる [10]。

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

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

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

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

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

また MD5( $hLen = 128$ ) の場合、

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

の時、署名の偽造が可能となる。例えば次のような場合に署名の偽造に成功する。

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

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

### 2.4.3.5 補助関数について

ESIGN 署名では、補助関数としてハッシュ関数を用いている。ESIGN(電子署名法に係る指針)には、ハッシュ関数として MD5 の使用を指定した方式と、SHA-1 の使用を指定した方式とがある。このうち、MD5 の使用を指定した方式は推奨できない。ハッシュ関数の安全性に関しては、文献 [5] および第 4 章を参照のこと。

### 2.4.3.6 実装性

提案者による実装 [7] では、Celeron 800MHz において、法  $n$  が 1152 ビット、安全性パラメータ  $e$  が 1024 の時、鍵生成 610ms、署名生成 1.04ms、署名検証 0.70ms である。

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

ESIGN 署名の署名生成速度は、RSA 署名と比べて高速であると言える。

### 2.4.3.7 ESIGN 署名のまとめ

- ESIGN(電子署名法に係る指針)：電子署名法に係る指針に記載された安全性パラメータの一部 (例えば SHA-1 を使い、かつ  $n$  が 2048 ビットの時、 $e$  が 8 以下の場合など) は、無視できない確率で署名の偽造に成功する。ハッシュ関数として指定された MD5 と SHA-1 のうち、MD5 の使用は推奨できない。
- TSH-ESIGN：証明される安全性は、従来知られていた安全性よりも少し弱い安全性である。
- ESIGN 署名が安全性の根拠としている  $e$  乗根近似問題は、RSA 署名が安全性の根拠としている、 $e$  乗根を求める問題よりも易しい。すなわち、根拠としている安全性の仮定が RSA 署名よりも強い。

■電子署名法に係る指針に記載された ESIGN 署名の使用について ESIGN(電子署名法に係る指針)は、指針に記載されているパラメータ  $e$ 、 $n$  の範囲に、無視できない確率で署名の偽造が可能なものが含まれている。よって電子政府での使用についての判断には、詳細な評価が必要である。また、ハッシュ関数として指定された MD5 と SHA-1 のうち、MD5 の使用は推奨できない。

## 参考文献

- [1] E. BRICKELL, J. DELAURENTIS, “An Attack on a Signature Scheme proposed by Okamoto and Shiraishi,” *Advances in Cryptology – CRYPTO’85*, LNCS, **218** (1986), Springer-Verlag, 28–32.

- [2] D. COPPERSMITH, “Finding a Small Root of a Univariate Modular Equation,” *Advances in Cryptology – EUROCRYPT’96*, LNCS, **1070** (1996), Springer-Verlag, 155–165.
- [3] D. COPPERSMITH, “Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known,” *Advances in Cryptology – EUROCRYPT’96*, LNCS, **1070** (1996), Springer-Verlag, 178–189.
- [4] “IEEE P1363a Draft Version 9 Standard Specifications for Public Key Cryptography: Additional Techniques,” *IEEE* (2001), available at <http://grouper.ieee.org/groups/1363/>
- [5] 情報処理振興事業協会, “暗号技術評価報告書 CRYPTREC Report 2000,” (2000).
- [6] NTT 情報流通プラットフォーム研究所, “ESIGN 仕様書,” (2001), available at <http://info.isl.ntt.co.jp/esign/CRYPTREC/index-j.html>
- [7] NTT 情報流通プラットフォーム研究所, “ESIGN 自己評価書,” 2001 年度 CRYPTREC 応募書類, (2001), available at <http://info.isl.ntt.co.jp/esign/CRYPTREC/index-j.html>
- [8] T. OKAMOTO, E. FUJISAKI, H. MORITA, “TSH-ESIGN:Efficient Digital Signature Scheme Using Trisection Size Hash,” Submission to P1363a, available at <http://grouper.ieee.org/groups/1363/StudyGroup/submissions.html> (1998)
- [9] V. SHOUP, “OAEP Reconsidered,” *Advances in Cryptology – CRYPTO 2001*, LNCS, **2139** (2001), Springer-Verlag, 239–259. Also available at <http://shoup.net/papers/>
- [10] J. STERN, “Evaluation Report on the ESIGN Signature Scheme,” (2001)
- [11] 宇根正志, 岡本龍明, “公開鍵暗号の理論研究における最近の動向,” IMES Discussion Paper Series 98-J-28, (1998)
- [12] B. VALLÉE, M. GIRAULT, P. TOFFIN, “How to Break Okamoto’s Cryptosystem by Reducing lattice Bases,” *Advances in Cryptology – EUROCRYPT’88*, LNCS, **330** (1988), Springer-Verlag, 281–291.
- [13] B. VALLÉE, M. GIRAULT, P. TOFFIN, “How to Guess  $l$ th Roots Modulo  $n$  by Reducing Lattice Bases,” *AAECC-6*, LNCS, **357** (1988), Springer-Verlag, 427–442.

## 2.4.4 RSA (RSA-OAEP, RSA-PSS, RSA 署名)

### 2.4.4.1 技術概要

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

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

RSA-OAEP は RSA 法のプリミティブが一方向性をみたす (この条件は、「RSA 問題の困難性」とよばれる) と仮定して、最強の安全性 (IND-CCA2) をみたすという証明可能安全が示された暗号方式である。

一方、RSA を利用した署名に関しては、1) いわゆる「教科書的」RSA 署名、2) ANSI X9.31 3) RSA-PKCS #1 v1.5 (電子署名法に係る指針に記載された方式\*4)、4) RSA-FDH (Full-Domain Hash Schemes: FDH)、5) RSA-PSS (Bellare-Rogaway の論文版)、6) RSA-PSS (IEEE P1363a 版) など、多数の仕様が存在する。

詳細評価にあたっては、電子署名法に係る指針に記載されている RSA-PKCS #1 v1.5、および 2001 年度 CRYPTREC に応募された RSA-PSS (IEEE P1363a 版) を対象とした。

#### 2.4.4.2 技術仕様

■**RSA プリミティブ** 公開鍵を  $(N, e)$ 、秘密鍵を  $(N, d)$  とする。ここで、 $e$  は 3 以上の奇数で  $\text{GCD}\{e, (p-1)(q-1)\} = 1$  をみたし、 $d$  は  $de \equiv 1 \pmod{\text{LCM}\{p-1, q-1\}}$  をみたす。

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

$$\text{RSAEP}((n, e), x) = \text{RSAVP}((n, e), x) = x^e \pmod{N} \quad (2.1)$$

で、復号プリミティブ RSADP / 署名生成プリミティブ RSASP を

$$\text{RSADP}((n, d), y) = \text{RSASP}((n, d), y) = y^d \pmod{N} \quad (2.2)$$

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

■**RSA-OAEP** RSA-OAEP の構成は以下の通り。

**EME-OAEP-Encode** ( $M, P, \text{emLen}$ )

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

\*4 本方式は、規格書 RSA-PKCS #1 v1.5 で規定され、規格書 RSA-PKCS #1 v2.0 以降にも引き継がれているため、本報告書では、方式名として RSA-PKCS #1 v1.5 と略記する。

\*5 RSA 社より提出された RSA-OAEP の仕様書 (01espdif) では (関数  $G$  に対応する)  $\text{MGF}$  の出力データのバイト数が  $\text{emLen} - \text{hLen}$  となっているが、 $\text{emLen} - \text{hLen} - 1$  が正しいと思われる。

**EME-OAEP-Decode** ( $EM, P$ )

1.  $P$  のオクテット長がハッシュ関数の入力制限 (SHA-1 の場合には  $2^{61} - 1$  オクテット) よりも長いなら、“decoding error” を出力して停止する。
2.  $emLen < 2hLen + 2$  なら、“decoding error” を出力して停止する。
3.  $EM = X \parallel MaskedSeed \parallel MaskedDB$  とおく。ここで  $|X| = 1, |MaskedSeed| = hLen, |MaskedDB| = emLen - hLen - 1$ 。
4.  $seedMask = MGF(MaskedDB, hLen)$  とおく。
5.  $seed = MaskedSeed \oplus seedMask$  とおく。
6.  $dbMask = MGF(seed, emLen - hLen - 1)$  とおく。
7.  $DB = MaskedDB \oplus dbMask$  とおく。
8.  $hLen$  オクテットの列を  $pHash = Hash(P)$  とする。
9.  $DB = pHash' \parallel M'$ 。ここで、 $|pHash'| = hLen$ 。
10.  $M' = \alpha \parallel T \parallel M$ 。  $T$  は  $M'$  中の zero でない最左 octet とする。  $|T| = 1$ 。
11. もし  $pHash' \neq pHash, X \neq 00$  あるいは  $T \neq 01$  ならば、“decoding error” を出力する\*6。
12.  $M'$  から  $T$  と  $\alpha$ (すべてが zero からなる) を除いて  $M$  を生成する。
13.  $M$  を出力する。

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

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

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

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

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

■**RSA-PSS** RSA-PSS の構成は以下の通り。

**EMSA-PSS-Encode** ( $M, emBits$ )

1.  $M$  のオクテット長がハッシュ関数の入力制限 (SHA-1 の場合には  $2^{61} - 1$  オクテット) よりも長いなら、“message too long” を出力して停止する。
2. 長さ  $hLen$  オクテットの列、 $mHash = Hash(M)$  を生成する。

\*6 このステップで、エラーの理由にかかわらず、同時に同一のエラー通知で出力することとした。文献 [9] で指摘された攻撃に対する対策として、今回新たに導入された実装法である。





7.  $MaskedDB$  の最左オクテット中の左から  $(8emLen - emBits)$  ビットが zero に一致しなければ、“inconsistent” を出力して停止する。
8.  $dbMask = MGF(H, emLen - hLen - t)$  とする。
9.  $DB = MaskedDB \oplus dbMask$  とする。
10.  $DB$  の左から  $(8emLen - emBits)$  ビットを zero に設定する。
11.  $DB$  の右から  $(emLen - hLen - sLen - t - 1)$  オクテットが zero に一致しない、あるいは右から第  $(emLen - hLen - sLen - t)$  オクテット目が 01 に一致しなければ、“inconsistent” を出力して停止する。
12.  $DB$  の最後の  $sLen$  オクテットを  $salt$  とする。
13.  $m' = 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ ||\ mHash\ ||\ salt$ 。  $m'$  は先頭に 8 個の zero octet を含む  $(8 + hLen + sLen)$  オクテットの列。
14.  $hLen$  オクテットの列を  $mHash' = Hash(m')$  とする。
15.  $H = H'$  ならば “consistent” を出力する。他の場合は “inconsistent” を出力する。

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

#### RSA-PSS-Sign $((n, d), M)$

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

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

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

■**RSA-PKCS #1 v1.5** RSA-PKCS #1 v1.5 (電子署名法に係る指針に記載された方式) は、規格書 RSA PKCS #1 v1.5 [11] で規定され、規格書 RSA PKCS #1 v2.0 [12] 以降にも引き継がれている。

RSA 社から出版されたこの二つの文書における署名法に関する記述は以下のように整理できる。

1. 規格書 v1.5 には、ハッシュ関数 MD5 についての記述 (OID) はあるが、ハッシュ関数 SHA-1 についての記述 (OID) はない (文献 [11] の 11 章)。
2. 規格書 v2.0 には、EMSA-PKCS1-v1.5 エンコーディング手法として、新たに SHA-1 が利用できるように記述されている (OID がある) (文献 [12] の 10.1 節)。

EMSA-PKCS1-v1.5 エンコーディング手法として規定されたフォーマットは図 2.3 のとおり。ここで、 $T$  は Distinguished Encoding Rule (DER) によって規定されており、先頭のフィールドはハッシュ関数を特定し、次のフィールドはハッシュ値を含んでいる。

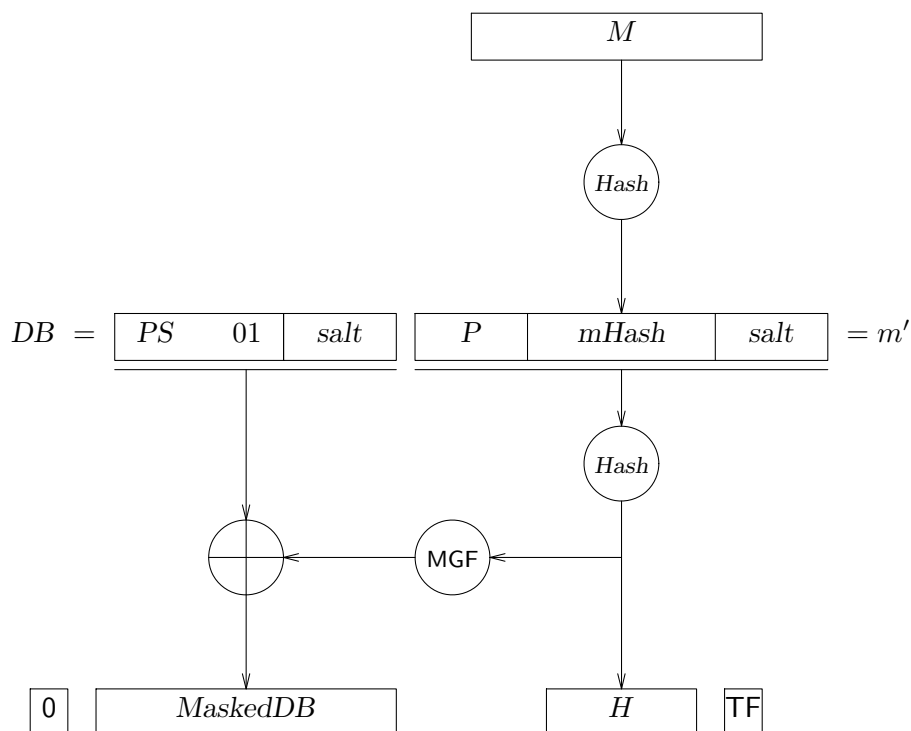


図 2.2: RSA-PSS

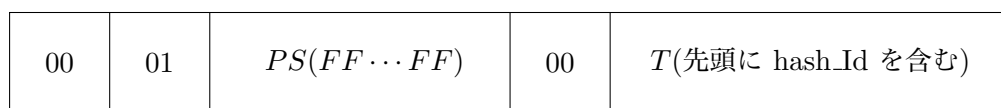


図 2.3: EMSA-PKCS1-v1.5 の出力形式

また、 $PS$  は、値  $FF$  (16 進表示) から構成された 8 オクテット以上のオクテット列である。

#### 2.4.4.3 安全性

■プリミティブの安全性 RSA 法のプリミティブは、

- $n = pq$  型素因数分解問題

に安全性の根拠をおいている。素因数分解問題についての評価結果は第 2.3.1 節に報告する。RSA 法の証明可能安全性については、上記の型の素因数分解の困難性との同値性は示せていないが、経験的に安全であると信じられている。長期間広く使われている実績、広範な観点からの安全性評価が行われてきた。すべて (4 名) の評価者から、RSA プリミティブに関する自己評価書の記述に問題なしと報告されている。すでに指摘されている使用制約としては、例えば、暗号・署名に共通の事項として、1) 法の値の共有、2) 秘密鍵  $d$  が小さい場合の脅威、3) 鍵の部分情報から全体の情報の導出の脅威などが報告されていた。また、暗号として使用する場合の事項として、1) 公開鍵  $e$  が小さいときの脅威

(Coppersmith の攻撃) 2) 同報通信環境での脅威 (Håstad, Coppersmith の攻撃) などが報告されている。

■**RSA-OAEP の安全性** 2000 年末に Shoup によって Bellare-Rogaway によるオリジナル論文 [3] の安全性の主張に誤りが指摘されたことで [10]、安全性についての議論が学会で議論された。

その結果、安全性の証明帰着の効率が低下するものの、RSA-OAEP としての証明可能安全性は RSA 問題の困難性を前提として成り立つことが示された [7]。また、2001 年に OAEP 実装上の問題点が指摘されたが [9]、現在の仕様では対処済みであることが確認されている。

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

安全性の証明された署名法として、全域ハッシュ法 (Full Domain Hash Schemes:FDH) [2] や、本人確認法からデジタル署名を構成する変換法 [6] が提案されているが、PSS はこれらの方式に比べてより緊密 (tight) な帰着関係を証明できる特徴を有している。

Coron は FDH について帰着関係を緊密にする技法 (Coron の技法) [5] を提案している。Jonsson は、PSS の安全性を Coron の技法を適用して再評価している [8]。論文 [8] では、帰着関係を緊密 (tight) に評価したことに加えて、以下を示した。

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

Jonsson の証明は、Hash-ID については検討の対象外としているが、乱数成分 (salt) 長が可変でそれを悪用する攻撃、エンコード中で使用される二つの関数に相関がある場合も含めて、安全性の評価が行われており、信頼できると判断した。

RSA-PSS SIGNATURE SCHEME. RSA-PSS has been proven to be existentially unforgeable against adaptive chosen-message attacks in the random oracle model under the RSA assumption. The reduction is tight, and so the assurances provided by the security proof are strong. More guidance should be given on the desired length of the salt used in the RSA-PSS encoding operation.

新たなパラメータの導入によって帰着の効率が低下する可能性もあるので、salt のサイズ、法のサイズの選択にも注意が必要となる。

■**RSA-PKCS #1 v1.5** RSA-PKCS #1 v1.5 については安全性に関する証明は与えられていないが、安全性に対する疑問点の指摘もなかった。ただし、多くの署名法のエンコーディング手法について署名の偽造手法が提案されていることにより、図 2.3 で示したエンコーディング手法の安全性について、検討を継続する必要がある。

RSA-PKCS #1 v1.5 には証明可能安全性が示されていないことより、同程度の効率を

実現でき、かつ安全性が保障されている RSA-PSS を RSA-PKCS #1 v1.5 の代わりに採用すればよい、との指摘があった。

RSA-PKCS1-v1.5 SIGNATURE SCHEME. There is no formal proof of security of RSA-PKCS1-v1.5. Since RSA-PSS is as efficient as RSA-PKCS1-v1.5 and has a security proof, there is no reason to use RSA-PKCS1-v1.5.

#### 2.4.4.4 まとめ

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

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

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

**注意 5** 文献 [3] ではビット単位の表記が使われており、RSA-OAEP ではバイト単位の表記が使われていることに注意を要する。

帰着の効率の観点から、RSA-OAEP の代わりに RSA-OAEP<sup>+</sup> を推奨する意見もあった。

RSA-OAEP ENCRYPTION SCHEME. RSA-OAEP has been proven to be semantically secure against adaptive chosen-ciphertext attacks in the random oracle model under the RSA assumption. However, the reduction is not tight, and thus it is not clear what security assurances the proof provides. We recommend that RSA-OAEP be modified to RSA-OAEP<sup>+</sup> which has a tighter security reduction, and furthermore can be easily modified to allow encryption of arbitrarily-long messages (see [10]).

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

RSA-PKCS# v1.5 の方式の利用状況、方式の寿命等を考慮して、RSA-PSS も電子署名法に係る方式に追加する議論を行う必要がある。

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

PSS[4]  $\longleftrightarrow$  EMESA-PSS

---

$m \longleftrightarrow P \parallel mHash$

$r \longleftrightarrow salt$

$k_0 \longleftrightarrow 8 \times sLen$

$0^{k_2} \longleftrightarrow PS$

$k_2 \longleftrightarrow emBits - 8 \times hLen - 8 \times sLen - 8$

$k \longleftrightarrow emBits$

$k_1 \longleftrightarrow 8 \times hLen$

$w \longleftrightarrow H$

$s \parallel t \longleftrightarrow MaskedDB$  で先頭  $(8emLen - emBits)$  ビットに zero を設定

$G(\cdot) \longleftrightarrow MGF(\cdot, emLen - hLen - 1)$  で先頭  $(8emLen - emBits)$  ビットに zero を設定

$H(\cdot) \longleftrightarrow Hash\ function$

---

**注意 6** 文献 [4] ではビット単位の表記が使われており、RSA-PSS ではバイト単位の表記が使われていることに注意を要する。

■電子署名法に係る指針に記載された RSA 署名 (RSA-PKCS #1 v1.5) の使用について 現時点で特に安全性の問題は存在しないが、図 2.3 で示したエンコーディング手法の安全性の検討を継続する必要がある。電子署名法に係る指針では、図 2.3 中の  $T$  を生成する方法として MD5 と SHA-1 がハッシュ関数として指定されているが、CRYPTREC Report 2000 で指摘されているように、MD5 の使用は推奨できない。

## 参考文献

- [1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO'98*, pages 26–45. Springer, 1998. Lecture Notes in Computer Science No. 1462.
- [2] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of the First ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
- [3] M. Bellare and P. Rogaway. Optimal asymmetric encryption — how to encrypt with RSA. In A.D. Santis, editor, *Advances in Cryptology — EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Berlin, Heidelberg, New York, 1995. Springer-Verlag.
- [4] M. Bellare and P. Rogaway. The exact security of digital signatures — how to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Berlin, Heidelberg, New York, 1996. Springer-Verlag.
- [5] J. S. Coron. On the exact security of full domain hash. In M. Bellare, editor, *Advances in Cryptology — CRYPTO2000*, volume 1880 of *Lecture Notes in*

- Computer Science*, pages 229–235, Berlin, Heidelberg, New York, 2000. Springer-Verlag.
- [6] A. Fiat and A. Shamir. How to prove yourself. In A.M. Odlyzko, editor, *Advances in Cryptology — CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–208, Berlin, Heidelberg, New York, 1986. Springer-Verlag.
- [7] E. Fujisaki, T. Okamoto, D. Pointcheval and J. Stern. RSA-OAEP is chosen-ciphertext secure under the RSA assumption. In J. Kilian, editor, *Advances in Cryptology — CRYPTO2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274, Berlin, Heidelberg, New York, 2001. Springer-Verlag.
- [8] Jakob Jonsson. Security proofs for the RSA-PSS signature schemes and its variants –draft 1.1. Available at <http://eprint.iacr.org/2001/053/>, 2001.
- [9] J. Manger. A chosen ciphertext attack on rsa optimal asymmetric encryption padding (OAEP) as standardized in PKCS# v2.0. In J. Kilian, editor, *Advances in Cryptology — CRYPTO2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 230–238, Berlin, Heidelberg, New York, 2001. Springer-Verlag.
- [10] V. Shoup. OAEP reconsidered. In J. Kilian, editor, *Advances in Cryptology — CRYPTO2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259, Berlin, Heidelberg, New York, 2001. Springer-Verlag.
- [11] PKCS #1: RSA Encryption Standard, An RSA Laboratories Technical Note Version 1.5, Revised November 1, 1993
- [12] PKCS #1 v2.0: RSA Cryptography Standard, RSA Laboratories, October 1, 1998

## 2.4.5 EPOC-2

### 2.4.5.1 技術概要

EPOC-2 は、守秘を目的とした暗号アルゴリズムである。合成数  $n = p^2q$  ( $p, q$  は異なる素数) の素因数分解問題が困難であるとの仮定と、ランダムオラクルモデルのもとで、適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であるという証明可能安全性を有することを旨として設計された公開鍵暗号である。基本的には、[2] で提案された暗号プリミティブを、[1] で提案された IND-CCA2 を実現する手法で変換することにより構成されている。この手法では共通鍵暗号をその一部として利用している。

EPOC-2 は CRYPTREC2000 応募の同名暗号の改訂版として応募された。その改訂では、エンコーディング手法がより厳密に規定されているのみであった。暗号技術としては本質的には変わっていないため、評価委員会での検討の結果、引続き詳細評価されることとなった。

### 2.4.5.2 技術仕様

EPOC-2 のプリミティブ部分の仕様 (概要) は以下の通りである。詳細は仕様書を参照されたい。

■**鍵生成:KGP-OU** KGP-OU は次のように定義される。

入力:  $k$  セキュリティパラメータ、非負整数  
 出力:  $PK$  OU 公開鍵  $(n, g, h, pLen)$   
 $SK$  OU 秘密鍵  $(p, q, pLen, w)$

処理手順:

1. 二つの素数  $p, q (2^{k-1} \leq p, q < 2^k, p \neq q)$  を選び、 $n := p^2q$  を計算する。
2.  $g_p := g^{p-1} \bmod p^2$  の位数が  $p$  となるような  $g \in (\mathbb{Z}/n\mathbb{Z})^*$  をランダムに選ぶ。
3.  $h_0 \in (\mathbb{Z}/n\mathbb{Z})^*$  をランダムに選ぶ。  $h := h_0^n \bmod n$  とする。
4.  $pLen := k$  とする。
5.  $w = L(g_p)$  とする。ここで  $L(x) = \frac{x-1}{p}$  である。
6.  $PK = (n, g, h, pLen), SK = (p, q, pLen, w)$  を出力する。

■暗号化プリミティブ:EP-OU EP-OU は次のように定義される。

入力:  $PK$  OU 公開鍵  $(n, g, h, pLen)$   
 $m$  メッセージ、 $0 \leq m < 2^{pLen-1}$  なる整数  
 $r$  乱数、 $0 \leq r < n$  なる整数  
 出力:  $c$  暗号文、 $0 \leq c < n$  なる整数  
 エラー: “invalid”  
 処理手順:

1.  $m$  が  $0 \leq m < 2^{pLen-1}$  を満たさないのならば “invalid” をエラー出力して処理を終了する。
2.  $c = g^m h^r \bmod n$  を計算する。
3.  $c$  を出力する。

■復号プリミティブ:DP-OU DP-OU は次のように定義される。

入力:  $SK$  OU 秘密鍵  $(p, q, pLen, w)$   
 $c$  暗号文、 $0 \leq c < n$  なる整数  
 出力:  $m$  メッセージ、 $0 \leq m < 2^{pLen-1}$  なる整数  
 エラー: “invalid”  
 前提条件: 秘密鍵  $SK$  が正しい  
 処理手順:

1. 暗号文  $c$  が  $0 \leq m < n$  を満たさないのならば “invalid” をエラー出力して処理を終了する。
2.  $c_p = c^{p-1} \bmod p^2$  を計算する。
3.  $m = \frac{L(c_p)}{w} \bmod p$  を計算する。
4.  $m$  が  $0 \leq m < 2^{pLen-1}$  ならばそれを出力する。それ以外は “invalid” をエラー出力して処理を終了する。

次にスキーム部分の仕様を簡単に紹介する。スキームは、上述の暗号プリミティブを [1] で提案された方法で変換したものである。スキームでは、EME3 と呼ばれるエンコーディング処理等を用いている。詳細は仕様書を参照されたい。

■暗号化スキーム:ES-EPOC-2-Encrypt ES-EPOC-2-Encrypt の入出力は以下の通りである。

入力:	$PK$	OU 公開鍵 $(n, g, h, pLen)$
	$M$	オクテット列として表現された暗号化されるメッセージ
	$P$	オクテット列 (空でもよい)
出力:	$(C_1, C_2)$	二つのオクテット列からなる暗号文
エラー:	“invalid”	

■復号スキーム:DS-EPOC-2-Decrypt DS-EPOC-2-Decrypt の入出力は以下の通りである。

入力:	$PK$	OU 公開鍵 $(n, g, h, pLen)$
	$SK$	OU 秘密鍵 $(p, q, pLen, w)$
	$(C_1, C_2)$	二つのオクテット列よりなる復号対象である暗号文
	$P$	オクテット列であるエンコーディングパラメータ (空でもよい)
出力:	$M$	オクテット列であるメッセージ
エラー:	“invalid”	

### 2.4.5.3 安全性評価

■証明可能安全性 用いられている暗号プリミティブは、元となった論文 [2] と若干異なっている。具体的には、KGP-OU における  $h_0$  の選択および  $h$  の生成方法である。EPOC-2 では、

$$h_0 \in (\mathbb{Z}/n\mathbb{Z})^* \text{ をランダムに選ぶ。 } h := h_0^n \bmod n \text{ とする。}$$

であるのに対し、[2] では、

$$h = g^n \bmod n$$

となっている。評価の結果、複数の評価者から「応募された EPOC-2 の仕様では IND-CCA2 であることが確認できない」との指摘があった。またこれらの評価者は、[2] と同様に  $h$  を選択すれば IND-CCA2 を満たすとも指摘している。この点については、次節のように提案者から仕様の変更希望が出されている。

■ $n = p^2q$  型素因数分解問題 EPOC-2 暗号で用いられている法  $n$  は、RSA 暗号で用いられる  $n = pq$  ( $p$  と  $q$  は同じ大きさ) という形とは異なり、 $n = p^2q$  ( $p$  と  $q$  は同じ大きさ) という形をしている。したがって暗号の安全性を議論するに際しては、この形の素因数分解問題の困難さを詳しく考察することが必要である。素因数分解問題の困難さに関しては、別項で評価結果を報告しているのでそれを参照されたい。

■その他注意すべき点 共通鍵暗号でブロック暗号を使用する際には、その利用方法 (mode of operation) に注意すべきである。これは、仕様書に利用方法についての注意が書かれていないため、利用方法によっては安全なブロック暗号を使用していても IND-CPA が満たされず、結果として全体が IND-CCA2 とならない危険性があるためである。



仕様書には最低限の記述しかないので、素数  $p$  等パラメータの選択にあたっては、細心の注意を払うべきである。

#### 2.4.5.4 その他

平成 14 年 1 月 28 日に開催された暗号技術評価ワークショップのランプセッションにおいて、提案者より仕様の変更希望が出された。さらに平成 14 年 1 月 30 日、暗号技術評価委員会が電子メールでコメントを募集している宛先に、提案者より仕様を変更した場合の安全性証明について自己評価した内容が送られてきた。以下 (A), (B) は電子メールで送られてきた内容である。

(A) EPOC-2 仕様書内 5.1 章「 $h$  の選択」において

- (1)  $h=g^n \bmod n$  と  $h$  を選択した場合 (他のパラメータは仕様書に準拠する), 証明付き安全性が保証されることを証明した詳細なレポートが NESSIE 2001 プロジェクト提出の自己評価資料にあります。同レポートは

<http://info.isl.ntt.co.jp/epoc/nessie/index-j.html>

から入手可能です。

- (2) 現仕様書のように  $h$  を選択した場合も特に具体的な攻撃法は知られておりません。

- (B) 現仕様書では、 $hLen$  が Eurocrypt'98 と比べて短く設定されています (付録 B 内) が、(1) の条件を満たす (すなわち  $h=g^n \bmod n$ ) ならば、証明付き安全性は保証されます。但し、安全性の帰着効率、 $hLen$  を Eurocrypt'98 と同様に設定したときと比べてほぼ  $2/3$  に低下いたします。上記自己評価資料に詳しく評価してあります。

なお暗号技術評価委員会では応募と異なる仕様については評価の対象としないため、この内容についての審議は行っていない。またこの  $hLen$  の記述については、[2] ではなく、[1] の記述と比較しての話であるように思われる。

#### 参考文献

- [1] E. Fujisaki and T. Okamoto, Secure Integration of Asymmetric and Symmetric Encryption Schemes, CRYPTO'99, LNCS1666, pp.537-554, Springer-Verlag, 1999.
- [2] T. Okamoto and S. Uchiyama, A New Public-Key Cryptosystem as Secure as Factoring, Eurocrypt'98, LNCS1403, pp.308-318, Springer-Verlag, 1998.
- [3] Specification of EPOC-2, CRYPTREC2001 応募書類.
- [4] Self Evaluation of EPOC-2, CRYPTREC2001 応募書類.

## 2.5 監視状態の暗号の評価

### 2.5.1 ECIES in SEC1

#### 2.5.1.1 技術概要

ECIES in SEC1 は 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。本暗号技術は、2000 年度は ECAES in SEC1 として CRYPTREC に応募されていたが、2001 年度には暗号技術名を ECIES in SEC1 に変更して応募されている。ただし、両者は同一の方式である。

#### 2.5.1.2 技術仕様

ECIES in SEC1 の仕様 (概要) は以下の通りである。詳細については、仕様書を参照されたい。

楕円曲線パラメータ (i)  $(p, a, b, G, n, h)$  または (ii)  $(m, f(x), a, b, G, n, h)$ :

(i) 素体  $\mathbb{F}_p$  上の楕円曲線  $E: y^2 = x^3 + ax + b (a, b \in \mathbb{F}_p)$ 、 $E$  上の素数位数  $n$  の有理点  $G \in E(\mathbb{F}_p)$  及び  $h = \#E(\mathbb{F}_p)/n$  からなるパラメータ。

(ii)  $\mathbb{F}_2$  上の  $m$  次既約多項式  $f(x)$  によって表される体  $\mathbb{F}_{2^m}$ 、 $\mathbb{F}_{2^m}$  上の楕円曲線  $E: y^2 + xy = x^3 + ax^2 + b (a, b \in \mathbb{F}_{2^m})$ 、 $E$  上の素数位数  $n$  の有理点  $G \in E(\mathbb{F}_{2^m})$  及び  $h = \#E(\mathbb{F}_{2^m})/n$  からなるパラメータ

KDF: 鍵導出関数 (Key Derivation Function)

MAC: メッセージ認証子 (Message Authentication Code)

ENC: 共通鍵暗号 (Symmetric Key Encryption) もしくは排他的論理和 XOR

秘密鍵: 整数  $d (\in [1, n-1])$

公開鍵:  $G$  の  $d$  倍点  $Q$

**暗号化:** 平文  $M$  の入力に対して、暗号文  $C$  の出力を以下の手順で行う。

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

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

1.  $C = R || EM' || D'$  により、 $R', EM', D'$  を計算する
2.  $dR'$  の  $x$  座標  $Z' = x(dR')$  を計算する
3.  $K' = \text{KDF}(Z')$  を計算し、ENC で用いる鍵  $EK'$  及び MAC で用いる鍵  $MK'$  を  $K' = EK' || MK'$  により計算する
4.  $D' = \text{MAC}(MK', EM')$  が成り立つかどうか検証する。もし成り立たなければ、“invalid” を出力する

5.  $EM'$  を鍵  $EK'$  を用いた ENC によって復号： $M = ENC(EK', EM')$  (XOR のときは、 $M = EK' \oplus EM'$ ) を行う
6. 平文  $M$  を出力する

### 2.5.1.3 安全性評価

ECIES in SEC1 は楕円曲線上の離散対数問題の困難性に依存した公開鍵暗号である。

現在、楕円曲線上の離散対数問題に対して種々の攻撃法が知られているが、ECIES in SEC1 においては、実用上有効な各種ビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC 2 ドキュメントに具体的に示されている（ただし、これらは推奨パラメータであって他の楕円曲線の使用を禁じるものではない）。それらの楕円曲線は検証可能な形でランダムに選定された楕円曲線と Koblitz 曲線と呼ばれる楕円曲線（2.4.2.3 節の「Koblitz 曲線の安全性」を参照）からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC 2 に含まれているが、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある（2.4.2.3 節の「Koblitz 曲線の安全性」を参照）。

ECIES in SEC1 の仕様は、その基になっている論文 [1][2][3] の記述にある方式と多少異なる箇所が見受けられる。ここで、論文 [1][2][3] にある方式は、用いる共通鍵暗号 ENC とメッセージ認証子 MAC が安全であるとき、適応的ハッシュ Diffie-Hellman 独立仮定 [1]（もしくは、オラクル Diffie-Hellman 仮定 [2][3]）のもとで、適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であることが証明されている。ただし、これらの仮定はごく最近に提案されたものであるため、その妥当性に関しては注意が必要である。また、ECIES in SEC1 の仕様は論文 [1][2][3] の方式と異なる箇所が見受けられるため、この証明可能安全性の議論の有効性についても注意すべきである。

一方、2000 年度詳細評価者によって、KDF の部分をランダム関数と仮定するランダムオラクルモデルのもとで ECIES in SEC1 の安全性が、楕円曲線上の Gap-Diffie-Hellman 問題に帰着できることが指摘されている（Gap-Diffie-Hellman 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）。つまり、ECIES in SEC1 は、用いる共通鍵暗号 ENC とメッセージ認証子 MAC が安全であるとき、ランダムオラクルモデルのもとで、楕円曲線上の Gap-Diffie-Hellman 仮定が正しければ適応的選択暗号文攻撃に対して強秘匿 (IND-CCA2) であることが指摘されている。しかし、Gap-Diffie-Hellman 問題は比較的新しい問題なので注意が必要である。

### 2.5.1.4 最近の動向

2000 年度評価報告では、ECIES in SEC1 の安全性に関して特に問題点は指摘されなかったため、2001 年度評価では本暗号を監視状態にある暗号技術として取り扱った。ところが、最近、V. Shoup により ECIES に対しての問題点が指摘されている [7]。したがって、今後はこの問題点を含めた継続的な安全性評価が必要であると思われる。

## 参考文献

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

- [2] M. Abdalla, M. Bellare and P. Rogaway, “The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES”, Topics in Cryptology, - CT-RSA 2001, LNCS 2020, pages 143-158, Springer-Verlag, 2001.
- [3] M. Abdalla, M. Bellare and P. Rogaway, “DHIES: An encryption scheme based on the Diffie-Hellman Problem”, full version of [4] [2], September 18, 2001. Available at <http://www-cse.ucsd.edu/users/mihir/>
- [4] M. Bellare and P. Rogaway, “Minimizing the Use of Random Oracles in Authenticated Encryption Schemes”, Information and Communications Security, LNCS 1334, pages 1-16, Springer-Verlag, 1997.
- [5] Certicom Research, standards for efficient cryptography group (SECG), September 20, 2000. Version 1.0. Available at <http://www.secg.org/>
- [6] IEEE P1363a/D9 - standard specifications for public key cryptography: Additional techniques, June 2001. Draft version 9.
- [7] V. Shoup, “A Proposal for an ISO Standard for Public Key Encryption”, Version 2.0, September 17, 2001. Version 2.1, December 20, 2001. Available at <http://shoup.net/papers/>

## 2.5.2 ECDH in SEC1

### 2.5.2.1 技術概要

ECDH in SEC1 は 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。本暗号技術は、2000 年度は ECDHS in SEC1 として CRYPTREC に応募されていたが、2001 年度には暗号技術名を ECDH in SEC1 に変更して応募されている。ただし、両者は同一の方式である。

### 2.5.2.2 技術仕様

ECDH in SEC1 の仕様 (概要) は以下の通りである。詳細については、仕様書を参照されたい。

楕円曲線パラメータ (i)  $(p, a, b, G, n, h)$  または (ii)  $(m, f(x), a, b, G, n, h)$ :

(i) 素体  $\mathbb{F}_p$  上の楕円曲線  $E: y^2 = x^3 + ax + b (a, b \in \mathbb{F}_p)$ ,  $E$  上の素数位数  $n$  の有理点  $G \in E(\mathbb{F}_p)$  及び  $h = \#E(\mathbb{F}_p)/n$  からなるパラメータ。

(ii)  $\mathbb{F}_2$  上の  $m$  次既約多項式  $f(x)$  によって表される体  $\mathbb{F}_{2^m}$ ,  $\mathbb{F}_{2^m}$  上の楕円曲線  $E: y^2 + xy = x^3 + ax^2 + b (a, b \in \mathbb{F}_{2^m})$ ,  $E$  上の素数位数  $n$  の有理点  $G \in E(\mathbb{F}_{2^m})$  及び  $h = \#E(\mathbb{F}_{2^m})/n$  からなるパラメータ

KDF: 鍵導出関数 (Key Derivation Function)

#### 初期設定:

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

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

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

### 2.5.2.3 安全性評価

ECDH in SEC1 の安全性は楕円曲線上の離散対数問題に依存している。

現在、楕円曲線上の離散対数問題に対して種々の攻撃法が知られているが、ECIES in SEC1 においては、実用上有効な各種ビット数に対して、既知の攻撃法が適用できない楕円曲線パラメータが SEC 2 ドキュメントに具体的に示されている（ただし、これらは推奨パラメータであって他の楕円曲線の使用を禁じるものではない）。それらの楕円曲線は検証可能な形でランダムに選定された楕円曲線と Koblitz 曲線と呼ばれる楕円曲線（2.4.2.3 節の「Koblitz 曲線の安全性」を参照）からなる。Koblitz 曲線は高速処理可能で使用実績があるため SEC 2 に含まれているが、限定されたクラスの楕円曲線であるため、そのクラス特有の攻撃法が出現する可能性に注意を払う必要がある（2.4.2.3 節の「Koblitz 曲線の安全性」を参照）。

ECDH in SEC1 は楕円曲線を利用した Diffie-Hellman 鍵共有方式であり、鍵共有方式として最も基本的なものである。受動的攻撃に対して、大きな問題点は指摘されていない。しかし、能動的な攻撃に対しては安全でなく、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。

鍵共有方式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、ECDH 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 を実現する鍵共有方式が複数の研究者によって提案されている。

## 参考文献

- [1] Certicom Research, standards for efficient cryptography group (SECG), September 20, 2000. Version 1.0.

## 2.5.3 DH

### 2.5.3.1 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 [1]。

### 2.5.3.2 技術仕様

#### システム共通パラメータの設定

1. 素数  $p$  を生成する。
2.  $g \in \mathbb{Z}_p^*$  を位数  $l$  の原始元とする。

$(p, l, g)$  をシステム共通のパラメータとする。

#### ユーザー A の初期設定

1.  $0 < x_A < l$  を満たす乱数  $x_A$  を生成する。
2.  $y_A = g^{x_A} \bmod p$  を計算する。

$x_A$  を秘密鍵、 $y_A$  を公開鍵とする。

#### ユーザー B の初期設定

1. 同様にして、 $0 < x_B < l$  を満たす乱数  $x_B$  を生成する。
2.  $y_B = g^{x_B} \bmod p$  を計算する。

$x_B$  を秘密鍵、 $y_B$  を公開鍵とする。

#### 鍵共有の処理

1. A の処理:  $K = y_B^{x_A} \bmod p = g^{x_B x_A} \bmod p$
2. B の処理:  $K = y_A^{x_B} \bmod p = g^{x_A x_B} \bmod p$

により、 $K$  を共有する。

#### 2.5.3.3 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、個々のプロトコ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 11770-3, Oakley, PGP）。評価対象は、基本的なスキームのみである。
- b) 秘密共有プロトコルとしての基本的な部分は、受動的攻撃のみを仮定した場合、Diffie-Hellman 問題に帰着される。ただし、ランダムなビット列と区別できないという意味においては、範囲を制限するなどの工夫によって Decisional Diffie-Hellman 問題に帰着される。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  $\alpha$  倍することにより、異なる鍵を共有させる。  
Reveal attacks: public な WS などでの操作で、secret coin(秘密にしておくべき情報) が漏れた場合、その他の秘密情報に影響を及ぼす (forward secrecy の欠如)。  
Attacks intrinsic 2-flow AKE(Authenticated Key-Exchange protocols): 2つしか flow がなく、2つめの flow が1つ目の flow と独立な場合には、strong-corruption model で forward secrecy がない、A-to-B/B-to-A authentication がないなどの問題がある。
- f) 公開鍵に対する署名を組み合わせるなどの改良を加えることにより、解決される問題もある。

## 参考文献

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

## 2.6 スクリーニング評価対象暗号の評価

### 2.6.1 OK-ECDSA

#### 2.6.1.1 技術概要

OK-ECDSA は署名方式に属する。ANSI X9.62 等で標準化されている ECDSA を変形した方式である。OK-ECDSA は IC カード上で安全に効率良く実装可能とすることを狙って設計されている。楕円曲線としてモンゴメリ型楕円曲線を利用することに特徴があり、署名のスキームには ECDSA をそのまま適用している。

モンゴメリ型楕円曲線上の演算は点の  $y$  座標を用いずに実行できる。このことによりコードサイズ、メモリ使用量を増大させることなく高速に処理可能であると提案者は主張している。さらに、秘密パラメータであるスカラー値に依存せずに同一の実行手順でスカラー倍算が計算できるため、タイミング攻撃や SPA 攻撃に対して高い耐性を備え、さらにランダム化射影座標を利用することで DPA 攻撃に対しても高い耐性を持つと提案者は主張している。

### 2.6.1.2 スクリーニング評価結果

OK-ECDSA はランダムオラクルモデルでの証明可能安全性を備えていない。また、提案者が主張するサイドチャネル攻撃に対する耐性は、自己評価書に記載されている内容だけではその強度を十分に確認できない。

■**証明可能安全性** OK-ECDSA は利用する楕円曲線の種類を除いて ECDSA と同じである。ECDSA にランダムオラクルモデルでの証明可能安全性が確認されていないことから、OK-ECDSA にもランダムオラクルモデルでの証明可能安全性は認められない。

なお、ECDSA には generic group model での安全性証明を与えた結果 [1] があるが、generic model による署名方式の安全性証明はランダムオラクルモデルでの証明に比べて学会での研究の歴史が浅い。ECDSA と generic group model とのギャップが小さくないとする意見がある反面、群演算をブラックボックス化して扱う攻撃者に対するモデルとして評価する意見もあり、そのモデルの妥当性や現実的な意味合いが十分かどうかに関して決着がついていない。

■**プリミティブの安全性** プリミティブの安全性はモンゴメリ型楕円曲線の離散対数問題をベースとする。モンゴメリ型楕円曲線は限定された曲線であるが、ワイエルシュトラス型楕円曲線の約 40%が変換可能との予想 [2] があり、限定の度合は小さいと考えられる。また、モンゴメリ型楕円曲線に固有の攻撃法も発見されていない。

なお、モンゴメリ型楕円曲線の位数は常に 4 で割り切れる。したがって、離散対数問題に対して、モンゴメリ型でない曲線でコファクタ 1 の場合と同等の安全性とするためには、OK-ECDSA の提案者が推奨しているように、モンゴメリ型楕円曲線での定義体のサイズを 2 ビット程度大きくし、コファクタが 4 の曲線を利用することが望ましい。

■**サイドチャネル攻撃耐性** OK-ECDSA のサイドチャネル攻撃耐性はその強度を十分に確認できない。

OK-ECDSA はスカラー倍演算の手順まで規定することでサイドチャネル攻撃耐性を高めることを狙って設計されているが、ECDSA 等の他の署名方式でも同様にスカラー倍演算の手順まで規定することで耐性を高めることが可能である。このように演算手順での対策は OK-ECDSA に固有のものではない。

また、サイドチャネル攻撃に対する耐性は、上記のような演算手順を含むアルゴリズムレベルでの差異以外に実装レベルでの差異にも大きく影響される可能性がある。OK-ECDSA のような演算手順での対策だけで十分なサイドチャネル攻撃耐性を持つかどうかは疑問であり、OK-ECDSA の提案者が主張する耐性は、ある側面だけをとらえたものに過ぎない可能性がある。サイドチャネル攻撃の脅威が大きい場面として IC カード上のインプリメントが挙げられるが、OK-ECDSA の自己評価書には IC カード実装の結果や電力解析攻撃の実験結果が示されていない。このため、演算手順で対策を講じていても実装段階での配慮が欠けた場合に攻撃可能となるのではという疑問を否定するだけの根拠がない。

なお、一般にサイドチャネル攻撃耐性に関しては、十分な耐性を確保するために考慮すべき観点に関して、現状では確立されたオープンな基準がない。したがって、詳細に評価



を行うにはコストがかかる反面、十分な評価であるかどうかの客観性は確保しにくい。サイドチャンネル攻撃耐性を特徴とする方式を正当に評価するためには、客観的評価法の確立が課題である。

■**実装性** 暗号技術仕様書の内容には特に不備は見られない。第三者実装も可能と考えられる。

サイドチャンネル攻撃に対して、演算手順での対策を施した方式間で処理速度やプログラムサイズ、メモリサイズ等を比較することができる。OK-ECDSA の自己評価書には演算量の観点でこうした比較が一部示されており、ダミー演算を加えたバイナリアルゴリズムによるスカラー倍算と比べて OK-ECDSA で規定されているスカラー倍算は約 1/2 の処理量であり、OK-ECDSA の高速性が認められる。ただし、ソフトウェアやハードウェアを実際に試作した上での性能比較は行われていない。したがって、特にメモリサイズやプログラムサイズ等の実装面での優位性は確認できない。

## 参考文献

- [1] D. Brown, “The exact security of ECDSA”, Technical Report CORR 200-34, Dept. of C&O, University of Waterloo, 2000. Available at <http://www.cacr.math.uwaterloo.ca>
- [2] 伊豆 哲也, “楕円曲線暗号演算の計算法について”, 1999 年暗号と情報セキュリティシンポジウム (SCIS'99), pp.275-280, 1999.

## 2.6.2 NTRU

### 2.6.2.1 技術概要

NTRU は、守秘を目的とした暗号化アルゴリズムである。演算は、整数係数一変数多項式環の演算を基本としており、安全性は、ラティスの最短ベクトル問題等に関連していると考えられる。

### 2.6.2.2 技術仕様

NTRU の基本演算を以下に記述する。

- パラメータ  
 $N, p, q$  : 正整数、ただし  $p, q, X^N - 1$  は互いに素  
 $R = \mathbb{Z}[X]/(X^N - 1)$   
 多項式の演算は、環  $R$  において行う
- 鍵生成  
 $f(X), g(X) \in \mathbb{Z}[X]$  : 小さな係数を持つ任意の多項式を選択  
 $f_q(X) \equiv f(X)^{-1} \pmod{q}$   
 $f_p(X) \equiv f(X)^{-1} \pmod{p}$   
 $h(X) \equiv pf_q(X)g(X) \pmod{q}$

$f(X)$ ,  $f_p(X)$  が秘密鍵、 $h(X)$  が公開鍵

- 暗号化
  - $m(X) \in \mathbb{Z}[X]$  : メッセージ、係数を mod  $p$  でみた多項式
  - $r(X) \in \mathbb{Z}[X]$  : 小さな係数を持つ任意の多項式を選択
  - $e(X) \equiv r(X)h(X) + m(X) \pmod{q}$  : 暗号文
- 復号
  - $a(X) \equiv f(X)e(X) \pmod{q}$
  - $d(X) \equiv f_p(X)a(X) \pmod{p}$  : 復号されたメッセージ

NTRU の詳細な仕様では、これらの基本演算以外にメッセージフォーマット等の処理を行う必要があるが、ここでは記述を省略する。

### 2.6.2.3 スクリーニング評価結果

- 既存の公開鍵暗号に対する処理性能の高速性を、応募者は主張している。スクリーニング評価では、処理性能測定評価を実施してはいないが、理論的見地からは、応募者の主張はある程度妥当と考えられる。
- 証明可能安全性について、応募者は、
  - (1) メッセージのランダムパディングにより IND-CPA になる
  - (2) IND-CPA に藤崎-岡本変換を施し、IND-CCA2 を達成できる
 と主張しているが、(1) の証明が与えられていない。実績が少ない比較的新規の暗号にとって、良く研究された問題への帰着証明が与えられていないことは、不利な点と考えられる。
- NTRU が安全性の根拠とする問題は、ラティスの最短ベクトル問題あるいは近隣ベクトル問題であるが、NTRU の場合、問題となるラティスは CML (Convolution Modular Lattice) と呼ばれる特殊な形をしたものとなる。このため、一般のラティスに対する LLL アルゴリズム以外の、特殊な形を利用した攻撃法の存在が懸念される。今後、Convolution Modular Lattice に対する攻撃研究動向に注意する必要があると考えられる。例えば [1] は、 $N$  が合成数の場合の NTRU 攻撃論文である。ただし、 $N$  を、推奨パラメータで挙げられているように素数にとれば、この攻撃法は適用することができない。

### 参考文献

- [1] C. Gentry, Key Recovery and Message Attacks on NTRU-Composite, Advances in Cryptology–EUROCRYPT 2001, Springer-Verlag, LNCS 2045, pp.182–194 (2001).

## 2.6.3 HIME(R)

### 2.6.3.1 暗号技術

HIME(R) は、法  $N = p^d q$  を用いた Rabin-OAEP である。すなわち、秘密鍵は素数  $p, q$ 、公開鍵は整数  $N$  であり、平文  $m$  の暗号化や暗号文  $y$  の復号は概略、以下のように行われる (詳細は仕様書を参照)。

#### 暗号化

1.  $x \leftarrow m$  の OAEP エンコード
2.  $y \leftarrow x^2 \bmod N$

#### 復号

1.  $x_i \leftarrow y^{1/2} \bmod N$
2.  $m_i \leftarrow x_i$  の OAEP デコード

復号の第 1 ステップで、 $N$  を法とする平方根演算を必要とするが、HIME(R) では、CRT を用いた従来法 [1] ではなく、整数の  $pq$  進展開を用いた独自のアルゴリズムが用いられている。応募者の主張によれば、

1. HIME(R) は、 $N = p^d q$  型の素因数分解問題の困難性を仮定として、ランダムオラクルモデルのもとで適応的選択暗号文攻撃に対して強秘匿であることが証明される。
2. HIME(R) は、暗号化、復号とも RSA-OAEP よりも高速である。

### 2.6.3.2 スクリーニング評価結果

スクリーニング評価の結果、HIME(R) の暗号技術仕様書は、記述レベルでの誤りはあるものの、概ね問題ない。しかし、自己評価書には以下に示す問題点があった。

1. 復号処理で用いられている独自の平方根演算アルゴリズムは、自己評価書において従来法よりも高速であるとされているが、その現実的な効果に対して疑問が残る。
2. 自己評価書における、 $N = p^d q$  型の素因数分解問題を解くのに必要な計算量の評価は十分でない。安全な  $N$  の具体的なビット長を算出するためには、数体ふるい法や楕円曲線法の計算量に対して、漸近的なレベルを越えて、実験に基づく定量的評価が必要である。
3. HIME(R) がランダムオラクルモデルのもとで IND-CCA2 であることは、Rabin-SAEP[2] が IND-CCA2 であるなど、他者の結果と比較して相応である。しかしながら、自己評価書における安全性証明の正しさには疑問が残る。
4.  $d = 2, 3$  以外の場合の記述がない。 $d = 2, 3$  以外の場合をどう扱うのか不明である。

以上の問題点は、上記応募者の主張 1,2 の正当性を覆すほどのものではない。

## 参考文献

- [1] T. Takagi, Fast RSA-type Cryptosystem Modulo  $p^kq$ , Advances in Cryptology - CRYPTO '98, LNCS 1462, Springer-Verlag, pp. 318-326, 1998.
- [2] D. Boneh, Simplified OAEP for the RSA and Rabin functions, Advances in Cryptology - CRYPTO 2001, LNCS 2139, Springer-Verlag, pp. 275-291, 2001.

## 2.6.4 OK-ECDH

### 2.6.4.1 技術概要

OK-ECDH は鍵共有方式に属する。ANSI X9.63 等で標準化されている ECDH を変形した方式である。OK-ECDH は IC カード上で安全に効率良く実装可能とすることを狙って設計されている。楕円曲線としてモンゴメリ型楕円曲線を利用することに特徴があり、鍵共有のスキームには ECDH をそのまま適用している。

モンゴメリ型楕円曲線上の演算は点の  $y$  座標を用いずに実行できる。このことによりコードサイズ、メモリ使用量を増大させることなく高速に処理可能であると提案者は主張している。さらに、秘密パラメータであるスカラー値に依存せず同一の実行手順でスカラー倍算が計算できるため、タイミング攻撃や SPA 攻撃に対して高い耐性を備え、さらにランダム化射影座標を利用することで DPA 攻撃に対しても高い耐性を持つと提案者は主張している。

### 2.6.4.2 スクリーニング評価結果

OK-ECDH はランダムオラクルモデルでの証明可能安全性を備えていない。また、提案者が主張するサイドチャネル攻撃に対する耐性は、自己評価書に記載されている内容だけではその強度を十分に確認できない。

■**証明可能安全性** OK-ECDH は利用する楕円曲線の種類を除いて ECDH と同様である。ECDH は受動的攻撃に対する問題点は指摘されていないものの、能動的攻撃に対して脆弱なことが指摘されている。署名方式との組み合わせ等が推奨される。

■**プリミティブの安全性** プリミティブの安全性はモンゴメリ型楕円曲線の離散対数問題をベースとする。モンゴメリ型楕円曲線は限定された曲線であるが、ワイエルシュトラス型楕円曲線の約 40%が変換可能との予想 [1] があり、限定の度合は小さいと考えられる。また、モンゴメリ型楕円曲線に固有の攻撃法も発見されていない。

なお、モンゴメリ型楕円曲線の位数は常に 4 で割り切れる。したがって、離散対数問題に対して、モンゴメリ型でない曲線でコファクタ 1 の場合と同等の安全性とするためには、OK-ECDH の提案者が推奨しているように、モンゴメリ型楕円曲線での定義体のサイズを 2 ビット程度大きくし、コファクタが 4 の曲線を利用することが望ましい。

■**サイドチャンネル攻撃耐性** OK-ECDH のサイドチャンネル攻撃耐性はその強度を十分に確認できない。

OK-ECDH はスカラー倍演算の手順まで規定することでサイドチャンネル攻撃耐性を高めることを狙って設計されているが、ECDH 等の他の鍵共有方式でも同様にスカラー倍算の手順まで規定することで耐性を高めることが可能である。このように演算手順での対策は OK-ECDH に固有のものではない。

また、サイドチャンネル攻撃に対する耐性は、上記のような演算手順を含むアルゴリズムレベルでの差異以外に実装レベルでの差異にも大きく影響される可能性がある。OK-ECDH のような演算手順での対策だけで十分なサイドチャンネル攻撃耐性を持つかどうかは疑問であり、OK-ECDH の提案者が主張する耐性は、ある側面だけをとらえたものに過ぎない可能性がある。サイドチャンネル攻撃の脅威が大きい場面として IC カード上のインプリメントが挙げられるが、OK-ECDH の自己評価書には IC カード実装の結果や電力解析攻撃の実験結果が示されていない。このため、演算手順で対策を講じていても実装段階での配慮が欠けた場合に攻撃可能となるのではという疑問を否定するだけの根拠がない。

なお、一般にサイドチャンネル攻撃耐性に関しては、十分な耐性を確保するために考慮すべき観点に関して、現状では確立されたオープンな基準がない。したがって、詳細に評価を行うにはコストがかかる反面、十分な評価であるかどうかの客観性は確保しにくい。サイドチャンネル攻撃耐性を特徴とする方式を正当に評価するためには、客観的評価法の確立が課題である。

■**実装性** 暗号技術仕様書の内容には特に不備は見られない。第三者実装も可能と考えられる。

サイドチャンネル攻撃に対して、演算手順での対策を施した方式間で処理速度やプログラムサイズ、メモリサイズ等を比較することができる。OK-ECDH の自己評価書には演算量の観点でこうした比較が一部示されており、ダミー演算を加えたバイナリアルゴリズムによるスカラー倍算と比べて OK-ECDH で規定されているスカラー倍算は約 1/2 の処理量であり、OK-ECDH の高速性が認められる。ただし、ソフトウェアやハードウェアを実際に試作した上での性能比較は行われていない。したがって、特にメモリサイズやプログラムサイズ等の実装面での優位性は確認できない。

## 参考文献

- [1] 伊豆 哲也, “楕円曲線暗号演算の計算法について”, 1999 年暗号と情報セキュリティシンポジウム (SCIS'99), pp.275-280, 1999.

## 2.6.5 PSEC-KEM

### 2.6.5.1 技術概要

PSEC-KEM は、楕円曲線 ElGamal 暗号関数をプリミティブとして利用した、[2] において述べられている意味での鍵カプセル化メカニズムである。[2] によれば、鍵カプセル

化メカニズムは、ハイブリッド暗号を構成する際に、その構成要素として利用できるものである。鍵カプセル化メカニズムは、暗号化アルゴリズム  $E$  と復号アルゴリズム  $D$  からなり、暗号化アルゴリズム  $E$  は受信者の公開鍵を入力として暗号文  $c_0$  と共有鍵  $k$  を出力する。復号アルゴリズム  $D$  は、秘密鍵と暗号文  $c_0$  を入力として、暗号側と同じ共有鍵  $k$  を求める。

PSEC-KEM は、守秘目的の PSEC-2 (CRYPTREC2000 への応募暗号) の改訂版として応募されたが、公開鍵暗号評価小委員会での検討の結果、「新規応募」としてスクリーニング評価を実施することになった。

### 2.6.5.2 スクリーニング評価結果

#### 証明可能安全性

[2] では、鍵カプセル化メカニズムとしての、IND-CCA2 (適応的選択暗号文攻撃に対して強秘匿) の定義が述べられている。応募者の主張によれば、PSEC-KEM はランダムオラクルモデルにおいて、ECDH の困難性の仮定のもと、上記意味での IND-CCA2 の証明可能安全性を有することが特徴である。自己評価書においては、次のような能力を持つ攻撃者 AdversaryA を用いれば、ECDH 問題を計算するアルゴリズムを構成できることを証明している。

AdversaryA : 暗号文  $c_0$  とビット列  $k^*$  を与えられたとき、オラクルを用いて、ビット列  $k^*$  がランダムなものか正しい鍵かを判別できる。

スクリーニング評価の結果、次の点が確認された。

- 鍵カプセル化メカニズムにおける証明可能安全性の定義は、[2] によるものであり、妥当であると認める。ただし、鍵カプセル化メカニズムの安全性定義は、[1] における公開鍵暗号の安全性定義とは異なっており、IND-CCA2 の強度が最強かどうかは示されていない。
- 自己評価書における、IND-CCA2 の証明に問題は見当たらない。
- ランダムオラクルモデルの概念は、暗号スキームの安全性証明の有効な手法として広く知られており、妥当と考えられる。

ただし、ある評価者からは、本応募暗号がより一般的な鍵共有法としての証明可能安全性を示したのではないことを注意することとの指摘がある。つまり、もし鍵共有法としての安全性を示すならば、セキュリティモデルを提示し、そのもとで議論すべきであるとの指摘である。

#### その他

- 自己評価書では、守秘目的の暗号方式との比較が記載されているが、本応募暗号と守秘目的の暗号方式とは、種別も安全性の定義が異なるため比較できないとの指摘がある。
- プリミティブとして用いる楕円曲線パラメータについては、SECGなどを参照せよ、と記述がされているのみであり、仕様書としての自己完結性に欠けるとの指摘がある。

## 参考文献

- [1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, “Relations Among Notions of Security for Public-Key Encryption Schemes,” Proc. of Crypto’98, Springer-Verlag, LNCS 1462, pp.26-45 (1998).
- [2] Victor Shoup : “A Proposal for an ISO Standard for Public Key Encryption,” Available at <http://shoup.net/papers/>





## 第 3 章

# 共通鍵暗号技術の評価

### 3.1 評価方法

#### 3.1.1 共通鍵暗号の評価方法

##### 3.1.1.1 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はなく、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 Challenge III (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。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。3.2.1 節に 64 ビットブロック暗号、3.2.3 節に 128 ビットブロック暗号に関する安全性評価について、3.2.5 節にストリーム暗号に関する安全性評価について記す。

### 3.1.1.2 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

- 構成部品ごとに評価を行い確率の上界を求める方法
- 計算機探索より求める方法

などによって見積もられる。

**■差分解読法・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、ラウンド関数の最大差分確率が  $p$  であるとき、段数が 4 段以上で構成されていれば、その暗号全体の最大差分確率は  $2p^2$  以下であることを数学的に証明した。その後、線形攻撃に対しても同様の指標を与え、差分解読法/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  $KM$  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、ラウンド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

- ラウンド関数単体
- データランダム化部
- 鍵スケジュール部単体
- 鍵スケジュール部を含む暗号化処理全体

の評価を行なった。

評価項目

アバランシュ性の評価項目として取り上げたものは

- 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  $\Delta X$ 、鍵差分  $\Delta K$  のハミング重みは  $m = 1, 2$  について実施。
- AVD (差分の拡散値)  
出力差分値のハミング重みの平均値。
- CC (差分出現状態の相関係数)  
出力差分値の  $i$  ビット位置と  $j$  ビット位置での相関係数。
- UKV (有効鍵量)  
ハミング重み 1 の鍵差分値を与えた時の AVA のうち相対基準値を満たしている評価値の割合。

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

### 3.1.1.3 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、ブロック暗号と同様に評価した。

### 3.1.1.4 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系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表 3.1 に示す。

表 3.1: ソフトウェア実装評価に使用した環境

PC 環境	
CPU	Pentium III (650MHz)
OS	Windows98 SE
搭載メモリ	64MB
コンパイラ	Visual C++ Ver6.0 SP3
サーバ環境	
CPU	Ultra SPARC Ili (400MHz)
OS	Solaris 7
搭載メモリ	256MB
コンパイラ	Forte C 6
ハイエンド環境	
CPU	Alpha 21264 (463MHz)
OS	Tru64 UNIX V5.1
搭載メモリ	512MB
コンパイラ	DEC C

ブロック暗号に関しては、

- データランダム化部
- 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いため、データランダム化部の測定のみを行った。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。

### 3.1.1.5 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、2000 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バイスとしては、0.25~0.35 $\mu\text{m}$  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、あくまでも参考値でしかあり得ない。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.)

## 3.2 総評

### 3.2.1 64 ビットブロック暗号

対象は、CIPHERUNICORN-E、Hierocrypt-L1、MISTY1 及び Triple DES の 4 種類である。2001 年度は、本カテゴリに対する新たな応募はなかった。CIPHERUNICORN-E から MISTY1 までの応募があり、Triple DES はその他評価が必要な暗号として 2000 年度から評価対象として追加した。評価概要を表を含め以下に示す。記述内容は以下の通り。

■**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攪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** 本評価は 2000 年度に行われた。暗号は安全面だけでなく、使用状況を想定し実装面も考慮する必要がある。電子政府における暗号実装に対する要求事項は現在のところ不明であるが、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** 本評価は 2000 年度に行われた。評価のためのターゲットデバイスとしては、 $0.25\sim 0.35\mu\text{m}$  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、あくまでも参考値でしかあり得ない。従って実装者の経験に左右されるところが大きい。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、評価結果を表 3.2 に示す。

### 3.2.2 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  $2^{-48}$  以下が保証されている。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。

表 3.2: 総評

UNI-E	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 $8 \times 8$ 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。
	総合評価 安全性について、今のところ問題は見つかっていない。処理速度は遅いグループである。
HC-L1	特徴
	東芝 (2000) 入れ子型 SPN 構造 6 段。各段は XS 関数の 2 並列及び 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 $8 \times 8$ 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。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。
	総合評価 安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
MISTY1	特徴
	三菱 (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 $7 \times 7$ の及び $9 \times 9$ 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。
	総合評価 安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
Triple DES (3-Key)	特徴
	IBM (1979) DES を 3 回繰り返した組み合わせ暗号。DES は、1977 年 FIPS で規格化。DES は、Feistel 型 16 段。S-box は $6 \times 4$ 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、FIPS 等で保証されている間は、問題ないと考える。

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。



CIPHERUNICORN-E はその段関数の複雑な構造のため、解析が難しい。自己評価書では、特性差分/線形確率はラウンド関数を簡略したものをを用いて見積もられている。2001 年度は、この簡略化の正当性を検証し、適切な見積もり方を検討した。本評価は、評価委員会だけでなく、国内 2 者と海外 2 者に評価を依頼し、さらに詳細な検討を行った。その結果、ラウンド関数の評価については自己評価書に記載されている結果と異なる結果が導かれたが、仕様段数である 16 段の CIPHERUNICORN-E は線形/差分攻撃に対して十分な安全性を持つと考えられる、という結論が得られた。

64 ビットブロック暗号のこれら特性確率が  $2^{-64}$  以下になる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の安全性保証という。以上のように、評価対象となった 64 ビットブロック暗号は全て、学術的な線形攻撃/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。FL 関数なしの変形 MISTY1 の場合、7 階の高階差分攻撃により、平文組数  $2^{11}$ 、計算量  $2^{93}$  で 6 段まで攻撃可能である。MISTY1 ではそれが、平文組数  $2^{22}$ 、計算量  $2^{33}$  で 5 段までとなる。

補間攻撃 (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 8 ビット単位の小ブロック 8 個 (64 ビットブロック暗号) に区切り、その小ブロックをガロア体  $GF(2^8)$  の多項式基底で表現した場合について線形和攻撃に対する耐性を評価した。いずれの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。

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** 本評価は2000年度に行われた。「鍵スケジュール部を含む暗号化処理全体」では、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、Hierocrypt-L1、MISTY1で期待値を満たさない部分を検出した。一方、「ラウンド関数部単体」でも、Hierocrypt-L1、MISTY1で期待値を満たさない部分を検出した。

表 3.3: アバランシュ性評価

UNI-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。
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** 本評価は2000年度に行われた。記載されている数値は、2000年度に測定された数値である。

■**データランダム化部** 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変更(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

この結果から、PC環境においては、Triple DESを比較対象とすると、CIPHERUNICORN-Eが遅いグループに分類され、残りは十分速いグループに属すると言える。暗号化と復号で若干の速度差が見られる暗号もあるが、実装に於いて問題となるほどのものではないと判断できる。また、平均値と最速値が著しく乖離している暗号も見られないので、応募暗号はPC環境において安定して動作することが期待できる。

### 2. サーバ環境

この結果から分かることは、CPUスペックの向上がそのまま直に暗号の処理速度向上に結びつかない場合があることである。Hierocrypt-L1は、応募者が測定プログラムを改変した場合の値が欄の下段に記載されている。メモリ確保を効率化することにより1割程度の速度向上が見られる。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

表 3.4: PC 環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

64 ビットブロック暗号	速度 [Mbps]
CIPHERUNICORN-E	29.0(28.9) / 29.3(29.2)
Hierocrypt-L1	209.0(207.0) / 203.9(202.2)
MISTY1	195.3(193.8) / 200.0(197.8)
Triple DES	48.7(48.6) / 48.7(48.6)

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。

表 3.5: サーバ環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

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)

### 3. ハイエンド環境

Alpha 21264 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。

表 3.6: ハイエンド環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

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)

#### 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  $\mu\text{sec}$  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  $\mu\text{sec}$  で処理が終了することが望ましい。

表 3.7: PC 環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

64 ビットブロック暗号	[ $\mu$ sec]
CIPHERUNICORN-E	3.72(3.73) / 3.70(3.72)
Hierocrypt-L1	0.58(0.58) / 0.95(0.95)
MISTY1	0.55(0.55) / 0.54(0.54)
Triple DES	3.02(3.03) / 3.03(3.04)

表 3.8: サーバ環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

64 ビットブロック暗号	[ $\mu$ sec]
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)

以上の結果から、評価対象となった暗号技術は、このような実装環境において充分実用に耐える動作が期待できることが分かる。

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** 本評価は 2000 年度に行われた。記載されている数値は 2000 年度に測定された値である。

HW 実装評価の対象となった応募 64 ビットブロック暗号は、Hierocrypt-L1、MISTY1 の 2 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 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、表 3.10 の通りである。

### 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表 3.11 の通りである。

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、Hierocrypt-L1 は Triple DES の約 2.5 倍となっている。一方、ループ・アーキ

表 3.9: ハイエンド環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

64 ビットブロック暗号	[ $\mu$ sec]
CIPHERUNICORN-E	5.14(5.16) / 5.66(5.69)
Hierocrypt-L1	0.84(0.85) / 1.35(1.41) 0.83(0.84) / 1.33(1.40)
MISTY1	0.72(0.73) / 0.68(0.73)

表 3.10: HW 評価対象のパラメータ

評価対象	繰り返し段数	鍵長 (ビット)
Hierocrypt-L1	6 段	128
MISTY1	8 段	128
Triple DES (参考)	48(= 16 × 3) 段	168

表 3.11: 回路規模、クリティカルパス遅延、処理速度

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
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	*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。

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表 3.12 の通りと評価された。

表 3.12: 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

評価対象		クリティカルパス (ns)	処理速度 (Mbps)
Hierocrypt-L1	*1	70.13	912.59
MISTY1	*2	11.86	600
		24.70	288
Triple DES	*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、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、表 3.13 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 3.13: 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)
UNI-E	16/-*	0.60	0.82
HC-L1	6/3.5	4.25	3.97
MISTY1	8/5	4.07	5.57
Triple DES	48/48	1	1

\*CIPHERUNICORN-E は、学術的な解読段数がまだ知られていない。

### 3.2.3 128 ビットブロック暗号

対象は、Camellia、CIPHERUNICORN-A、Hierocrypt-3、SC2000、RC6 Block Cipher、AES (Rijndael)、SEED の 7 種類である。Camellia から RC6 までの応募があり、AES はその他評価が必要な暗号として 2001 年度から評価対象として追加した。SEED は暗号技術検討会からの依頼で評価を行った。評価概要を表を含め以下に示す。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攪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** 本評価は SEED の評価以外は 2000 年度に行われた。暗号は安全面だけでなく、使用状況を想定し実装面も考慮する必要がある。電子政府にお

ける暗号実装に対する要求事項は現在のところ不明であるが、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** 本評価は 2000 年度に行われた。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、 $0.35\mu\text{m}$  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、評価結果を表 3.14~3.16 に示す。

表 3.14: 総評 (1/3)

Camellia	特徴
	NTT、三菱 (2000) Feistel 型、18 段 (128 ビット鍵)、24 段 (192/256 ビット鍵)、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 $8 \times 8$ 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。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。
UNI-A	特徴
	NEC(2000) Feistel 型、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 $8 \times 8$ 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。
	総合評価
	今までのところ、CIPHERUNICORN-A の安全性について、学術上の課題が残ると言わざるを得ないものの、実用上の重大な問題点は見つかっていない。処理速度は遅いグループである。
Hierocrypt-3	特徴
	東芝 (2000) 入れ子型 SPN 構造 6 段 (128 ビット鍵)、7 段 (192 ビット鍵)、8 段 (256 ビット鍵)。各段は XS 関数の 4 並列及び 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 $8 \times 8$ 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。
	総合評価
	安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。

表 3.15: 総評 (2/3)

RC6	特徴
	RSA セキュリティ (1998) 32 ビット 4 block からなる変形 Feistel 20 段。段関数は、簡潔な構造。32 ビット入力、32+5 ビット出力。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 ビット語長に対する処理で、32 ビット CPU を意識した構成。ワード長、段数、鍵長の選択可能な可変パラメータ構造。RC5 の設計思想を継承。
	総合評価 安全性について、今のところ問題は見つかっていない。Pentium III 上の暗号化で最速であるが、ソフトウェア処理速度はプラットフォームに大きく依存。
SC2000	特徴
	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 ビット鍵)、22 段 (192/256 ビット鍵)。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。
	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。
AES	特徴
	米国 NIST (2000) SPN 構造 10 段 (128 ビット鍵)、12 段 (192 ビット鍵)、14 段 (256 ビット鍵)。S-box は 8×8 の 1 種類。GF(2 <sup>8</sup> ) 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 (ShiftRow)、byte 処理による 4byte 内拡散 (MixColumn) で構成。テーブル参照、EXOR、AND。SQUARE 暗号の後継。P 層設計では、活性 S-box 数の考えで評価。
	総合評価 安全性について、今のところ問題は見つかっていない。処理速度は速いグループである。

### 3.2.4 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、評価対象暗号には無い。

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



表 3.16: 総評 (3/3)

SEED	特徴
	韓国 KISA(1997) Feistel 構造 16 段。ブロックサイズおよび鍵サイズは共に 128 ビット。入力データは二つの 64 ビットブロックに分割され片方はラウンド関数 $F$ に入力、その出力は残りの 64 ビットと排他的論理和された後、2つのブロックの入替が行なわれる。 $F$ 関数の入力は 32 ビット 2つのブロックに分割、それぞれ 32 ビットの subkey と排他的論理和。次に、右側 32 ビットが左側 32 ビットに排他的論理和され、関数 $G$ と $2^{32}$ を法とする加算演算を用いた 3 段の攪拌処理を施される。 $G$ 関数は二つの 8 ビット入力、8 ビット出力の S ボックス ( $S_1$ と $S_2$ ) を用いて実現。
	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。PC 上での処理速度はやや遅いグループに属する。

がある。

特性確率の上界が活性 S-box 数評価を使って示されているのが Camellia、Hierocrypt-3、AES と SEED 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、AES 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。SEED は、最大差分特性確率が 13 段で  $2^{-192}$  と見積もられている。線形攻撃に対しては、マルチプルパスを考慮していないが、6 段以上で  $2^{-128}$  より大きい確率を持つ線形特性が見つかっていない。

CIPHERUNICORN-A は、その段関数  $F$  の複雑な構造のため、解析が難しい。自己評価書では、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。2001 年度は、この簡略化の正当性を検証し、適切な見積もり方を検討した。本評価は、評価委員会だけでなく、国内 2 者と海外 2 者に評価を依頼し、さらに詳細な検討を行った。その結果、簡略化した段関数  $mF$  による評価では、CIPHERUNICORN-A の安全性を示す根拠として乏しいことが明らかとなった。また、安全性を脅かすものではないが、弱鍵の存在が明らかとなった。このように、安全性に関する懸念材料が完全に払拭されず、理論的に安全であるという確証を得てはいないが、仕様段数である 16 段の CIPHERUNICORN-A は線形/差分攻撃に対して十分な安全性を持つと考えられる、という結論が得られた。

RC6 は構造は簡明であるが、32 ビット語長の処理が基本であり厳密な評価が難しい。しかし、その前身の RC5 に対する評価研究及び 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 AES である。Hierocrypt-3 に対しては、32 階の高階差分攻撃 (32 階の SQUARE 攻撃) を基本にする攻撃法で 128 ビット鍵に対しては 6 段中 3 段まで、192 ビット (又は 256 ビット) 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。AES についても SQUARE 攻撃 (32 階の高階差分攻撃) を適用し部分総和法を用いることで、それぞれ 128 ビット鍵に対しては 10 段中 7 段まで、192 ビット鍵に対しては 12 段中 8 段まで、256 ビット鍵に対しては 14 段中 8 段までが全数探索より効率よく解読可能である。AES に対する、これらの攻撃方法は 128 ビットブロック暗号で生成可能な平文組数  $2^{128}$  とほぼ同等である  $2^{128} - 2^{119}$  個の平文組を必要とする。256 ビット鍵については、関連鍵攻撃を用いることでさらに 14 段中 9 段までが全数探索より効率よく解読できる。

Camellia に関しては、制御型高階差分攻撃により 128 ビット鍵に対して 18 段中 8 段まで、256 ビット鍵に対して 24 段中 10 段まで攻撃が可能である。

SEED に関しては、SQUARE 攻撃で 6 段程度まで攻撃の可能性がある。

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 ビット鍵に対しては 12 段まで、192 ビット鍵に対しては 14 段まで、256 ビット鍵に対しては 15 段まで全数探索より効率よく解読可能である。その他、不能差分攻撃、ブーメラン攻撃、mod n 攻撃、非全単射攻撃等について、現在のところ、どの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** 本評価は 2000 年度に実施された。「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、SC2000 で期待値を満たさない部分を検出した。一方、「ラウンド関数単体」では、Camellia、Hierocrypt-3、RC6、SC2000 で期待値を満たさない部分を検出した。

表 3.17: 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。
UNI-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。
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 ビットおよび 256 ビットの際に特徴が見られる。

**■ソフトウェア (SW) 実装評価** 本評価は SEED の測定以外は 2000 年度に行われた。記載されている数値は、2000 年度に測定された数値である。

### データランダム化部

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

最終段に比較のため Triple DES の測定値を記載した。Triple DES は 64 ビットブロック暗号である。この結果から、PC 環境においては、Triple DES を比較対象とすると、CIPHERUNICORN-A と SEED 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ないと判断できる。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。

表 3.18: PC 環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

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)
SEED	98.3(95.9) / 98.3(95.7)
Triple DES	48.7(48.6) / 48.7(48.6)

### 2. サーバ環境

この結果から分かることは、CPU スペックの向上がそのまま直に暗号の処理速度向上に結びつかない場合があることである。例えば、PC 環境に於いて最速の 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

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

表 3.19: サーバ環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

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)

Camellia が最速である。Alpha 21264 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。

表 3.20: ハイエンド環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

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)

#### 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  $\mu\text{sec}$  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  $\mu\text{sec}$  で処理が終了することが望ましい。

以上の結果から、このような実装環境において、十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。

表 3.21: PC 環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

128 ビットブロック暗号	[ $\mu\text{sec}$ ]
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)
SEED	1.90(1.93) / 1.90(1.93)
Triple DES	3.02(3.03) / 3.03(3.04)

表 3.22: サーバ環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

128 ビットブロック暗号	[ $\mu\text{sec}$ ]
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)

■**ハードウェア (HW) 実装評価** 本評価は 2000 年度に行われた。記載されている数値は 2000 年度に測定された値である。

HW 実装評価の対象となった 128 ビットブロック暗号は、Camellia、CIPHERUNICORN-A、Hierocrypt-3、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、2000 年度の応募時点では、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、RC6 の 3 方式であり、ループ・アーキテクチャを採用したアルゴリズムは 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、表 3.24 の通りである。

### 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表 3.25 の通りである。

表 3.23: ハイエンド環境 (暗号化: 最速値 (平均値) / 復号: 最速値 (平均値))

128 ビットブロック暗号	[ $\mu\text{sec}$ ]
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)

表 3.24: HW 評価対象のパラメータ

評価対象	繰り返し段数	鍵長 (ビット)
Camellia	24 段	256
CIPHERUNICORN-A	16 段	128
Hierocrypt-3	6 段	128
RC6	20 段	128
SC2000	19 段	128
AES (参考)	10 段	128

表 3.25: 回路規模、クリティカルパス遅延、処理速度

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
Camellia	*2	16,327	22,755	266	39,348
		9,668	13,304	141	23,124
Hierocrypt-3	*1	538,078	106,302	-	724,380
RC6	*1	77,785	975,391	-	1,753,076
SC2000 (参考)	-	-	-	-	62,000
AES (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、AES は Triple DES の約 4.1 倍と小型であり、RC6 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 (256 ビット鍵) 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 (256 ビット鍵)、Hierocrypt-3 と AES は、回路規模的には小型の部類であり、RC6 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。

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表 3.26 の通りと評価された。

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 AES は Triple DES の約 4 倍を越え、RC6 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 (0.25 $\mu$ m 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。

表 3.26: クリティカルパス遅延と想定される処理速度

評価対象		クリティカルパス (ns)	KeySetup(ns)	処理速度 (Mbps)
Camellia (256 ビット鍵)	*2	5.46	-	837
		11.51	-	397
Hierocrypt-3	*1	75.55		1,694.24
RC6	*1	698.05	2,112.26	183.36
SC2000 (参考)	-	-	-	914
AES(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。

■**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。256 ビット鍵仕様における学術的な解読可能段数と実際の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、表 3.27 に示す。なお、速度は、128 ビット鍵仕様の暗号化と復号の最速値の平均である。基準となる Triple DES の 128 ビットデータ処理時間は、測定値より次式で換算した。128 ビットブロック暗号間の相対比較として数値を利用されたい。

$$\begin{aligned} \text{データランダム化部処理時間 (128 ビット)} &= \text{データランダム化部処理時間 (64 ビット)} \times 2 \\ \text{鍵スケジュール込み処理時間 (128 ビット)} &= \text{鍵スケジュール込み処理時間 (64 ビット)} \\ &\quad + \text{データランダム化部処理時間 (64 ビット)} \end{aligned}$$

### 3.2.5 ストリーム暗号

#### 3.2.5.1 スクリーニング評価

■**評価対象** 2001 年度の以下の応募暗号技術に対しスクリーニング評価を行った。

- C4-1
- FSAnGo
- MUGI

■**評価内容** 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。

表 3.27: 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	攻撃法	速度	速度
			(データランダム化部)	(鍵スケジュール部込み)
AES	14/8 14/9	SQUARE 攻撃 関連鍵攻撃	2.15	1.23 注 1
Camellia (w/o FL)	24/10	高階差分攻撃	5.24	6.00
UNI-A	16/*	未 注 2	1.02	0.59
Hierocrypt-3	8/3.5	SQUARE 攻撃	4.12	2.73
RC6	20/15	カイ 2 乗攻撃	6.57	1.73
SC2000	22/13	差分攻撃	4.29	3.49
SEED	16/7	差分攻撃	2.02	2.29
(Triple DES)	48/48	中間一致攻撃	1	1

注 1. 参考値 Pentium III 600MHz, C, 文献 Lawrence E. Bassham, “Efficiency Testing of ANSI C Implementations of Round 2 Candidate Algorithms for the Advanced Encryption Standard,” AES3 conference, 5.1 節, Table 6 (128Blocks)

注 2. 学術的な解読段数がまだ知られていない。

- 記述の有無、記述内容の論理的整合性/自己完結性の確認。
- 書面上で容易に判明するような欠点の検査。
- 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の確認。

## ■評価結果

**C4-1** 第三者実装が可能と考えられるだけの十分なアルゴリズム情報が記載されていない。参照プログラムには応募暗号本体の記述がない。

**FSAnGo** 評価に必要な、参照プログラム、テストベクタ生成プログラムがない。

**MUGI** 安全性について、いまのところ問題は見つかっていない。1998年に発表された PANAMA の改良であるとはいえ、発表されてから日が浅く、さらなる安全性及び実装性の評価が必要と考えられる。

### 3.2.5.2 継続評価

対象は MULTI-S01 である。

**■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 ビット) から鍵ストリームを生成する。この鍵ストリームを用いてメッセージを暗号化する。メッセージ秘匿だけでなく、メッセージ認証を同時に達成する点が特徴である。MULTI-S01 は擬似乱数生成器として PANAMA を用いている。

MULTI-S01 は擬似乱数生成器 PANAMA と mode 部で構成される。これら構成部分について以下の評価を行った。



**評価内容 1[MULTI-S01 の安全性に関する検証]**

MULTI-S01 のデータランダム化部は、ブロック暗号の Modes of Operation と似た構造である。NIST 主催の Modes of Operation Workshop で見られるような Mode に対する安全性評価手法を用い、MULTI-S01 のデータランダム化部の安全性を検証する。

**評価内容 2[擬似乱数生成器 PANAMA の安全性に関する検証]**

MULTI-S01 の安全性は PANAMA に負うところがあるが、その安全性は十分検証されているとは言い難い。暗号用擬似乱数生成器に対して提案されている様々な攻撃法を適用し、PANAMA の安全性を検証する。

**評価内容 3[擬似乱数生成器 PANAMA の統計的性質の検証]**

PANAMA の乱数性は十分検証されているとは言い難い。2001 年度評価では FIPS-140 に記載されている暗号用乱数として最低限必要な評価を行った。一方 NIST SP 800-21 には、暗号用乱数に対する、更に詳細な評価方法が記載されている。この評価を PANAMA に対し適用し、統計的性質の検証を行う。

本評価は評価委員会だけでなく、それぞれ国内外併せて 3 者程度の研究者に依頼した。その結果、以下のことが明らかになった。

- MULTI-S01 の安全性は PANAMA に帰着できる
- PANAMA の乱数検定からは特段の欠陥は見あたらない
- PANAMA の安全性について致命的な問題点は見あたらない

これらより、MULTI-S01 は「安全性について、いまのところ問題は見つかっていない。ソフトウェアによる処理速度は速いグループである。」との結論を得た。

■**ソフトウェア (SW) 実装評価** 本評価は 2000 年度に行われた。記載されている数値は、2000 年度に測定された数値である。

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場合、データランダム化部の測定のみを行った。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利用した。このため、実際の性能を十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。

表 3.28: PC 環境 (最速値 (平均値))

ストリーム暗号	速度 [Mbps]
MULTI-S01	237.7 (233.7)

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 で使用される擬似乱数生成器 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。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。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、表 3.29 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。

表 3.29: HW 実装評価結果

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
S01	18.8	1.203	19,811/42,240 ATOMs(46%)	EP20K1000E

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、妥当な回路規模で、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。

## 3.3 詳細評価対象暗号 (個別暗号) の評価

### 3.3.1 CIPHERUNICORN-E

#### 3.3.1.1 技術概要

CIPHERUNICORN-E は、1998 年に日本電気株式会社 (NEC) が開発したブロック長 64 ビット、鍵長 128 ビットの 64 ビットブロック暗号であり、NEC より提案された。暗号の基本構造は 16 段の Feistel 型暗号である [1]。この暗号の特徴は、暗号の基本となるラウンド関数での拡大鍵探索を難しくすることで安全性を高めることを意図して、本流部と一時鍵生成部とで構成される極めて複雑なラウンド関数を利用している点である。また、多くの暗号の設計方針とは異なり、ラウンド関数をブラックボックスとみなして、設計者が定めた初等統計量評価を行う暗号強度評価支援システム [2] により有意な相関関係が見出せないラウンド関数を設計することを主要な設計方針としている。その結果、ラウンド関数における初等統計量評価では、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。

### 3.3.1.2 技術仕様

ブロック長 64 ビット、鍵長 128 ビット、16 段 Feistel 型構造を採用した 64 ビットブロック暗号であり、2 段ごとに L 関数が挿入される。鍵スケジューリングは、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関係が存在しない
- 高い確率で成立する入力ビットの変化と出力ビットの変化の関係が存在しない
- 高い確率で成立する鍵ビットの変化と出力ビットの変化の関係が存在しない
- 高い確率で 0 あるいは 1 となる出力ビットが存在しない

### 3.3.1.3 その他

暗号強度評価支援システムによって同じように設計された暗号として、128 ビットブロック暗号である CIPHERUNICORN-A がある。

### 3.3.1.4 安全性評価結果

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。このため、CIPHERUNICORN-E は 2000 年度の CRYPTREC Report 2000

において継続的な評価が必要であるとの総合評価を受けた。そこで2001年度は以下の観点から安全性評価を継続的に実施した。

- 差分特性確率の観点からみた差分解読法に対する安全性
- 線形特性確率の観点からみた線形解読法に対する安全性
- その他の解読法に対する安全性

CRYPTREC Report 2000では、概ね適切な考慮に基づいてラウンド関数を構成を簡略したモデル ( $mF$  関数) で、12段以上で最大差分確率の上界が  $2^{-64}$  を下回ることが示されている。また線形特性確率においては、やはりラウンド関数を簡略したモデル ( $mF^*$  関数) で、8段でその上界が  $2^{-70.72}$  となり、 $2^{-64}$  を下回ることが示されている。さらに2001年度は4名(チーム)による評価者が差分特性確率および線形特性確率について、各々適切と考慮する手法に基づいてラウンド関数および暗号全体の評価を実施した結果、いずれも仕様段数である16段よりも小さい段数で各々の上界値が  $2^{-64}$  を十分下回るという結果が得られた。これらの評価結果はいずれも CIPHERUNICORN-E のラウンド関数に何らかの近似を施したラウンド関数に基づいて算出されたものである。しかし、多数の評価者が異なる手法による近似を利用しながらほぼ同じ安全性評価結果を得られたことから、CIPHERUNICORN-E の差分解読法や線形解読法に対する安全性は、少なくとも今回見積もられた評価結果と同程度以上であると期待される。

また、上記以外の解読法については、CRYPTREC Report 2000で示すように現時点までに問題となるような点は特に発見されていない。一方、ラウンド関数の構成上、実装攻撃に対する耐性が低い恐れがあるので、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。

以上の結論を総合すると、現在までに CIPHERUNICORN-E の安全性について問題点は見つかっていない。したがって、電子政府用の暗号として CIPHERUNICORN-E を用いた場合、安全性の面で問題となることは恐らくないと考えられる。

#### 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。

CRYPTREC Report 2000では、(1)算術加算を排他的論理和に置換、(2)Y関数は32ビットデータの上位1バイトへ入力ビットを集約する処理に置換など、おおむね適切な考慮に基づいてラウンド関数の構成を簡略化した  $mF$  関数を利用したモデルで安全性の評価を行った。その結果、少なくとも12段以上で最大差分特性確率の上界が  $2^{-64}$  を下回ることが示されている。

2001年度は4名(チーム)の評価者により以下の評価結果が得られた。

**評価者 1:** ラウンド関数の最大差分特性確率の上界が  $2^{-21}$  であり、13段での最大差分特性確率の上界が  $2^{-126}$  であると指摘した。これにより仕様段数である16段の CIPHERUNICORN-E は差分解読法では解読不可能であると示している。

**評価者 2:** ラウンド関数の最大差分特性確率の上界は自己評価書の結果と同じく  $2^{-12}$  であるが、アルゴリズム全体としての上界は  $2^{-72}$  であると指摘した。しかし、仕様段数である 16 段の CIPHERUNICORN-E は差分解読法では解読不可能であるという結論は同じである。

**評価者 3:** ラウンド関数の最大差分特性確率の上界が  $2^{-14}$  であると指摘。これにより 15 段での最大差分特性確率の上界が  $2^{-98}$  であると示している。その結果として、仕様段数である 16 段の CIPHERUNICORN-E は差分解読法では解読不可能であるという結論を示している。

**評価者 4:** ラウンド関数の最大差分特性確率の上界が  $2^{-16}$  であると示し、10 段以上あれば  $2^{-64}$  を下回ることを示している。仕様段数である 16 段の CIPHERUNICORN-E が差分解読法では解読不可能であるという結論は同じである。

以上の評価結果を総合的に判断すると、様々な異なる近似モデルでの安全性評価結果のいづれについても仕様段数である 16 段よりも小さい段数で上界値が  $2^{-64}$  を十分下回ることから、CIPHERUNICORN-E は差分解読法に対して安全であることが期待される。

#### C) 線形解読法

CRYPTREC Report 2000 では、ラウンド関数を簡略したモデル ( $mF^*$  関数) で、ラウンド関数の最大線形特性確率の上界値が  $2^{-17.68}$  となり、8 段でその上界が  $2^{-70.72}$  となり、 $2^{-64}$  を下回ることが示されている。また自己評価書では、ラウンド関数を簡略したモデル ( $mF$  関数) で、ラウンド関数の最大線形特性確率の上界値は  $2^{-63.90}$  と示されている。

2001 年度は 4 名 (チーム) の評価者により以下の評価結果が得られた。

**評価者 1:** ラウンド関数の最大線形特性確率の上界が  $2^{-24.64}$  であり、13 段での最大差分特性確率の上界が  $2^{-147.84}$  であると指摘した。これにより仕様段数である 16 段の CIPHERUNICORN-E は線形解読法では解読不可能であると示している。

**評価者 2:** ラウンド関数の最大線形特性確率の上界が  $mF$  関数を用いて  $2^{-62}$  となると指摘した。しかし、仕様段数である 16 段の CIPHERUNICORN-E は線形解読法では解読不可能であるという結論は同じである。

**評価者 3:** ラウンド関数の最大線形特性確率の上界が  $mF$  関数を用いて  $2^{-27.3}$  であると指摘した。これにより 15 段での最大線形特性確率の上界が  $2^{-191.2}$  であると示している。その結果として、仕様段数である 16 段の CIPHERUNICORN-E は差分解読法では解読不可能であるという結論を示している。

**評価者 4:** ラウンド関数の最大線形特性確率の上界が  $2^{-16}$  であると示し、10 段以上あれば  $2^{-64}$  を下回ることを示している。仕様段数である 16 段の CIPHERUNICORN-E が差分解読法では解読不可能であるという結論は同じである。

以上の評価結果を総合的に判断すると、様々な異なる近似モデルでの安全性評価結果のいづれについても仕様段数である 16 段よりも小さい段数で上界値が  $2^{-64}$  を十分下回ることから、CIPHERUNICORN-E は線形解読法に対して安全であることが期待される。

#### 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、実装攻撃に対する防御策を注意深く抗ることが望まれる。

### 3.3.1.5 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。評価結果は表 3.30 および 3.31 の通りである。

表 3.30: データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1435 / 1438	1424 / 1426
2 回目	1434 / 1444	1422 / 1425
3 回目	1436 / 1440	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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1462 / 1469	1462 / 1468
2 回目	1462 / 1468	1462 / 1468
3 回目	1462 / 1469	1462 / 1468
Alpha 21264 (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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1575 / 1583	1566 / 1579
2 回目	1575 / 1583	1568 / 1582
3 回目	1575 / 1583	1568 / 1580

表 3.31: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

Pentium III (650MHz)		
言語	ANSI C+アセンブラ	
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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	2421 / 2426	2406 / 2453
2 回目	2418 / 2428	2406 / 2424
3 回目	2420 / 2424	2410 / 2414
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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	2882 / 2892	2936 / 2944
2 回目	2882 / 2890	2935 / 2944
3 回目	2883 / 2890	2935 / 2944
Alpha 21264 (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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	2381 / 2393	2621 / 2634
2 回目	2381 / 2390	2619 / 2635
3 回目	2381 / 2390	2623 / 2634

### 3.3.1.6 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、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.

## 3.3.2 Advanced Encryption Standard (AES)

### 3.3.2.1 技術概要

AES は基本的に、1998 年に J. Daemen (Proton World International) と V. Rijmen (Katholieke Universiteit Leuven) によって AES (Advanced Encryption Standard) プロジェクトに提案された共通鍵ブロック暗号 Rijndael であり、ブロック長・鍵長ともに 128、192、256 ビットが利用可能である [1]。Rijndael は、AES での公開の議論を経て、2000 年 10 月に NIST (National Institute of Standards and Technology) によって AES winner に選定され [2]、2001 年 11 月に AES (Advanced Encryption Standard) として FIPS-197 AES (Federal Information Processing Standard-197) に制定された。Rijndael は、ブロック長・鍵長、繰り返し段数の可変なパラメータを持つが、FIPS においては、AES の仕様としてそれらを限定している。

### 3.3.2.2 技術仕様

AES の主な設計方針は、(1) 既存の攻撃法に対して十分な安全性を確保する、(2) 様々なハードウェアにおいて実装可能とする、(3) 安全性に関する分析が容易になるようにアルゴリズムの構造をシンプルにする、である。AES は SPN 型暗号で、データブロックはラウンド関数内で 8 ビット単位で変換される。アルゴリズムの段数はブロック長と鍵長に依存し、128 ビットブロックの場合は、鍵長が 128、192、256 ビットに対応して、10 段、12 段、14 段となる。ラウンド関数は三種類の変換部によって構成されており、線形変換層 (ビットシフト等)、非線形変換層 (換字変換)、拡大鍵変換層 (拡大鍵との排他的論理和) を用いて変換が行われる。鍵スケジュール部では、ブロック長と同じ長さの拡大鍵が  $(r + 1)$  個 ( $r$  は段数) 生成される。鍵スケジュール部の変換には、データランダム化部のビットシフトと換字変換が利用される。

### 3.3.2.3 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。

### 3.3.2.4 評価結果

■**安全性評価** 128 ビットブロック暗号の AES の安全性について現在まで報告されてきた公知文献等の主な評価結果をまとめると次のようになる。

- 128、192、256 ビット鍵の仕様通りの AES を解読可能な攻撃法は発見されていない。
- 128 ビット鍵の場合、10 段のうち 6 段あるいは 7 段まで解読可能な攻撃法が発見されている。
- 192 ビット鍵の場合、12 段のうち 7 段まで解読可能な攻撃法が発見されている。
- 256 ビット鍵の場合、14 段のうち 7 段、8 段あるいは 9 段まで解読可能な攻撃法が発見されている。



以上の結果、NIST は AES の報告書において Rijndael はその安全性において及第点に達している (adequate セキュリティマージンを持つ) と報告し [2]、AES として FIPS に制定した。以下、これらに関してもう少し詳しく述べる。

(1) AES 提案時の提案者による自己評価報告

Rijndael の提案者は AES への提案時に、Rijndael の差分解読法、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示し、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、Rijndael には適用困難であると示している。

(2) AES 提案後の安全性評価結果

AES に提案後、多くの研究者によって Rijndael の安全性に関する研究報告が行われた。それらのうち主なものを以下に示す。

- Collision attack の適用で、192 ビット鍵および 256 ビット鍵の Rijndael の場合には、 $2^{32}$  の選択平文を用いて 7 段まで解読可能であることが報告されている [5]。
- SQUARE 攻撃を 192 ビット鍵および 256 ビット鍵に適用することで、 $2^{32}$  の選択平文を用いて 7 段の Rijndael が解読可能であることが報告されている [6]。
- SQUARE 攻撃を改良し、128 ビット鍵の場合は 7 段まで、256 ビット鍵の場合は 8 段まで解読可能な攻撃法が報告されている [7]。ただしこの解読法に必要な選択平文数は、ほぼ全数にあたる  $2^{128} - 2^{119}$  となっている。
- 鍵関連攻撃によって 256 ビット鍵の Rijndael が 9 段まで解読可能であると報告されている [7]。

以上のように、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]。

■ **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 Pentium III 上での C 言語による実装評価 [8] を示す。なお、以下の評価結果にある鍵セットアップ時間は暗号化または復号時間を含まないことに注意せよ。

〈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 ビット鍵 1289[cycles]

192 ビット鍵 2000[cycles]

256 ビット鍵 2591[cycles]

復号鍵セットアップタイム:  
 128 ビット鍵 1724[cycles]  
 192 ビット鍵 2553[cycles]  
 256 ビット鍵 3255[cycles]

暗号化 (ECB) 速度:  
 128 ビット鍵 805[cycles]  
 192 ビット鍵 981[cycles]  
 256 ビット鍵 1155[cycles]

復号 (ECB) 速度:  
 128 ビット鍵 784[cycles]  
 192 ビット鍵 955[cycles]  
 256 ビット鍵 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 3.32: 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 ビット鍵	237	1276	805	362	7770
192 ビット鍵			981	428	
256 ビット鍵			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 Pentium III 600MHz, C.Ref. [8], 5.1, Table 6 (128blocks)  
 D: Intel Pentium II/III, C. Source: Ref. [12], Table 1.  
 E: Ultra SPARC-I, W/JDK1.2, JIT, Java. Ref. [13], Table 2.

表 3.33: 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 ビット鍵	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 3.34: 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 ビット鍵	9464	25494

J: Motorola 6805 CPU Core, C. Ref. [17], Table 3.  
 K: Z80 CPU+coprocessor. Ref. [18], Table 8.

表 3.35: 32 ビットプロセッサ (復号: C 言語)

	B	C	D
	cycles	cycles	cycles
128 ビット鍵	1276	784	358
192 ビット鍵		955	421
256 ビット鍵		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 3.36: 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 ビット鍵	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 3.37: 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 ビット鍵	17742(18886)	1289(1724)	215(1334)
192 ビット鍵		2000(255 3)	215(1591)
256 ビット鍵		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 3.38: 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 ビット鍵	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 3.39: 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 ビット鍵	10318

K: Z80 CPU+coprocessor. Ref. [18], Table 8.

■**ハードウェア (HW) 実装評価** 2001年度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 0.35 $\mu$  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, 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 Technology, Gaithersburg, MD, April 13-14, 2000, pp.279-285.
- [10] K. Aoki and H. Lipmaa, Fast Implementations of AES Candidates, in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14, 2000, pp.106-120.

- [11] E. Biham, A Note on Comparing the AES Candidates , in The Second AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, March 22-23, 1999, pp.85-92.
- [12] B. Gladman, AES Second Round Implementation Experience, AES Round2 public comment, May 15, 2000
- [13] O. Baudron, et al., Report on the AES Candidates, in The Second AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, March 22-23, 1999, pp.53-67.
- [14] J. Worley, et al., AES Finalists on PA-RISC and IA-64: Implementations & Performance, in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburgs, MD, April 13-14, 2000, pp.57-74.
- [15] J. Worley, E-mail comments, AES Round 2 public comment, May 15, 2000, available at AES home page.
- [16] R. Weiss and N. Binkert, A comparison of AES Candidate on the Alpha 21264, in The Th ird 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

### 3.3.3 CIPHERUNICORN-A

#### 3.3.3.1 技術概要

CIPHERUNICORN-A は、2000年に日本電気株式会社 (NEC) が開発したブロック長 128 ビット、鍵長 128、192、256 ビットの 128 ビットブロック暗号 [1] であり、NEC より応募された。この暗号の基本構造は 16 段の Feistel 型暗号である。

最大の特徴は、暗号の基本となるラウンド関数での副鍵探索を難しくすることによって安全性を高めることを意図して、本流部 (main stream) と一時鍵生成部 (temporary key generation) とで構成される極めて複雑なラウンド関数を利用している点である。また、ラウンド関数をブラックボックスとみなし、設計者が定めた初等統計評価を行う暗号強度評価支援システム [2] によって有意な相関関係が見出せないように、ラウンド関数を設計することを主要な設計方針としており、この点が最近の多くの主要な暗号の設計方針と大きく異なる。

応募者によれば、ラウンド関数における初等統計評価においても、すべての評価項目でデータ攪拌の偏りは検出されなかったとしている。実装面では、32 ビットプロセッサ上でより高速に処理できるように設計したと述べている。

#### 3.3.3.2 技術仕様

ブロック長 128 ビット、鍵長 128、192、256 ビット、16 段 Feistel 型構造を採用した 128 ビットブロック暗号であり、AES と同じインタフェースを有する。鍵スケジュール部では、秘密鍵を攪拌しながら、2304 ビット分の副鍵 (32 ビット副鍵 72 個) を生成する。

### ■ラウンド関数 (データランダム化部)

- 4 個の 32 ビット副鍵 (関数鍵とシード鍵各 2 個) を用いた 64 ビット入出力関数であり、4 個の S-box (T 関数)、32 ビット算術加算、32 ビット定数算術乗算およびローテーション (A3 関数) の組合せにより構成される。
- 全単射関数ではない。
- 64 ビットの入力データは本流部と一時鍵生成部に分岐し、関数鍵は本流部に、シード鍵は一時鍵生成部にそれぞれ入力される。
- 一時鍵生成部では入力データとシード鍵から一時鍵が生成される。
- 生成された一時鍵は本流部に挿入され、最終的に 64 ビットの出力データが得られる。また、本流部の構成の一部は一時鍵の値によって変化するデータ依存関数となっている。

### ■鍵スケジュール部

- MT 関数をラウンド関数とする拡張 Feistel 型構造をしており、秘密鍵を攪拌しながら、各 MT 関数から 32 ビットの間接鍵を出力する。
- MT 関数は、ラウンド関数と同じ T0 関数および 32 ビット定数算術乗算の組合せにより構成される。
- 72 個の間接鍵を生成した後、その順番を入れなおして各段における副鍵とする。

■設計方針 差分解読法や線形解読法はラウンド関数でのデータ攪拌の偏りを利用して鍵情報 (副鍵) を推定することから、ラウンド関数においてデータ攪拌の偏りが検出できない構造にすることを CIPHERUNICORN-A の本質的な設計方針としている。そこで、ラウンド関数をブラックボックスとみなして初等統計評価を行う暗号強度評価支援システムを利用して、以下の条件を満たすようにラウンド関数の設計を行っている。

- 高い確率で成立する入力ビットと出力ビットの関係が存在しない。
- 高い確率で成立する出力ビット間関係が存在しない。
- 高い確率で成立する入力ビットの変化と出力ビットの変化の関係が存在しない。
- 高い確率で成立する鍵ビットの変化と出力ビットの変化の関係が存在しない。
- 高い確率で 0 あるいは 1 となる出力ビットが存在しない。

#### 3.3.3.3 その他

暗号強度評価支援システムによって同じように設計された暗号として、64 ビットブロック暗号である CIPHERUNICORN-E がある。

#### 3.3.3.4 評価結果

■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はじめとする、理論的な解読技術に対する安全性を正確に

評価・解析することは困難である。このため、CRYPTREC Report 2000 での継続的な評価が必要との総合評価を受け、2001 年度は CIPHERUNICORN-A に対する 3 段消去攻撃を想定し、差分解読法や線形解読法に対して 13 段で十分な安全性を有しているかという観点から安全性評価を継続的に実施した。

CRYPTREC Report 2000 では、おおむね適切な考慮に基づいてラウンド関数の構成を簡略化した mF 関数を利用したモデルでは、少なくとも 15 段以上で最大差分特性確率の上界が、また 14 段以上で最大線形特性確率の上界がそれぞれ  $2^{-128}$  を下回ることが示されている。さらに、2001 年度は、応募者ならびに 4 人 (チーム) による評価者が、各々独自に適切と考慮する手法に基づいてラウンド関数および暗号全体の評価を実施した結果、一部の評価を除き、いずれも 13 段における最大差分特性確率の上界は  $2^{-100}$  以下、最大線形特性確率の上界は  $2^{-128}$  前後と見積もられている。これらの評価結果はいずれも CIPHERUNICORN-A のラウンド関数そのものではなく、何らかの近似を施していたラウンド関数に基づいて算出されたものである。しかし、多数の評価者が異なる手法による近似を利用していながらほぼ同じ安全性評価結果が得られたことから、CIPHERUNICORN-A の差分解読法や線形解読法に対する安全性は、少なくとも今回見積もられた評価結果と同程度以上であると期待される。したがって、3 段消去攻撃を想定した場合の差分解読法や線形解読法に対して、学術的に攻撃不可能であるとまでは証明されないものの、現実にはほぼ不可能であろうと推定される。

また、上記以外の解読法については、CRYPTREC Report 2000 で示すように現時点までに問題となるような点は特に発見されていない。一方、ラウンド関数の構成上、実装攻撃に対する耐性が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。

加えて、安全性に関する新しい指摘として、すべての副鍵の値が (秘密鍵の鍵長に関わりなく) 秘密鍵の上位 32 ビットと同一になるという、非自明と考えられる弱鍵が少なくとも一つ存在することが示された。もっとも、現時点では、 $2^{128}$  個の秘密鍵 (128 ビット秘密鍵の場合) のうちのひとつが弱鍵として指摘されているだけであるので、この指摘だけで安全性に重大な問題が生じたということではない。

以上の結論を総合すると、今までのところ、CIPHERUNICORN-A の安全性について、学術上の観点からは課題が残ると言わざるを得ないものの、実用上の重大な問題点は見つかっていない。したがって、電子政府用の暗号として用いた場合に実用上は問題となることはおそらくないと考えられる。

#### ■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評価のすべての項目について良好な結果を得ているなど、乱数性に関してはおおむね良好と判断される。ただし、データ攪拌偏りが検出できないようにラウンド関数を設計したとしているが、このように設計されたラウンド関数がランダム関数とほぼ同じ特性をもつことを意味しているわけではない。例えば、自己評価書では本流部、一時鍵生成部のどちらか一方でも十分な攪拌が行われていると述べているが、入力データや鍵の値によっては高い確率で複数個の T 関数の効果が打ち消しあい、どちらか一方だけでは十分な攪拌が行われていない場合があるとの指摘もある。

##### 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、適切な考慮に基づいてラウンド関数を簡略化した暗号モデルを考え、そのモデル上での安全性を議論することがある。これは、実際の暗号が適切な考慮に基づく簡略化モデルでの安

全性と同程度以上の安全性を有していると一般に期待されるためである。

CRYPTREC Report 2000 では、(1) 算術加算を排他的論理和に置換、(2) 定数乗算は 32 ビットデータの上位 1 バイトへ入力ビットを集約する処理に置換、(3) A3 関数は truncated vector 単位でのローテーション処理に置換、などによって簡略化した mF 関数を利用したモデルで安全性の評価を行った。その結果、少なくとも 15 段以上で最大差分特性確率の上界が  $2^{-128}$  を下回ることが示されている。

2001 年度は別の観点による近似手法に基づく評価を以下のように実施した。

**評価者 1:** mF 関数を利用したモデルでの安全性を再評価し、その結果、定数乗算の近似処理が不完全であったことを発見した。また、この近似処理を完全に行った場合、mF 関数での最大差分特性確率の上界が  $2^{-7}$ 、13 段での最大差分特性確率の上界が  $2^{-56}$  までしか示せない指摘した。ただし、本来、定数乗算は入力データに依存して差分特性確率になんらかの影響を与え、安全性向上に寄与すると期待されるが、ここでは定数乗算の近似処理において差分特性確率に影響を与えないもの(設計者に対して不利な評価)として安全性の評価を実施していることに注意を要する。

**応募者:** 暗号技術評価ワークショップのランプセッションで応募者が発表した新しい安全性自己評価に関して、さらに詳細な報告を検討する必要があると認められたため、応募者に対して追加レポートの提出を要求した。この追加レポートによれば、定数乗算による差分特性確率への影響度を実験的に調査(継続中)しており、評価者 1 が指摘したようなケースにおける定数乗算では差分特性確率に対して  $2^{-6}$  の影響を少なくとも与えるとしている。また、mF 関数での最大差分特性確率の上界が  $2^{-13}$ 、13 段での最大差分特性確率の上界が  $2^{-104}$  となるとも述べている。この新しい評価結果に関して検討した結果、評価者 4 も定数乗算の効果を  $2^{-7}$  と見積もっていることなどを考慮すると、ここでの評価結果は妥当なものと考えられる。

**評価者 2:** 本流部のみで構成されるラウンド関数とした時のモデルに対し、6 段繰返し表現(最大差分特性確率  $2^{-56}$ )によって安全性評価を実施している。この結果、13 段での最大差分特性確率の上界が  $2^{-119}$  となることを示している。

**評価者 3:** A3 関数および定数乗算の効果を完全に除外した場合について安全性評価を行っている。その結果、ラウンド関数での最大差分特性確率の上界が  $2^{-14.4}$ 、13 段での最大差分特性確率の上界が  $2^{-115.2}$  となることを示している。

**評価者 4:** T 関数での効果のほかに、本流部の算術加算と A3 関数の(実験的に調査した)効果および一時鍵生成部の定数乗算の効果をそれぞれ加味すると、本流部、一時鍵生成部、ラウンド関数の最大差分特性確率の上界はそれぞれ  $2^{-14}$ 、 $2^{-7}$ 、 $2^{-21}$  となる。これより、13 段での最大差分特性確率の上界が  $2^{-126}$  となることを示している。

以上の評価結果を総合的に判断すると、様々な異なる近似モデルでの安全性評価いづれについても 13 段での最大差分特性確率の上界が  $2^{-100}$  以下と見積もられ、また実際の CIPHERUNICORN-A についても同程度以上の安全性を有するであろうと期待される。したがって、3 段消去攻撃を想定した場合の差分解読法に対して、学術的に攻撃不可能であるとまでは証明されないものの、現実にはほぼ不可能であろうと推定される。

#### c) 線形解読法

ここでは、いづれの評価者も mF 関数を利用したモデルをベースに評価を実施している。

**評価者 1:** ラウンド関数での最大線形特性確率の上界が  $2^{-21.37}$ 、13 段での最大線形特性確率の上界が  $2^{-128.2}$  となることを示している。

**評価者 3:** ラウンド関数での最大線形特性確率の上界が  $2^{-21.68}$ 、13 段での最大線形特性確率の上界が  $2^{-130.1}$  となることを示している。

**評価者 4:** 応募者による S-box の最大線形特性確率の評価が正しいと仮定した場合、



ラウンド関数での最大線形特性確率の上界が  $2^{-13.9}$ 、13 段での最大線形特性確率の上界が  $2^{-83.4}$  となることを示している。なお、評価者 4 の検査では、応募者の評価と矛盾する結果が出ており、ラウンド関数での線形特性確率の上界が今回の評価よりも高くなる可能性を否定していない。その一方、A3 関数、定数乗算および一時鍵生成部の影響をほとんど考慮していないことも合わせて注意を要する。そのため、実際には差分解読法に対する耐性よりも強いと期待されるとも述べている。

以上の評価結果を総合的に判断すると、線形解読法に対する耐性は差分解読法に対する耐性よりも強いと期待され、具体的な評価としては 13 段での最大差分特性確率の上界が  $2^{-128}$  程度以下と推定される。したがって、3 段消去攻撃を想定した場合の線形解読法による攻撃はほぼ不可能であろうと考えられる。

- 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。

### ■鍵スケジュール部に対する安全性

- a) 弱鍵存在の指摘

鍵スケジュール部における中間鍵の取り出しを以下のように行っている。ここで、すべてのシンボルは 32 ビットデータを表すものとし、128 ビット鍵は  $(A, B, C, D)$ 、192 ビット鍵は  $(A, B, C, D, E, F)$ 、256 ビット鍵は  $(A, B, C, D, E, F, G, H)$  を入力とする。

入力:  $(A, B, C, D, \dots, y)$   
以下を指定回数繰り返す  
 $(A^*, B^*) \leftarrow MT(A, B)$   
 $A \leftarrow B^*, B \leftarrow C, C \leftarrow D, \dots, y \leftarrow A^*$   
指定箇所で中間鍵出力:  $A$

この鍵スケジュールでは、入力が  $(A, B, B, B, \dots, B)$  であるとき、もし  $(B, A) \leftarrow MT(A, B)$  を満たすならば、繰り返し中のデータは (何回繰り返そうと) 常に  $(A, B, B, B, \dots, B)$  のままである。つまり、どの指定箇所での中間鍵もすべて  $A$  となり、中間鍵生成のための鍵スケジュールが実効的にまったく作用していない状態となる。

このような条件を満たす入力を計算した結果、 $A = 0x61db99c8, B = 0x9f3d61c8$  のときに  $(B, A) \leftarrow MT(A, B)$  を満たすことが判明した。つまり、秘密鍵が  $(0x61db99c8, 0x9f3d61c8, 0x9f3d61c8, 0x9f3d61c8, \dots, 0x9f3d61c8)$  であるとき、すべての中間鍵が秘密鍵の上位 32 ビットと同一な値  $0x61db99c8$  となる。また、副鍵は中間鍵の順番だけを入れ替えて生成することから、すべての副鍵が同じ値  $0x61db99c8$  となることをも意味する。

本来の鍵スケジュールの役割に照らし合わせ、CIPHERUNICORN-A の鍵スケジュールの構成から推測するに、この種の秘密鍵は非自明な弱鍵であると考えほうが自然である。なお、現時点で判明している弱鍵はこの一つ (一種類) だけである。

- b) 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が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、実装攻撃に対する防御策を注意深く講じることが望まれる。

### 3.3.3.5 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。評価結果は表 3.40、3.41 の通りである。

表 3.40: データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1569 / 1574	1574 / 1578
2 回目	1570 / 1574	1574 / 1577
3 回目	1570 / 1574	1574 / 1578
Ultra SPARC Ili (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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	2273 / 2282	2302 / 2326
2 回目	2273 / 2282	2309 / 2327
3 回目	2273 / 2282	2310 / 2327
Alpha 21264 (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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1834 / 1843	1769 / 1782
2 回目	1828 / 1842	1769 / 1782
3 回目	1828 / 1842	1769 / 1782

暗号化および復号処理、鍵生成まで含めた暗号化および復号処理のすべての測定項目について、今回応募された 128 ビットブロック暗号のなかで、測定プラットフォームによらずに処理速度が最も遅いグループである。また、Pentium III 上ではすべての測定項目について Triple DES と同程度である。

応募者による実装例として、Pentium III (866MHz) 上での ANSI C (インラインアセンブリ有り) による速度評価結果 [単位: cycles] が以下のように示されており、上記とほぼ同様の処理速度である。

また、IC カードを代表とする、8 ビット CPU 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。

表 3.41: 鍵スケジューラ部+データランダム化部速度測定結果 (単位 [cycles/block])

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/鍵スケジューラ含む)		
コンパイラオプション	/O2 /Oy- (実行速度) を指定		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)	
1 回目	4788 / 4822	4799 / 4931	
2 回目	4788 / 4814	4798 / 4815	
3 回目	4787 / 4830	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/鍵スケジューラ含む)		
コンパイラオプション	-v -fast		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)	
1 回目	7970 / 8160	8802 / 9025	
2 回目	7961 / 8164	8817 / 9034	
3 回目	7900 / 8161	8823 / 9028	
Alpha 21264 (463MHz)			
言語	ANSI C		
プログラムサイズ	8552 Byte (暗号化/復号/鍵スケジューラ含む)		
コンパイラオプション	-O4		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)	
1 回目	4610 / 4623	5071 / 5092	
2 回目	4610 / 4628	5071 / 5100	
3 回目	4610 / 4624	5071 / 5095	
測定項目	128 ビット鍵	192 ビット鍵	256 ビット鍵
鍵スケジューラ	3219	4032	3518
暗号化	1565	1565	1565
復号	1559	1559	1559
鍵スケジューラ+暗号化	4780	5593	5079
鍵スケジューラ+復号	4791	5604	5090

### 3.3.3.6 ハードウェア (HW) 実装評価

応募者からは以下の自己評価結果が報告されている。

0.25 $\mu$ m CMOS ASIC: 128 ビット鍵のみ利用可。

- 速度優先: 170.60Mbps, 325.3K gates
- 規模優先: 86.80Mbps, 290.4K gates

ALTERA EP20K1500EFC33-1 (FPGA): 128, 192, 256 ビット鍵利用可。

- 44.33Mbps, 7072 Logic Cell + 66ESB

## 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年1月.
- [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年1月.

## 3.3.4 SEED

### 3.3.4.1 技術概要

SEED は 1997 年の韓国政府の標準暗号アルゴリズム開発決定を受けて、Korea Information Security Agency (KISA) により設計された暗号アルゴリズムである [1]。

SEED は韓国国内で標準化が進んでおり、1999 年に TTA (Telecommunications and Technology Association) 標準 (産業標準) となり (TTA KO-12.0004)、韓国情報通信省 Ministry of Information and Communication (MIC) により推進されている政府標準 Korean Information Communication Standard (KICS) への制定作業も進んでいる [2]。

### 3.3.4.2 技術仕様

SEED はいわゆる Feistel 型の構造をもつブロック暗号であり、処理の繰り返し段数は 16 段である。また SEED のブロックサイズおよび鍵サイズは共に 128 ビットである。

各ラウンドにおいて、入力データは二つの 64 ビットブロックに分割されて、片方はラウンド関数  $F$  に入力され、その出力は残りの 64 ビットと排他的論理和された後、2 つのブロックの入替が行なわれる。

$F$  関数の入力は 32 ビット 2 つのブロックに分けられて、それぞれ 32 ビットの subkey と排他的論理和される。次に、右側 32 ビットが左側 32 ビットに排他的論理和される。さらに、関数  $G$  と  $2^{32}$  を法とする加算演算を用いた 3 段の攪拌処理を施される。

$G$  関数は二つの 8 ビット入力、8 ビット出力の S-box( $S_1$  と  $S_2$ ) を用いて実現されている。

### 3.3.4.3 評価結果

■**総合評価** 差分解読・線形解読・高階差分攻撃に関する耐性を詳細に評価した。また補間攻撃・SQUARE 攻撃・Non-surjective 攻撃・スライド攻撃に対する耐性について考察を加えた。その結果、現時点では 16 段 SEED を、鍵の総当り以上に効率よく解読する方法は見出せないと判断される。

PC 上での処理速度は、CRYPTREC で評価された暗号の中ではやや遅いグループに属する。

なお、SEED の鍵スケジュールは以下で示すような、特異な性質を持つことが判明したが、この性質が SEED の安全性に対して直ちに脅威になることはないと思われる。

#### (1) データランダム化部の安全性

##### 差分解読特性

自己評価書においては SEED の最大差分特性確率は、6 段で  $2^{-130}$  以下と評価し、この値を根拠に 6 段まで攻撃可能と結論づけている。これに対して、文献 [3] の詳細な評価結果によれば、最大特性確率は 6 段で  $2^{-124}$  であり、攻撃可能段数は 7 段である。一方、ラウンド関数内の加算演算 ( $2^{32}$  を法とする) を排他的論理和演算におきかえるという近似を施してアクティブ S-box 数に基づく分析を行なうこともできる。この場合、ラウンド毎少なくとも 4 つの S-box がアクティブで、各 S-box の特性確率は  $2^{-6}$  であるから、13 ラウンドでの最大差分特性確率は  $2^{-192}$  と見積もられる。

$2^{32}$  を法とする加算を近似しないオリジナルの仕様の場合には、正確に特性確率を見積もることは困難である。しかしラウンド数が大きければ、1 ラウンド当たりアクティブな S-box 数が 3 を下回るとはまず起こらないと考えられる。この予想の下で、最大差分特性確率は 13 ラウンドで  $2^{-144}$  以下と見積もれる。

いずれの考察結果も 16 段の SEED を差分解読法により解読するのは困難であることを示している。

##### 線形解読特性

自己評価書の解析ではマルチプルパスを考慮していないが、6 段以上で  $2^{-128}$  より大きい確率を持つ線形特性が見つかっていないので、16 段 SEED は線形解読に対し十分な安全マージンを持つと考えられる。ラウンド関数内の加算演算 ( $2^{32}$  を法とする) を排他的論理和でおきかえるという近似をした場合、13 ラウンドで最大線形特性確率は  $2^{-192}$  以下と見積もれる。加算演算を用いた場合、線形特性は排他的論理和で近似した場合よりも悪くならないと予想されるため、やはり 16 段 SEED は線形解読法によって解読するのは困難であると判断される。

##### 高階差分攻撃

ラウンド関数  $F$  は、32 ビット入出力の  $G$  関数により構成され、 $G$  関数はさらに 8 ビット入出力の全単射 S-box により構成される。S-box の代数次数は最大の 7 となることが確認される。また、 $G$  関数のすべての出力ビットは入力の 7 次のブール多項式で表される。このことをベースに  $F$  関数の代数次数を考察すると、 $F$  関数のすべての出力ビットは 63 次となる。 $F$  関数は 64 ビット入出力の全単射関数であるため、これは  $F$  が達成可能な最大の次数である。16 ラウンドの SEED 全体では充分高い代数次数が実現されると予想され、高階差分攻撃の適用は困難と考えられる。

##### 補間攻撃

SEED は異なる代数体上の演算を組み合わせられており、どのような体上での多項式または有理表現に置換えても充分複雑な表現になると予想され、補間攻撃の適用は困難と考えられる。

##### SQUARE 攻撃 (Integral 攻撃)

6 ラウンド程度までは適用可能性があるが、それ以上の段数に対しては適用不能と考えられる。

##### Non-surjective 攻撃

ラウンド関数  $F$  が全単射であるため、Non-surjective 攻撃は適用できない。

## (2) 鍵スケジュール部の安全性

## 総当たり攻撃

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 SEED で指定されている 128 ビットの場合、鍵の全数探索は現実的ではないと考えられる。

## スライド攻撃

構造が単純な鍵スケジュールに対して有効な攻撃法としてスライド攻撃が知られている。SEED の鍵スケジュール部は S-box とラウンド毎異なる定数を用いているため、スライド攻撃は有効に働かないと考えられる。

## 鍵スケジュール部の特異な性質

SEED の鍵スケジュール部は、入力された 128 ビット鍵から 16 個のラウンド鍵を生成する。各ラウンド鍵は 2 つの 32 ビットブロック (合計 64 ビット) から構成される。SEED の鍵スケジュール処理の概要を以下に示す。

- for  $i:=1$  to 16 do
  - $k_{i,0} = G(a + c - kc_i)$
  - $k_{i,1} = G(b - d + kc_i)$
  - if  $i$  odd do  $b||a = (b||a)^{>>8}$
  - else do  $d||c = (d||c)^{<<8}$ ,

ここで、 $a, b, c, d$  は鍵を 32 ビットずつ 4 ブロックに分けたもの、 $kc_i$  ( $i = 1, \dots, 16$ ) はラウンド毎の定数である。また  $k_{i,0}, k_{i,1}$  は第  $i$  ラウンドの鍵を構成する。

$k_{i,0}$  は、各段でローテートされた  $a$  および  $c$  とラウンド定数  $kc_i$  に依存する。従って、もしユーザが選んだ 2 つの鍵  $K$  と  $K^*$  が、 $a + c$  が定数となるような形をしている場合には  $K$  のラウンド鍵  $k_{i,0}$  と、 $K^*$  のラウンド鍵  $k_{i,0}^*$  は一致する。

この方針に従って、具体的値を探索する手続きを考える。 $a + c$  が全ての  $i$  について定数とすると、関係式  $(b||a) = (b||a)^{<<8} = (d||c) = (d||c)^{<<8}$  が成り立たなければいけない。32 ビットブロック  $a$  をさらに 4 つのバイトに分轄し  $a = a_0, a_1, a_2, a_3$  のように書くものとする、上記関係式が成り立つためには  $e$  を定数として次のような関係式を満たす必要がある。

- $a_i + c_j = e$  任意の  $i$  かつ  $j = 0, 1, 2, 3$ ,
- $a_i + d_j = e$  任意の  $i$  かつ  $j = 0, 1, 2, 3$ ,
- $b_i + c_j = e$  任意の  $i$  かつ  $j = 0, 1, 2, 3$ ,
- $b_i + d_j = e$  任意の  $i$  かつ  $j = 0, 1, 2, 3$

以上の考察により、次の条件を満たす (複数の) 鍵は、ラウンド鍵の半分が相互に一致することを意味する: ある値  $x$  について  $a_0 = a_1 = a_2 = a_3 = x$  かつ  $b = a$ 、またある値  $y$  に対して  $c_0 = c_1 = c_2 = c_3 = y$  かつ  $d = c$ 。

以上の条件を満たす鍵は  $2^{16}$  個あって、それらはさらに 256 個のクラスに分類され、各クラスに属する鍵は展開されたラウンド鍵の左半分が相互に一致する。そのようなクラスの例を表 3.42 に示す。

表 3.42 では各中間鍵の左側 32 ビットが相互に一致しているが、右側 32 ビットが相互に一致するような鍵も同様の考察により探索できる。表 3.43 にその様な例を示す。鍵スケジュールに関するこの特性がただちに SEED の攻撃に結びつく訳ではないが、このような特異な性質が生じないように鍵スケジュールを設計するのが普通である。

## (3) その他のコメント

SEED の提案者は S-box の設計にあたり、 $GF(2^8)$  のべき乗関数の出力にアフィン変換を施して不動点 (0 及び 1) が生じるのを避けたとしている。今回の評価の過程で、0 及び 1 以外の不動点があることが判明した。 $S_1$ -box の新たな不動点は 23 および 230、また  $S_2$ -box の新たな不動点は 28 である。不動点に関しては、設計者が主張す

Key =	9b9b9b9b	9b9b9b9b	11111111	11111111
Round key no.				
1	4124db1d	3451bd29		
2	9a0f9a3a	4b127456		
3	79efee8e	273d39c9		
4	57215006	b12689b3		
5	03c24bbc	5f7092c7		
6	c0a53c4c	2b831b79		
7	cf3ebb62	d29fac9a		
8	2a14ef6c	a2c6cfe2		
9	7b85aa09	07894284		
10	f527f311	9100f2f9		
11	4ee60e85	14546a91		
12	26d5c935	864101db		
13	803e5e92	34e0e2c0		
14	c91d482b	2b10ede5		
15	0788fd30	2d60d71e		
16	f92d78ce	2bd7ef41		

Key =	3a3a3a3a	3a3a3a3a	72727272	72727272
Round key no.				
1	4124db1d	e0ef1874		
2	9a0f9a3a	711b066c		
3	79efee8e	5c178ff9		
4	57215006	0b809197		
5	03c24bbc	26afe9b0		
6	c0a53c4c	3c1b8a18		
7	cf3ebb62	573ddeb6		
8	2a14ef6c	c0be0d10		
9	7b85aa09	75080ba7		
10	f527f311	56ab375e		
11	4ee60e85	39e99972		
12	26d5c935	1591baad		
13	803e5e92	0ffc828b		
14	c91d482b	2d9680fc		
15	0788fd30	8e5a5bd0		
16	f92d78ce	5e235141		

Key =	2a2a2a2a	2a2a2a2a	82828282	82828282
Round key no.				
1	4124db1d	07460ff4		
2	9a0f9a3a	b82298f4		
3	79efee8e	7ee3b13e		
4	57215006	46c3d6b0		
5	03c24bbc	4af65578		
6	c0a53c4c	1bb446d4		
7	cf3ebb62	0b5a1d9e		
8	2a14ef6c	bfaa5324		
9	7b85aa09	4c16e012		
10	f527f311	1d68f56f		
11	4ee60e85	7def7131		
12	26d5c935	52eff20b		
13	803e5e92	3c3c924e		
14	c91d482b	e02f858f		
15	0788fd30	74fd6be4		
16	f92d78ce	a9ccd586		

表 3.42: 各ラウンド鍵の前半部が一致する鍵の例

Key =	9b9b9b9b	9b9b9b9b	efefefef	efefefef
Round key no.				
1	68c9edf1	7d28cbaf		
2	7e8e4d27	f9c76fad		
3	aa37e9ee	f59dd258		
4	5da694ad	7605924a		
5	c61b186a	b3c83014		
6	45dc4ae5	bf0fcbe		
7	05ce5df3	fd6a1882		
8	9ab323b3	6ef967c7		
9	a08e3ccc	d883dcd7		
10	8c92b184	13ddd10c		
11	77553f19	af7cecc4		
12	24e69b24	e007b43e		
13	bca52806	5f7651a0		
14	dd2474e9	1e09a2f2		
15	0eeecd5b	9c28a623		
16	3685e91e	bcad5740		
Key =	3a3a3a3a	3a3a3a3a	8e8e8e8e	8e8e8e8e
Round key no.				
1	0d92c044	7d28cbaf		
2	de60205a	f9c76fad		
3	d9258549	f59dd258		
4	9d84df1f	7605924a		
5	ea0a79a8	b3c83014		
6	638fd5fa	bf0fcbe		
7	470a077c	fd6a1882		
8	c252c5d8	6ef967c7		
9	a6b5f762	d883dcd7		
10	a55b43b7	13ddd10c		
11	ca3a056e	af7cecc4		
12	a678af9c	e007b43e		
13	4aa21758	5f7651a0		
14	a0ab171a	1e09a2f2		
15	1d432710	9c28a623		
16	ad80bb01	bcad5740		
Key =	2a2a2a2a	2a2a2a2a	7e7e7e7e	7e7e7e7e
Round key no.				
1	f23c7655	7d28cbaf		
2	0b5f9dbd	f9c76fad		
3	656eb6da	f59dd258		
4	886b8015	7605924a		
5	caac1ba9	b3c83014		
6	54d62348	bf0fcbe		
7	a9bdeb44	fd6a1882		
8	8bb07ddf	6ef967c7		
9	661831f3	d883dcd7		
10	05090fea	13ddd10c		
11	60f094cc	af7cecc4		
12	3393a0f5	e007b43e		
13	770ab190	5f7651a0		
14	10702afd	1e09a2f2		
15	0ef8e298	9c28a623		
16	7c8e917d	bcad5740		

表 3.43: 各ラウンド鍵の後半部が一致する鍵の例



る設計ポリシーと設計結果とが一致していないように思われる。ただし、これらの不  
動点が攻撃に結びつくことは現時点では考えられない。

### 3.3.4.4 ソフトウェア (SW) 実装評価

ソフトウェア実装については、提案者から提出されたソースプログラムを CRYPTREC  
事務局が委員立会いの下、PC にてコンパイル実行してその性能を測定した。評価結果を  
表 3.44 および表 3.45 に示す。

表 3.44: データランダム化部速度測定結果 (単位 [cycles/block])

Pentium III (650MHz)		
言語	C	
プログラムサイズ	45056 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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	846 / 871	846 / 873
2 回目	846 / 867	846 / 867
3 回目	846 / 866	846 / 867

表 3.45: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

Pentium III (650MHz)		
言語	C	
プログラムサイズ	49152 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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1233 / 1255	1235 / 1256
2 回目	1233 / 1255	1235 / 1257
3 回目	1233 / 1255	1235 / 1257

## 参考文献

- [1] Korean National Body, “Contribution for Korean Candidates of Encryption Algorithm (SEED)”, related to ISO/IEC JTC1 SC27 N2563, 2000. [http://www.kisa.or.kr/seed/data/algorithm/seed\\_english.doc](http://www.kisa.or.kr/seed/data/algorithm/seed_english.doc).
- [2] Korean Cryptography Standards, [http://dosan.skku.ac.kr/~sjkim/kg\\_std.html](http://dosan.skku.ac.kr/~sjkim/kg_std.html).
- [3] 屋並仁史, 下山武司, 「SEED の差分攻撃」 2002 年暗号と情報セキュリティシンポジウム, 2002 年 1 月 29 日-2 月 1 日.

### 3.3.5 MULTI-S01

#### 3.3.5.1 技術概要

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 ビット) から鍵ストリーム  $A, B, S$  を (処理するデータの長さに応じた長さだけ) 生成する。暗号化は、メッセージ  $M$  ( $n \times 64$  ビット)、冗長符号  $R$  (64 ビット)、秘密鍵  $A$  ( $A \neq 0$ , 64 ビット)、秘密鍵  $B_i$  ( $(n+2) \times 64$  ビット)、秘密鍵  $S$  (64 ビット) を入力として、暗号文  $C$  ( $(n+2) \times 64$  ビット) を出力する。復号は、暗号文  $C$  ( $64 \times n'$  ビット)、冗長符号  $R$  (64 ビット)、秘密鍵  $A$  ( $\neq 0$ , 64 ビット)、秘密鍵  $B$  ( $64 \times n'$  ビット)、秘密鍵  $S$  (64 ビット) を入力して、改ざん検出信号、またはメッセージ  $M$  ( $64 \times (n' - 2)$  ビット) を出力する。安全性については、メッセージ秘匿とメッセージ認証を同時に達成することと、現実的に攻撃の適用が困難となるような構成 (暗号解読の標的となる擬似乱数生成器の出力が一意に決められない) を目指した、としている。安全性は擬似乱数を発生する機構「擬似乱数生成器」の安全性に基づく。MULTI-S01 は擬似乱数生成器として PANAMA を用いている。

#### 3.3.5.2 技術仕様

暗号化処理では、メッセージ  $M$ 、冗長性  $R$  (64 ビット)、秘密鍵  $K$  (256 ビット) をそれぞれバイト列によるデータ ( $M(8)_i$  ( $i = 1, \dots, \lceil m/8 \rceil$ ),  $R(8)_i$  ( $i = 1, \dots, 8$ ),  $K(8)_i$  ( $i = 1, \dots, 32$ )) として入力する。暗号化処理の出力は暗号文  $C$  であり、 $C$  の長さは  $64 \times (\lceil m/64 \rceil + 2)$  ビットで、バイト列として出力する。これに対応する復号処理では、暗号文  $C$  ( $c$  ビット)、冗長性  $R$  (64 ビット)、秘密鍵  $K$  (256 ビット) をそれぞれバイト列によるデータ ( $C(8)_i$  ( $i = 1, \dots, \lceil c/8 \rceil$ ),  $R(8)_i$  ( $i = 1, \dots, 8$ ),  $K(8)_i$  ( $i = 1, \dots, 32$ )) として入力する。復号処理の出力は復号結果  $M'$  または改ざん検出信号であり、メッセージが出力される場合には、これをバイト列として出力する。暗号化・復号処理の内部は 64 ビットのブロックごとの処理で構成され、処理全体のブロックの数を  $n = \lceil m/64 \rceil + 2$  とする。擬似乱数生成器は、 $K$  を入力として、 $A$  (64 ビット) と  $B$  ( $64 \times (n+2)$  ビット)、 $S$  (64 ビット) を出力する。よって、暗号化処理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 ビットのデータ型との変換の際、Big-Endian により変換される。

#### 3.3.5.3 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。なお MULTI-S01 で用いているのは PANAMA の擬似乱数生成器の機能のみである。

### 3.3.5.4 詳細評価結果 (2000 年度)

#### ■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。

#### ■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。評価結果は表 3.46 の通りである。

表 3.46: データランダム化部速度測定結果 (単位 [cycles/block])

Pentium III (650MHz)		
言語	ANSI C	
プログラムサイズ	12868 KByte	
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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	176 / 180	-
2 回目	175 / 177	-
3 回目	176 / 177	-

**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、表 3.47 の結果となっている。

#### ■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、

表 3.47: 自己評価書での評価結果

	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	3.7Kbyte	267.3Mbps	18.0

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 -O6

CPU: Alpha 21164A 600 MHz RAM:512 Mbyte

OS: DIGITAL UNIX 4.0E

- ・ ModelSim VHDL/Verilog Version 5.4e (Model Technology)
- ・ Synplify (Synplicity Inc.)

である。評価結果は表 3.48 の通りである。

表 3.48: HW 実装評価結果

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。

0.35 $\mu$ m 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

### ■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。

### 3.3.5.5 詳細評価結果 (2001 年度)

CRYPTREC Report 2000 における MULTI-S01 に対する暗号評価の結果は、次のようにまとめられる。

- 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みつかっていない。
- 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。

- SW における処理速度は速いグループ。

上記の評価結果を受け、本年度は MULTI-S01 の安全性について継続的な評価を実施した。具体的手順としては、擬似乱数生成部 (PANAMA) と利用モードを実現する暗号化部を切り離し、各々について安全性を評価する手段によって、より精密な評価が可能となるであろうと考えた。各詳細評価内容は次の通りである。([ ] は評価者数。)

1. MULTI-S01 を暗号利用モードとみた評価 [2]
2. PANAMA の理論的暗号解析に対する安全性 [2]
3. PANAMA の計算機による乱数性検定 [1]

以下、2001 年度に行なったこれらの安全性評価について各々の概要とその結果を述べる。なお次節、次々節において、記号“●”に続く記述は外部評価者によるレポート内での評価をまとめたものであり、“√”に続く記述はそれらを受けた委員会コメントである。

### 3.3.5.6 MULTI-S01 の暗号利用モードとしての安全性評価

MULTI-S01 装置内部の暗号部品である PANAMA の利用方法と安全性との関連について評価を行う。なお MULTI-S01 で用いているのは PANAMA の擬似乱数生成器の機能のみであるため、本評価では PANAMA の内部構造までは踏み込まない。

#### ■MULTI-S01 仕様書について

- MULTI-S01 の仕様書は、一つのメッセージを暗号化する方法について述べているが、複数のメッセージを暗号化する方法について明確に述べていない (評価者 2)。
- √ 評価者 2 は複数のメッセージの暗号化方法を提案している。提案暗号化方法は、MULTI-S01 の仕様書にある暗号化方法の自然な拡張になっている。

#### ■MULTI-S01 の安全性の定義について

- 自己評価書の安全性の定義は、標準的な安全性の定義と比べると非常に弱い定義である。自己評価書では、privacy に関しては「一つの暗号文による暗号文単独攻撃を行う敵」に対する安全性を、authenticity に関しては「一つの平文・暗号文ペアからなる既知平文攻撃を行う敵」に対する安全性を証明している。これに対し、提案者は privacy 及び authenticity 両方に関して「適応的選択平文攻撃を行う敵」に対する安全性を証明しなければならない。(評価者 1)
- ストリーム暗号の authenticated encryption の安全性の定義は、一般的には知られていない。ストリーム暗号に適した安全性の定義を与える (評価者 2)。
- √ 評価者 1 及び評価者 2 共に提案者が考えるべき安全性の定義を与えている。privacy に関する安全性の定義 (indistinguishability from random bits) は評価者 1 及び評価者 2 で全く同一の定義となっている。authenticity に関しては評価者 1 は verification oracle に 1 回しか質問を許さないのに対し、評価者 2 は複数回の質問を許しており、評価者 2 の方が強い定義になっている。

#### ■MULTI-S01 の安全性について

- 評価者 1 は MULTI-S01 が自ら提案した安全性の定義を満たすことの証明の指針、概要を与えている。評価者 1 の証明は未完成であり、提案者自身が証明を完成させるべきである、と述べている。
- 評価者 2 は MULTI-S01 が自ら提案した安全性の定義を満たすことを示している。
- ✓ MULTI-S01 の計算量的安全性は評価者 2 によって示された。また SCIS2002 において、提案者による同様の結果が報告されている [5]。

#### ■代替方式について

- Vernam 暗号と Carter-Wegman MAC を組み合わせれば MULTI-S01 と同様のストリーム暗号の authenticated encryption を作ることができ、この構成の方が、構造的に簡単で暗号文が短く、任意の長さのメッセージを扱うことができる。(評価者 2)
- ✓ Vernam 暗号と Carter-Wegman MAC の組み合わせの場合と、MULTI-S01 とはそれぞれに優位点があると考えられるため、現状では比較が困難と考えられる。

#### ■結論

- ✓ 評価者 1 及び評価者 2 は MULTI-S01 の安全性を PANAMA の安全性に帰着させる、という観点から、MULTI-S01 の計算量的安全性を評価している。MULTI-S01 の安全性を適切に定義しその観点のもとで評価した結果、本暗号の安全性は PANAMA の安全性に帰着可能であることが示された。

#### 3.3.5.7 PANAMA の理論的暗号解析

MULTI-S01 の安全性は、PANAMA の影響が大きいことからモジュールとして利用されている PANAMA 自身の安全性を理論的に評価する必要がある。なお前節の結果から、MULTI-S01 の安全性は PANAMA に帰着されることが示されたことから、本節の結果は直接 MULTI-S01 の安全性に影響する。PANAMA は 1998 年に Daemen と Clapp によって提案された暗号アルゴリズムであり、ハッシュ関数と擬似乱数生成器の 2 種類を含むが、MULTI-S01 で用いているのは擬似乱数生成器のみである。よって本評価では PANAMA の擬似乱数生成部の安全性のみ評価する。

本評価で実施された暗号解析は次の 5 項目である。

1. Chosen-IV Collision Attacks (評価者 1)
2. Chosen-IV Differential Attacks (評価者 1)
3. Chosen-IV Related-Key Attacks (評価者 1)
4. An Equivalent Representation (評価者 2)
5. Analysis of Simpler Variants of PANAMA (評価者 1, 評価者 2)

1 から 3 までの解析はいずれも PANAMA の Deviation Parameter と呼ばれる 256 ビットの値を、攻撃者が自由に選択できる場合を想定している。この値は PANAMA へ入力される公開情報であると考えられる。秘密鍵 (256 ビット) を  $K$ 、Deviation Parameter を  $Q$  とし、PANAMA の出力する Key Stream を  $PANAMA(K, Q)$  とする。安全性の評価基準としては、あらゆる  $K$  について、 $Q$  を攻撃者がどのように定めたとしても  $PANAMA(K, Q)$

と 2 値の一樣確率変数との区別が計算量的に困難となることが挙げられる。解析 4 は PANAMA のある等価表現を示し、その中で互いに独立な要素を示している。解析 5 は PANAMA に対する数種類の簡略化バージョンについて暗号解析を行なった結果である。

### ■ Chosen-IV Collision Attacks

- ある  $K$  について、 $PANAMA(K, Q) = PANAMA(K, Q')$  となる  $(Q, Q')$  ( $Q \neq Q'$ ) が存在するのならば、このような  $Q$  と  $Q'$  を選択することにより秘密鍵が  $K$  であるか否かを、ほぼ正確に判定できるが、PANAMA の構造を調査したところ、このような  $K$  は存在しないと分かった。しかし、 $Q$  がより長いビット長を持つ場合ははっきりしておらず、例えば、 $Q$  が 512 ビットの場合でも存在するかもしれない。(評価者 1)
- ✓ 少なくとも任意に  $Q$  を長くとれる場合は、このような  $K$  が存在するはずである。また Rijmen et.al. の結果 [4] を見れば、 $K$  が与えられた元では、そのような  $(Q, Q')$  の組は現実的に見つけられるといえる。

### ■ Chosen-IV Differential Attacks

- $K$  と  $Q$  を Push (バッファへ入力) した直後の PANAMA の内部状態を  $M$ 、Blank Pull を済ませて Key Stream を出力する直前の内部状態を  $M'$  とすると、 $Q, M, M'$  の差分に関して、高い確率で成立する式 (Differential Equation) が存在すれば、この式の計算を攻撃の手段として利用できるが、解析の結果高い確率を持つ Differential Equation は見つけられなかった。ただし Rijmen et.al. の結果からは、入力全体  $(K, Q)$  の差分と  $M, M'$  の差分については、高い確率の Differential Equation が存在することを示している。(評価者 1)

### ■ Chosen-IV Related-Key Attacks

- $K$  と  $Q$  に任意の差分値を加えた時の Key Stream  $PANAMA(K \oplus \Delta K, Q \oplus \Delta Q)$  を得られるという仮定のもとでの攻撃が考えられる。(評価者 1)
- ✓ この攻撃は、 $Q$  が 512 ビットであるときは Collision Attacks と関連するが、そうでない限りは非現実的な状況のもとでの攻撃である。

### ■ An Equivalent Representation of PANAMA

- 時点  $t$  におけるバッファの  $i$  番目 ( $i = 0, \dots, 31$ ) の段の  $j$  番目 ( $j = 1, \dots, 8$ ) の語 (word) の、 $k$  番目 ( $k = 1, \dots, 32$ ) のビットを、 $b_{j,k}^i(t)$  で表す。 $j, k$  を省略した場合は、それぞれについてのベクトル表現であるとする。同様に、時点  $t$  における State の  $j$  番目 ( $j = 0, \dots, 17$ ) の語 (word) の  $k$  番目のビットを  $a_{j,k}(t)$  で表す。バッファの 25 番目の更新が  $b_j^{25}(t+1) = b_j^{24}(t) \oplus b_{j+2 \bmod 8}^{31}(t)$  となることから、バッファの更新だけを見ればビットの変化が影響する範囲は  $\{b_0, b_2, b_4, b_6\}$  と  $\{b_1, b_3, b_5, b_7\}$  とに分けられる。また、更新が語単位であるので、ある段での  $k$  番目のビットの変更は、他の段の  $k$  番目以外のビットには影響しない。従って、PANAMA の構成を 64 個の部分構造に分解して表現することができる。(評価者 2)
- ✓ バッファが上述の部分構造に分割できるため、Push モードではバッファ内の 1 ビットを変えても、その影響は各部分構造内に限定される。しかし Pull モードでは、

バッファへの入力が入力が State に依存するため、バッファ内の 1 ビットの影響が全体に及ぶことになる。

### ■Analysis of Simpler Variants of PANAMA

PANAMA を簡略化したものとして評価者 1 により Blank Pulls を省略した場合、評価者 2 により PANAMA-S1, PANAMA-S2, PANAMA-SM があげられているが、ここでは主に、最も重要と思われる Blank Pulls を省略した場合と PANAMA-S2, PANAMA-SM について説明する。

#### (1) PANAMA における Blank Pulls を省略した場合

- Blank Pulls を省略した場合は、 $K$  と  $Q$  を Push した後の内部状態のうち、State の一部である  $a_8, \dots, a_{16}$  は Key Stream の最初の 256 ビットに相当するが、これらは  $Q$  に依存せず決まる。従って、 $Q$  を変えた時 Key Stream の最初の 256 ビットが変化するか否かで区別すればよい。同様のことは、Blank Pull の回数が、正規の回数である 33 回よりも少ない場合には起きうる。Blank Pulls が 14 回以内の場合、適当な差分ベクトルを持つ入力ペアを用いて一様確率変数との区別が可能となる。(評価者 1)
- ✓ この方法は Related-Key Attack に分類されるため、あまり現実的とは言えないものの  $K$  の分布が低いエントロピーを持つような場合には有効と思われる。ここで示したような解析が、正しく 33 回の Blank Pulls を行う場合にも有効であるかは不明である。
- ✓ MULTI-S01 は、Blank Pulls を行うかどうかについて、何も言及していない。Reference Implementation では Blank Pulls を行っているため問題がないが、前述の Related-Key Attack を考えると、Blank Pulls を行うということを明記する必要がある。

#### (2) PANAMA-S2

- PANAMA-S2 とは、State の更新の関数  $\rho = \sigma \circ \theta \circ \pi \circ \gamma$  の代わりに  $\rho = \sigma \circ \pi$  という更新の関数を用いるバージョンである。攻撃は、長さ  $n$  (語単位) の Key Stream  $[a_j(t)]_{j=9}^{16}$ ,  $t = 1, \dots, n$  を得たもとで、Pull を行っているある時点での、バッファと State の内容を推定することにより、以後の Key Stream を計算することが可能となる。(評価者 2)
- ✓ 攻撃アルゴリズムは論理的には間違いがないと思われるが、攻撃アルゴリズムの計算量を、“Proportional to  $2^{65}$ ” であるとしているが、アルゴリズム中で探索する変数の数は  $2^{4 \cdot 7} \cdot 2^{25} = 2^{53}$  なので、トータルでは “Proportional to  $2^{58}$ ” が正しいと思われる。

#### (3) PANAMA-SM

- 最も PANAMA と似ているバージョンである PANAMA-SM では、State の更新関数を  $\rho = \sigma \circ \theta^* \circ \pi \circ \gamma^*$  と置く。報告書では  $\theta^*$  と  $\gamma^*$  が満たすべき条件を具体的に挙げているが、要は  $a_j^*(t) \equiv \theta^* \circ \pi \circ \gamma^*(a_j(t))$  としたときに、 $a_2^*(t)$ ,  $a_4^*(t)$ ,  $a_7^*(t)$ ,  $a_9^*(t)$ ,  $a_{11}^*(t)$ ,  $a_{12}^*(t)$ ,  $a_{14}^*(t)$ ,  $a_{16}^*(t)$  については Key Stream のみで計算可能であるという条件を満たしていればよい、というものである。この条件により、攻撃アルゴリズム



自体は PANAMA-S2 となら変わりなく実行することが可能となっている。(評価者 2)

### ■結論

- 結果として、一番シンプルな攻撃については安全であり、その他 2 つについても特に安全性が低いとは結論できていないが、これは評価にかかる時間の少なさに依るところが大きい。(評価者 1)
- √ PANAMA-S2 と PANAMA-SM では、State の更新における中間の計算結果の一部が、Key Stream より確定的に求まるという、PANAMA にはない性質を持つ。この性質自体が攻撃アルゴリズムに大きく寄与しているため、報告書で示された攻撃が直接的な脅威となることはないと思われる。しかし An Equivalent Representation で述べられているように、“PANAMA の構造を 64 個の部分構造へ分割して表現することができる”という性質があり、この性質を利用した攻撃が存在する可能性はある。
- √ 実際の PANAMA では、 $\theta \circ \pi \circ \gamma(a_j(t))$  が上記のような都合のよい条件を持つ訳ではないので、この攻撃アルゴリズム自体をわずかに修正 (例えば、探索すべき変数の数を増やすなど) することで実際の PANAMA へ攻撃できるとは考えられない。
- √ 評価者 2 は PANAMA の改良案として、バッファの  $b_j^{25}(t)$  の更新をより複雑にすることで、上記の性質を持たないようにすべきだとしている。これは妥当な案だが、バッファの更新を並列処理することが困難となるという、実装上の問題を同時に引き起こす可能性がある。

#### 3.3.5.8 PANAMA の乱数性に関する統計検定結果

ここでは MULTI-S01 で使用されている PANAMA の統計的性質について解析を行う。すなわち PANAMA の内部構造には一切立ち入らず、PANAMA を擬似乱数生成器として Black-Box 的に捉え、その出力系列の各種統計的性質について評価を行ない、ストリーム暗号 MULTI-S01 の構成要素として必要十分な性能を備えているかを検定した。擬似乱数性の検定法には NIST の SP 800-22 に付属する擬似乱数検定プログラムを用いて検証した。

その結果、初期攪拌 32 回後では、入力の変りによる出力乱数の系列間の偏りは全く見られないと考えられる。また、PANAMA 生成の出力系列と真性乱数系列とは多くの統計的性質において区別がつかないと判断される。

**■擬似乱数生成器 PANAMA** PANAMA は、1998 年に J.Daemen らによって提案された暗号モジュールであり、PANAMA によってストリーム暗号やハッシュ関数らが構成可能である。

PANAMA は 256 ビットの秘密鍵  $K$  と 256 ビットの乱数列番号  $Q$  を入力とし、任意長の擬似乱数列を出力する。PANAMA は次の 3 つの動作モードを有する。

- reset モード: 内部状態のリセット
- push モード: 秘密鍵  $K$  や乱数列番号  $Q$  の入力
- pull モード: 初期攪拌と擬似乱数列の生成

■**統計試験ツール、SP 800-22 とは?** ここで採用した統計試験ツールは NIST の SP 800-22 である。SP 800-22 は NIST が公開している暗号アプリケーションのための乱数と擬似乱数の統計試験ツール及びそのドキュメントである [6, 7]。なお、AES 選定では、その候補暗号に対して本ツールによる統計検定が行われた。SP 800-22 には 16 通りの検定法があり 189 種類の試験を行うことができる。次に SP 800-22 で実行可能な統計検定法の一覧を示す。

- Frequency Test
- F-T within a Block
- Cusum Test
- Runs Test
- Test for the Longest Runs of Ones in a Block
- Binary Matrix Rank Test
- Discrete Fourier Transform Test
- Non-overlapping Template Matching Test
- Overlapping Template Matching Test
- Maurer's Universal Statistical Test
- Approximate Entropy Test
- Random Excursion Test
- Random Excursion Variant Test
- Serial Test
- Lempel-Ziv Compression Test
- Linear Complexity Test

一方、SP 800-22 の出力は「合格率」と「分布」である。各試験では、検査対象の擬似乱数列が正規分布もしくは  $\chi^2$  分布のどの辺りに位置するかを P-Value という値 (0 ~ 1) で表現する。出力の「合格率」とは P 値が 0.01 以上を合格とし、300 系列に対する合格率である。「分布」とは 300 系列分の P 値が一樣かどうかを検査するものであり、各系列の P 値を 10% ごと 10 区間で集計して評価する。

■**乱数検定 (実験結果)** 評価方法としては、暗号アルゴリズムの特性を検討するために、暗号アルゴリズムの一部を使用して攪拌過程が検討可能な [Partial Round Test] と暗号アルゴリズム全体での擬似乱数性の検査する [Full Round Test] の 2 種類のテストを行った。

#### [Partial Round Test]

PANAMA を 256 ビットブロック暗号とみなし、秘密鍵を鍵、乱数列番号を平文、32 回の初期攪拌を Round と見なすことで Partial Round Test を行った。PANAMA では鍵と平文 (乱数列番号) の扱いが同じであることに注意して、Plaintext/Ciphertext Correlation と同様に鍵との相関を見る Key/Ciphertext Correlation を追加した。こうして秘密鍵や乱数列番号に偏りがある場合に出力系列間に偏りがあるかどうかを評価できる。その結果を Round 順に並べることで、入力された鍵や乱数列番号の攪拌されてゆく様子が観測できる。

#### [Full Round Test]

以下の手順で Full Round Test を実行した。

1. PANAMA に対して乱数で生成した秘密鍵と乱数列番号を入力し初期攪拌後の 314572800 ビットの擬似乱数列を生成
2. 1. で生成した擬似乱数列を  $1048576 \times 300$  とみなし SP 800-22 で統計試験を実行

3. 1. 及び 2. を 128 回繰返し SP800-22 の「合格率」と「分布」を出力  
ここで手順 3. を 128 回繰り返したのは信頼性向上のためである。

以上の統計検査の結果、PANAMA の疑似乱数性には特段の欠陥は見当たらないと判断される。

■その他 今回統計検定を実際に行った過程で、NIST の乱数検定ドキュメント SP 800-22 に付属している検定プログラム [7] には、少なくとも 2 箇所 (検定項目 DFT、Lempel-Ziv 各々の「分布」評価) で Heuristic なパラメータが真性乱数のそれからずれている可能性があることが判明した。

■結論 PANAMA の統計的性質に関して SP 800-22 を用いて評価を行った。PANAMA の初期攪拌における秘密鍵と乱数列番号の攪拌性を評価した結果、入力する秘密鍵や乱数列番号のデータの偏り、差分の偏り、及び入力データとの相関の全てについて、初期攪拌回数は 7 回程度で十分な攪拌が行われていることが判明した。そのために PANAMA 仕様の疑似乱数列が出力される初期攪拌回数 32 回では、入力の偏りによる出力乱数の系列間の偏りは全く見られないと考えられる。

また PANAMA を用いて長い疑似乱数系列を出力した場合の統計的性質を評価した結果、その疑似乱数系列は数多くの統計的性質において真性乱数系列と区別がつかないと判断できる。以上 PANAMA に対する乱数検定からは、PANAMA の疑似乱数性に関して特段の欠陥は見当たらないと言える。

### 3.3.5.9 総評

2001 年度の詳細評価によって、MULTI-S01 の安全性が疑似乱数生成器 PANAMA の性質に帰着されることが示された。一方で PANAMA に関する安全性解析、統計検定の結果からは、致命的な欠点は見付からなかった。2001 年度の評価から、本暗号の安全性に対する不安が完全に払拭されたとは言いがたいものの、疑似乱数生成器としての安全性が理論的に証明されている例はごく少なく、特に高速演算を特徴とする場合には、安全性を完全に保証することが一般的に言って困難であると考えられることから、この状況を本暗号が持つ特有の問題点として指摘し続けることが適切であるとは考えにくい。本暗号は 2002 年度以降の監視対象暗号と判断する。

### 参考文献

- [1] 古屋総一, 高橋昌史, 渡部大, 宝木和夫 “疑似乱数生成器を使ったメッセージ認証可能な共通鍵暗号の提案,” 信学技報 ISEC2000-8, 2000.
- [2] 古屋総一, 渡部大, 宝木和夫 “MULTI-S01 のパディングと安全性についての考察,” 信学技報 ISEC2000-68, 2000.
- [3] J. Daemen, C. Clapp. “Fast Hashing and Stream Encryption with Panama,” FSE’98., 1998.
- [4] V. Rijmen, B. Rompay, B. Preneel, J. Vandewalle. “Producing Collisions for Panama.” FSE’01., 2001.
- [5] 古屋総一 “MULTI-S01 の計算量的安全性,” Proceedings of SCIS2002, pp.253–258. 2002.

- [6] NIST Special Publication 800-22, “A Statistical test suite for random and pseudorandom number generators for cryptographic applications,” (<http://csrc.nist.gov/rng/SP800-22.pdf>, <http://csrc.nist.gov/rng/errata2.pdf>)
- [7] NIST Special Publication 800-22, “NIST Statistical Test Suite,” (<http://csrc.nist.gov/rng/sts-1.4.tar>, <http://csrc.nist.gov/rng/sts.data.tar>)

## 3.4 監視状態の暗号の評価

### 3.4.1 Hierocrypt-L1

#### 3.4.1.1 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  $4 \times 4$  MDS 行列を 2 個並列に並べたバイト置換層 ( $MDS_L$ )、 $GF(2^{32})$  の  $2 \times 2$  MDS 行列からなるバイト置換層 ( $MDS_H$ )、鍵加算層 (K) から構成される。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

#### 3.4.1.2 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ることで拡大鍵を生成する。復号時にも on-the-fly での鍵設定の初期遅延が小さくなるよう、中間鍵列が途中で逆転して戻ってくる折り返し型の構造を採用している。

### 3.4.1.3 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。

#### 3.4.1.4 安全性評価結果

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らない。

■**解読可能段数** Ferguson らの手法 [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 [5]。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。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ことが詳細評価により示されている。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 バイトのみ活性化  $\Delta$  集合を与えると、4 層目の入力は 8 バイト全てが活性化  $\Delta$  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以下の 2 点が確認されている。(1) 最大差分/線形特性確率は 2 段 (4 層) で  $2^{-90}$  を超えない。(2) 最大差分/線形確率は 2 段以降  $2^{-48}$  を超えない。(1) は活性 S-box 数の下限から明らかであり、(2) は各段の鍵が独立かつ一様との仮定の基で Hong らの手法 [4] を用いて証明可能である。なお、3 段以上の段数において最大差分/線形確率が  $2^{-48}$  より確実に小さな値をとるか否かは現在のところ明らかになっていない。ちなみに、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  $2^{-48}$  より小さくなるの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。

自己評価書において、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おり、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であろうと予想されている。

### 3.4.1.5 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。評価結果は表 3.49 及び 3.50 の通りである。

表 3.49: データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	199 / 201	204 / 206
2 回目	199 / 201	204 / 206
3 回目	200 / 201	204 / 205
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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	378 (332) / 380 (336)	500 (304) / 504 (307)
2 回目	378 (332) / 380 (336)	500 (304) / 504 (308)
3 回目	378 (332) / 380 (336)	500 (304) / 504 (308)
Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	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 -O3	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	210 (179) / 214 (182)	210 (179) / 212 (182)
2 回目	210 (179) / 214 (182)	210 (179) / 212 (182)
3 回目	210 (179) / 213 (182)	210 (179) / 212 (182)

**備考** Ultra SPARC Iii と Alpha 21264 の測定において、(カッコ)内の値は応募者による測定プログラムの改変した場合の測定値。測定プログラムは汎用性を持たせるため巨大なバッファ領域を確保しているが、その領域を必要な分だけ取るように改変した。速度評価の主旨を違えるような改変は行っていないことは確認済み。

応募者からは以下の自己評価が報告されている。

CPU: Pentium III 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

表 3.50: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	374 / 375	616 / 618
2 回目	374 / 377	616 / 617
3 回目	374 / 375	616 / 618
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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	718 (616) / 721 (620)	1203 (1014) / 1215 (1031)
2 回目	718 (616) / 721 (619)	1203 (1012) / 1215 (1030)
3 回目	718 (616) / 721 (620)	1203 (1015) / 1215 (1031)
Alpha 21264 (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 -O3	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	390 (386) / 394 (389)	625 (617) / 654 (648)
2 回目	390 (386) / 394 (389)	625 (617) / 653 (648)
3 回目	390 (386) / 394 (389)	625 (617) / 653 (648)

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

### 3.4.1.6 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 0.35 $\mu$ m CMOS ASIC ライブラリ  
回路記述: Verilog-HDL  
Synthesizer: Design Compiler

表 3.51: HW 実装評価結果

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 Compiler 1999.10-3  
シミュレーション条件: 1.35V 70 °C (標準ケースでは、1.5V25 °C)  
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 at <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



## 3.4.2 MISTY1

### 3.4.2.1 技術概要

MISTY1 は 1996 年に三菱電機株式会社が開発したブロック長 64 ビット、暗号化鍵長 128 ビットの共通鍵ブロック暗号であり [10, 11]、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 [8, 9]。実装面では IC カード向け 8 ビットプロセッサから 64 ビット RISC プロセッサまであらゆる分野に適した暗号であり、とりわけレジスタが多いプロセッサでは Bitslice 実装法によってソフトウェアによる高速処理が実現できる点は大きな特徴である (Alpha プロセッサでは 68 cycle/block) [12, 13, 14, 15, 16]。さらにハードウェアでは 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。

### 3.4.2.2 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ェアで実用的な性能を達成すること。できる限り多くのアプリケーションで利用可能な暗号をめざし、特定のプロセッサでのみ高速処理が可能となるような命令を用いず、あらゆるプロセッサで適度な高速性と小型化が実現できる基本的な命令のみが採用されている。また IC カードでの実装を考慮しワークメモリサイズが小さくなるよう設計されている。
- ハードウェア上で十分な高速性を実現すること。算術演算はハードウェアでの速度低下につながるため採用せず、論理演算とテーブル参照だけからアルゴリズム全体を構成されている。またテーブルの設計においてはハードウェアで最適化されるように考慮されている。

### 3.4.2.3 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、Algorithm 1 として記述された暗号が MISTY1 のベースになっていると考えられる [8, 9]。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 [10, 11]。

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 [21]。

### 3.4.2.4 安全性評価結果

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ェ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 [19, 20]。また計算機を用いた解読実験では、5 段 MISTY1 (FL 関数あり) の拡大鍵の一部が約 1 時間で導出されたとの結果が発表されている [4] (表 3.52 参照)。さらに、Impossible Differential を用いた攻撃 [7]、Integral Cryptanalysis (SQUARE Attack) を用いた攻撃 [6, 5]、複数次変数による補間多項式を用いた攻撃 [18, 4] 等が発表されている。しかしこれらの攻撃はいずれも段数を減らすなど標準仕様と異なるものに対する解析結果であり、現時点では標準仕様の MISTY1 の安全性に影響を与える結果は知られていない。

表 3.52: 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	$2^{8.4}$	$2^{85}$	$2^4$	$2^{7.2}$
5 段	$2^{22}$	$2^{33}$	$2^{10.5}$	$2^{17}$
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にはできないものの、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  $K_1, K_2, \dots, K_8$  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  $K_1 = K_2 = \dots = K_8$  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  $K_1 = K_2 = \dots = K_8$  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、ソフトウェア、ハードウェアともに、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 関数に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ェア実装時には注意した方がよいと考えられる。

### 3.4.2.5 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。評価結果は表 3.53、3.54 の通りである。

表 3.53: データランダム化部速度測定結果 (単位 [cycles/block])

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		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	213 / 215	208 / 210
2 回目	213 / 215	208 / 210
3 回目	213 / 214	209 / 211
Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	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		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	203 / 205	206 / 208
2 回目	203 / 206	206 / 208
3 回目	203 / 205	206 / 208

復号の処理時間が暗号化処理時間に比べ最大で 5 cycle 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、Pentium III (800MHz) と Alpha 21264 (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 cycles/block, 192 cycles/block であるとしている。

表 3.54: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

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		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	357 / 358	350 / 351
2 回目	357 / 358	350 / 351
3 回目	357 / 358	350 / 351
Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	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		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	334 / 338	337 / 340
2 回目	337 / 338	337 / 340
3 回目	334 / 338	337 / 340

■IC カード実装他 応募者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 [16]。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 [17]。暗号化、鍵スケジュール速度の単位はいずれも cycles/block である。

プロセッサ	暗号化、鍵スケジュール速度	ROM	RAM
M16C	1877,743	3400byte	64byte
H8/300	6018,1240	1900byte	43byte
Z80	25486 (鍵スケジュール込)	1598Byte	44byte

### 3.4.2.6 ハードウェア (HW) 実装評価

以下の環境でハードウェ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  $\mu$ CMOS ASIC Design Library

論理合成ツール: Design Compiler .2000.05-1

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 [1, 2]、ならびに Xilinx 社製 FPGA である Vertex-E シリーズおよび Vertex II シリーズを用いた実装例 [3] が示されている。

実装 1: 7.6K ゲート 72Mbps

実装 2: 50K ゲート 800Mbps

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

## 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] 市川哲也, 反町亨, 粕谷智巳, 松井充, NESSIE 提案ブロック暗号のハードウェア実装について (I), Proceedings of SCIS2002, SCIS2002-12C-3, 2002.
- [4] 秦野康生, 田中秀磨, 金子敏信, MISTY1 の高階差分攻撃, Proceedings of SCIS2002, SCIS2002-13A-5, 2002.
- [5] I. Kim, Y. Yeom, H. Kim, Square Attacks on the Reduced-Round MISTY1, Proceedings of SCIS2002, SCIS2002-13A-2, 2002.
- [6] L. Knudsen, D. Wagner, Integral Cryptanalysis, Pre-proceedings of the International workshop of Fast Software Encryption 2002, pp.108-122, (to appear in Lecture Notes in Computer Science), 2002.
- [7] U. Kuehn, Improved cryptanalysis of MISTY1, Pre-proceedings of the International workshop of Fast Software Encryption 2002, pp.56-70, (to appear in Lecture Notes in Computer Science), 2002.
- [8] 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.
- [9] M. Matsui, New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis, Proceedings of the 3rd international workshop of Fast Software Encryption, Lecture Notes in Computer Science 1039, pp.205-218, Springer Verlag, 1996.
- [10] 松井充, ブロック暗号アルゴリズム MISTY, 信学技報 ISEC96-11, 1996.
- [11] M. Matsui, New Block Encryption Algorithm MISTY, Proceedings of the 4th international workshop of fast software encryption, Lecture Notes in Computer Science 1267, Springer Verlag, pp.54-68, 1997.
- [12] 中嶋純子, 松井充, MISTY のソフトウェアによる高速実装について (I), 信学技報 ISEC97-12, 1997.
- [13] 中嶋純子, 松井充, MISTY のソフトウェアによる高速実装について (II), Proceedings of SCIS 98, SCIS98-9.1.B, 1998.

- [14] J. Nakajima, M. Matsui, Fast Software Implementation of MISTY1 on Alpha Processors, IEICE Trans. Functionals, Vol E82-A, No.1 January 1999.
- [15] 中嶋純子, 松井充, MISTY のソフトウェアによる高速実装について (III), 信学技報 ISEC2000-81, 2000.
- [16] 中嶋純子, 松井充, 共通鍵暗号 MISTY1 の最適なソフトウェア実装について, Proceedings of SCIS 2001, 13A-3, 2001.
- [17] 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.
- [18] 渋谷 香士, 小林 篤, 下山 武司, 辻井 重男, MISTY1 の内部関数の多項式表現と暗号解析への応用, Proceedings of SCIS2002, SCIS2002-10A-3, 2002.
- [19] H. Tanaka, M. Hisamatsu, T. Kaneko, Higher Order Differential Attack of MISTY1 without FL functions, JWIS'98, ISEC98-66, pp.143-150, 1998.
- [20] 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.
- [21] 3GPP, KASUMI, ETSI/SAGE Specification, 1999.  
(<http://www.etsi.org/dvbandca/3GPP/3gppspecs.htm>)

### 3.4.3 Triple DES

詳細は 6.3.2 節を参照のこと。

### 3.4.4 Camellia

#### 3.4.4.1 技術概要

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 ビットブロック長の共通鍵暗号であり、2000 年に学会発表された [3]。鍵長は 128/192/256 ビットの 3 通りである。その基本構造は 18 段 (128 ビット鍵長) もしくは 24 段 (192/256 ビット鍵長) の Feistel 型構造であり、6 段ごとに FL/FL<sup>-1</sup> 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 (22.0Mbits/(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。

#### 3.4.4.2 技術仕様

ラウンド関数の構成要素の基本部分は S-box と排他的論理和から、また 関数の構成要素は論理和、論理積、排他的論理和およびローテーションからなっている。算術演算は一切用いていない。これにより、長いクリティカルパスを排除し回路規模の小型化を実現している。また拡大鍵の生成関数の設計においては、on-the-fly 鍵生成が可能な設計を用いている。

#### ■データランダム化部

**128 ビット鍵の場合** データランダム化部は 18 段の Feistel 構造と FL/FL<sup>-1</sup> 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<sup>-1</sup> 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<sup>-1</sup> 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<sup>-1</sup> 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 つの拡大鍵を接続したものと排他的論理和が計算され、それを 2 等分する。そして以下の演算を  $r = 1 \sim 18$  まで実行する (但し、 $r = 6, 12$  は除外)。

$$\begin{aligned} L_r &= R_{r-1} \oplus F(L_{r-1}, k_r) \\ R_r &= L_{r-1} \end{aligned} \quad (3.1)$$

$r = 6, 12$  の場合は FL/FL<sup>-1</sup> 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<sup>-1</sup> 関数とからなる。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  $K_a$  と  $K_b$  を生成する。但し、 $K_b$  の方は 192 あるいは 256 ビット鍵の場合のみ使用する。拡大鍵は中間的な鍵を循環シフトさせた値の左あるいは右半分の値になっている。鍵スケジュール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。

### 3.4.4.3 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<sup>-1</sup> 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。

#### 3.4.4.4 安全性評価結果

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<sup>-1</sup> 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])。さらに、制御型高階差分攻撃で、補助関数を除いた 10 段 Camellia が解読可能である事が示されている [9]。詳細評価結果の概要は以下の通りである。

- FL/FL<sup>-1</sup> 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。
- 6 段 Camellia 暗号は、制御型高階差分攻撃で 2<sup>17</sup> 平文、2<sup>22</sup> 計算量で解読可能。同じ手法で、補助関数無しの Camellia は、2<sup>21</sup> 平文、2<sup>258</sup> 計算量\*1で解読可能であり、これは、秘密鍵総当たりよりも少ない計算量である。
- 全単射なラウンド関数を用いていることから、FL/FL<sup>-1</sup> 関数を除いた変形 Camellia 6 段で秘密鍵総当た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<sup>-1</sup> 関数を除いた変形 Camellia 8 段が、秘密鍵総当たりよりも少ない計算量で鍵を推定することが可能であろう。
- 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。

#### 3.4.4.5 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。評価結果は表 3.55 および 3.56 の通りである。

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 Pentium III (800MHz) 上で 276Mbps、Pentium Pro (200MHz) 上で 308cycles 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

\*1 ラウンド関数の計算回数。



表 3.55: データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	326 / 327	326 / 328
2 回目	326 / 327	326 / 327
3 回目	326 / 327	326 / 327
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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	355 / 360	355 / 357
2 回目	355 / 358	355 / 358
3 回目	355 / 357	355 / 357
Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	282 / 288	282 / 288
2 回目	282 / 289	282 / 288
3 回目	282 / 288	282 / 289

ROM: 1,698bytes

RAM: 62bytes

#### 3.4.4.6 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  $\mu$ CMOS ASIC Design Library

論理合成ツール Design Compiler .2000.05-1

動作条件 0 ~ 70 度, 3.3V  $\pm$ 5%

ハードウェアの評価結果を表 3.57 に示す。

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

表 3.56: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	467 / 487	474 / 493
2 回目	467 / 487	474 / 494
3 回目	467 / 487	474 / 493
Ultra SPARC Ili (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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	403 / 408	403 / 407
2 回目	403 / 407	403 / 407
3 回目	403 / 408	403 / 408
Alpha 21264 (463MHz)		
言語	アセンブラ	
プログラムサイズ	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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	448 / 454	435 / 439
2 回目	448 / 454	435 / 439
3 回目	448 / 455	435 / 439

表 3.57: ハードウェアの評価結果

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

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 (0.35  $\mu\text{m}$  CMOS ASIC)。スループット: 212.2Mbits/s

#### 3.4.4.7 補足事項

最近、提案者以外からも Camellia 暗号に関する実装技術に関する検討がなされていて、回路規模及び処理性能での改善が見られている [5]。また、その安全性に対する考察も引き続きなされていて評価の精密化が進展しているが、その安全性に特段の問題点が生じている訳ではない [6, 7, 8]。

### 参考文献

- [1] M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta, and T. Matsumoto, A New 128-bit Block Cipher E2, Technical Report ISEC98-12, IEICE, 1998
- [2] M. Matsui, New Block Encryption Algorithm MISTY, In E. Biham, editor, Fast Software Encryption — 4th International Workshop, FSE97, Vol.1267, LNCS, 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, SAC2000 pp.41-54, 2000 (日本語版は、128 ビットブロック暗号 Camellia, 信学技報 ISEC2000-6, 2000 年 5 月)
- [4] 渋谷、下山、辻井, Byte-Oriented な暗号に対する Truncated Linear Attack, SCIS2001, pp.591-596, Jan.2001
- [5] 佐藤、森岡、張, 128 ビットブロック暗号 Camellia の小型ハードウェアアーキテクチャ, SCIS2002, pp.595-598, Jan.2002
- [6] Y. Yeom, S. Park, and I. Kim, On the Security of CAMELLIA against the Square Attack, FSE2002, Feb.2002
- [7] T. Shirai, S. Kanamaru, and G. Abe, Improved Upper Bound of Differential and Linear Characteristic Probability for Camellia, FSE2002, Feb.2002
- [8] 武田、金子, Camellia の制御型高階差分攻撃に関する一考察, SCIS2002, pp.915-919, Jan. 2002
- [9] T. Kawabata and T. Kaneko, "A Study on Higher Order Differential Attack of Camellia," 2nd NESSIE workshop, Sept. 2001

### 3.4.5 Hierocrypt-3

#### 3.4.5.1 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 128 ビットであって、三つの鍵長 (128/192/256 ビット) をサポートする。鍵長と同程度に期待される安全性と、効率的なソフトウェア/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。

### 3.4.5.2 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させて拡大鍵を生成する。復号時にも on-the-fly での鍵設定の初期遅延が小さくなるよう、中間鍵列が途中で逆転して戻ってくる折り返し型の構造を採用した。
- 段数は鍵長に依存し、鍵長 128、192、256 ビットに対し各々 6、7、8 段である。

### 3.4.5.3 その他

Hierocrypt は東芝が開発した共通鍵ブロック暗号のファミリーに付けられた名前。このファミリーには、ブロック長 128 ビットの Hierocrypt-3 とブロック長 64 ビットの Hierocrypt-L1 があり、いずれもデータランダム化部が入れ子型 SPN 構造と呼ばれる SPN 構造の一種で設計されているという共通点がある。

### 3.4.5.4 安全性評価結果

現時点 (2002 年 3 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るようになり、今後の解析結果にも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。特に差分解読法、線形解読法については信頼性の高い評価を行っている。また、新しい評価技術 [10, 12] に対し継続的に、新しい評価手法を取り入れた、あるいは改良した評価による安全性評価結果の更新を行っている [13, 9]。Hierocrypt-3 の設計者が最も注目する攻撃法のひとつである SQUARE 攻撃については、3.5 段での解読可能性が指摘されている [11]。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の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 [5]。また、アバランシュ性の検証では、鍵スケジュール部、ラウンド関数で偏りがあることが示された。また設計指針の一つとして、「MDS 行列と S-box を組合せたときの多項式表現の項数が最大であること」があっ

たが、これに反して評価結果として、多項式表現の項数は(比較的大きな値をとってはいるものの)最大値でないことが確認されている。その他、補間攻撃に関する検討 [8]、不可能差分攻撃に関する検討 [6]、実験的な乱数検定結果 [15] などが新しい結果が発表された。しかし、これら評価どれもが仕様どおりの Hierocrypt-3 の安全性を脅かすものではない。安全性に関する結果をまとめた情報は NESSIE プロジェクトのいくつかの文書で確認することができる [14, 17]。SPN 構造、S-Box 評価、MDS などほとんどの要素技術がこれまでの暗号学の研究結果を踏まえた設計となっており、個々については今後の明らかかつ致命的な欠点は起らないと考えられる。最後に、設計指針とアルゴリズムは直感的、理論的に結びつくものであり、設計者が落とし戸を意図的に組み込んだとは考えにくい。

### 3.4.5.5 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。評価結果は表 3.58 および 3.59 の通りである。

■備考 Ultra SPARC Iii と Alpha 21264 の測定において、(カッコ)内の値は応募者による測定プログラムの改変した場合の測定値。測定プログラムは汎用性を持たせるため巨大なバッファ領域を確保しているが、その領域を必要な分だけ取るように改変した。速度評価の主旨を違えるような改変は行っていないことは確認済み。

表 3.58: データランダム化部速度測定結果 (単位 [cycles/block])

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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	404 / 406	426 / 428
2 回目	404 / 406	426 / 428
3 回目	404 / 406	426 / 428
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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	511 (471) / 554 (473)	759 (612) / 826 (616)
2 回目	510 (471) / 556 (473)	758 (612) / 826 (616)
3 回目	510 (471) / 555 (473)	757 (612) / 826 (616)
Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	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 -O3	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	420 (399) / 424 (406)	427 (386) / 429 (393)
2 回目	420 (399) / 424 (406)	427 (386) / 430 (394)
3 回目	420 (399) / 423 (407)	427 (386) / 430 (393)

表 3.59: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	726 / 728	1345 / 1358
2 回目	726 / 729	1344 / 1357
3 回目	726 / 728	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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	823 (761) / 828 (822)	2673 (2612) / 2684 (2627)
2 回目	823 (761) / 828 (821)	2671 (2611) / 2683 (2627)
3 回目	824 (761) / 828 (823)	2670 (2610) / 2683 (2627)
Alpha 21264 (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 -O3	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	675 (668) / 679 (672)	1130 (1130) / 1142 (1141)
2 回目	675 (668) / 678 (673)	1130 (1130) / 1142 (1142)
3 回目	675 (668) / 679 (672)	1130 (1130) / 1142 (1142)

### 3.4.5.6 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。

表 3.60: HW 実装評価結果

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。

0.14 $\mu$ m CMOS ASIC 897Mbps 81.5 キロゲート  
0.14 $\mu$ m CMOS ASIC 84.6Mbps 26.7 キロゲート  
ALTERA Max+plusII 51.0Mbps 11 キロセル (Flex 10K ファミリー)  
ALTERA Max+plusII 4.1Mbps 6.3 キロセル (Flex 10K ファミリー)

## 参考文献

- [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.
- [4] 清水秀夫, 佐野文彦, 本山雅彦, 大熊建司, 川村信一, 「SPN 型ブロック暗号の実装について」, 電子情報通信学会技術研究報告, ISEC2001-55, 2001.
- [5] S. Furuya and V. Rijmen, “Observations on Hierocrypt-3/L1 Key-scheduling Algorithms,” Proceedings of the second open NESSIE Workshop, 2001.
- [6] C. M.J. Kim and K. Kim, “Impossible Differential Cryptanalysis of Hierocrypt-3 Reduced to 3 Rounds,” Proceedings of the second open NESSIE Workshop, 2001.
- [7] F. Sano, K. Ohkuma, H. Shimizu, M. Motoyama, S. Kawamura, “Efficient Implementation of Hierocrypt,” Proceedings of the second open NESSIE Workshop, 2001.
- [8] 古屋聡一, 櫻井幸一, 「SPN 構造をもつブロック暗号の代数近似について」第 4 回コンピュータセキュリティシンポジウム (CSS2001) 講演予稿集, CSS2001, 6B-1, 2001.
- [9] 大熊建司, 佐野文彦, 清水秀夫, 川村信一, 「入れ子型 SPN 構造の証明可能安全性に関する補足事項」, 2002 年暗号と情報セキュリティシンポジウム SCIS2002 講演予稿集, SCIS2002 5B-2, 2002.
- [10] L. Keliher, H. Meijer, and S. Tavares, “Improving the Upper Bound on the Maximum Average Linear Hull Probability for Rijndael,” Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001 Toronto, LNCS 2259, Springer-Verlag, 2001.
- [11] P. S.L.M. Barreto, V. Rijmen, J. Nakahara Jr., B. Preneel, J. Vandewalle, and H. Y. Kim, “Improved SQUARE Attacks Against Reduced-Round HIEROCRYPT,” Preproceedings of FSE2001, 8th Fast Software Encryption Workshop, Yokohama, 2001.
- [12] L. Keliher, H. Meijer, S. Tavares, “Dual of New Method for Upper Bounding the Maximum Average Linear Hull Probability for SPNs,” IACR’s ePrint archive, 2001/033, available at <http://eprint.iacr.org/>.
- [13] K. Ohkuma, H. Shimizu, F. Sano, S. Kawamura, “Security Assessment of Hierocrypt and Rijndael against the Differential and Linear Cryptanalysis (Extended Abstract),” IACR’s ePrint archive, 2001/070, available at <http://eprint.iacr.org/>.
- [14] B. Van Rompay, V. Rijmen, J. Nakahara Jr., “A first report on CS-Cipher, Hierocrypt, Grand Cru, SAFER++, and SHACAL,” Public reports of NESSIE project, NES/DOC/KUL/WP3/006/1, available at <http://www.cosic.esat.kuleuven.ac.be/nessie/reports/>.
- [15] Yan Braziler, “The statistical evaluation of the NESSIE submission Hierocrypt-

- 3,” Public reports of NESSIE project, NES/DOC/TEC/WP3/021/1, available at <http://www.cosic.esat.kuleuven.ac.be/nessie/reports/>.
- [16] Yan Braziler, “The statistical evaluation of the NESSIE submission Hierocrypt-L1,” Public reports of NESSIE project, NES/DOC/TEC/WP3/022/1, available at <http://www.cosic.esat.kuleuven.ac.be/nessie/reports/>.
- [17] B. Preneel, B. Van Rompay, L. Granboulan, G. Martinet, S. Murphy, R. Shipsey, J. White, M. Dichtl, P. Serf, M. Schafheutle, E. Biham, O. Dunkelmann, M. Ciet, J.-J. Quisquater, F. Sica, L. Knudsen, H. Raddum, “NESSIE Phase I: Selection of Primitives,” NESSIE deliverables, available at <http://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/>.

## 3.4.6 RC6

### 3.4.6.1 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[1]。設計には、その前身であるRC5の思想を受け継ぎ、簡潔な構造で、高速で効率的な実装や広い範囲の解析が可能になる事を目指している。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。

### 3.4.6.2 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ェ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。

### 3.4.6.3 安全性評価結果

RC6はAES提案暗号として、評価を受け、詳細評価対象の5暗号の1つに選ばれている。今回のCRYPTRECの評価も受け、これらにおいて、提案版のRC6の欠陥は報告されておらず、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ないが、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 [2]。なお、複数の線形近似式を使用した攻撃において、 $2^{-90}$  の割合で存在する弱鍵の場合、 $2^{126.9}$  の平文と  $2^{192.9}$  の計算量で、18段 RC6 の、鍵推定が可能である [4]。

高階差分やその他の攻撃の中で、RC6 に対し、効果を上げているのは、カイ 2 乗攻撃である。この攻撃はカイ 2 乗統計量を使う攻撃であり、それによると 15 段が、 $2^{119}$  の選択平文と  $2^{215}$  の計算量で、 $2^{138}$  のメモリを使い鍵の推定が可能である [3]。これらの範囲の段数では RC6 は期待する暗号強度に達していない。しかし、平文数等の数字は、通信速度や計算機能力が毎年 10 倍で上昇しても、今後 10 年間はある得ない環境であり、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あるが、これら攻撃法に対する防御方法の研究は、まだ途上であり、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。

#### 3.4.6.4 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。評価結果は表 3.61 および 3.62 の通りである。

**備考:** Pentium III で測定したコードは Microsoft Windows9X 用の製品プログラム、Ultra SPARC Ii で測定したコードは SUN Solaris 用の製品プログラムを性能評価テスト仕様にあわせて修正したものであり、ベースとなった製品 (「BSAFE-Crypto-C5.1」) は現在販売中である。

Pentium III における暗号化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、Pentium III の測定結果で、最も遅い暗号に近い。UltraSPARC Ii 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ないことである。前者はアセンブリ言語、後者は C 言語で記述してある。

■IC 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、IC カード、DSP による実装に関し以下の特徴が記載されている。

**Java** 暗号処理の簡易性は Java における、コードの大きさ、パフォーマンス、およびダイナミック RAM の量といった点に反映される。AES の評価過程で行われた各調査

表 3.61: データランダム化部速度測定結果 (単位 [cycles/block])

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	1200 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 (マイクロソフト C コンパイラ)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	258 / 260	262 / 266
2 回目	258 / 260	262 / 265
3 回目	258 / 259	262 / 265
Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	3940 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	xo5 (WS Compiler C/SPARC オプティマイズ 5)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	2048 / 2088	2024 / 2076
2 回目	2047 / 2088	2023 / 2074
3 回目	2048 / 2089	2026 / 2077

表 3.62: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

Pentium III (650MHz)		
言語	アセンブラ	
プログラムサイズ	1200 Byte (暗号化/復号) 1500 Byte(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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1631 / 1644	1633 / 1639
2 回目	1630 / 1645	1633 / 1643
3 回目	1630 / 1642	1633 / 1640
Ultra SPARC IIi (400MHz)		
言語	ANSI C	
プログラムサイズ	3940 Byte (暗号化/復号) 2196 Byte(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)	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	4078 / 4111	4026 / 4054
2 回目	4078 / 4111	4024 / 4055
3 回目	4075 / 4112	4019 / 4054

では、RC6 は Java 環境の中で、顕著なパフォーマンスを示している。

**IC カード** RC6 のパフォーマンスは、ARM チップや他の高機能プロセッサを採用したスマート・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。

### 3.4.6.5 ハードウェア (HW) 実装評価

HW 実装評価の結果は表 3.63 の通りである。0.35 $\mu$ m 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。

表 3.63: HW 実装評価

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 <sup>[1]</sup>
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。

表 3.64: FPGA<sup>[5]</sup>

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

表 3.65: ASIC<sup>[4]</sup>

暗号化	XCV1000	0.5 $\mu$ m	反復方式
暗号化	XCV1000	0.5 $\mu$ m	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] L.R. Knudsen and W. Meier, Correlations in RC6 with a reduced number of rounds. FSE2000, LNCS 1978, pp.94-108, 2001
- [4] T. Shimoyama, M. Takenaka, and T. Koshihara, Multiple Linear Cryptanalysis of a reduced round RC6, SCIS2002, Proceedings of the 2002 Symposium on Cryptography and Information Security, pp.931-936, 2002 (also presented at FSE2002)
- [5] A. Elbirt, W. Yip, B. Chetwynd, and C. Parr, An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists, Proceedings of 3rd AES conference, pp.13-27, (2000)
- [6] B. Weeks, M. Bean, T. Rozyłowicz, and C. Ficke, Hardware performance simulations of Round 2 AES algorithms, Proceedings of 3rd AES conference, pp.286-304, (2000)

### 3.4.7 SC2000

#### 3.4.7.1 技術概要

- 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128ビットデータ入出力、128、192、256ビット鍵長を持つ共通鍵ブロック暗号である。
- 暗号全体の構造はFestel構造と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。

#### 3.4.7.2 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、256ビット鍵の場合は32ビット拡大鍵が64個である。

■**復号関数** 32ビット×4の入力暗号文データを、入力の拡大鍵テーブルを用いて復号し、32ビット×4のデータを復号文として出力する。内部関数として32ビット×4の入出力であるI関数、 $B^{-1}$ 関数、R関数を持つ。このうちI関数、R関数はデータランダム化部のものと同じで、 $B^{-1}$ 関数はB関数の逆関数である。各関数をI-B-1-I-R×Rのように接続しこの処理を繰り返す。

■**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。

### 3.4.7.3 安全性評価結果

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とする場合は  $2^{-150}$  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とする場合は  $2^{-150}$  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、高階差分解読法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 128 ビット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、重複する場所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、小さいコストに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。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。2001 年 1 月以降に [4, 5, 6, 7] が発表されている。

#### 3.4.7.4 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。評価結果は表 3.66 および 3.67 の通りである。

表 3.66: データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	389 / 391	408 / 410
2 回目	388 / 392	408 / 411
3 回目	388 / 391	408 / 411
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 -x05	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	310 (275) / 313 (277)	309 (283) / 312 (286)
2 回目	310 (276) / 313 (278)	309 (283) / 312 (287)
3 回目	310 (276) / 314 (279)	309 (282) / 312 (285)
Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	39845 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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	289 (262) / 297 (276)	282 (275) / 296 (289)
2 回目	289 (262) / 297 (277)	282 (275) / 288 (289)
3 回目	289 (262) / 296 (276)	282 (275) / 288 (289)

**備考** Ultra SPARC III と Alpha 21264 の測定において、(カッコ)内の値は応募者による測定プログラムの改変した場合の測定値。測定プログラムは汎用性を持たせるため

表 3.67: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	800 / 803	818 / 822
2 回目	800 / 803	818 / 821
3 回目	800 / 803	818 / 819
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 -x05	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	623 / 627	618 / 622
2 回目	623 / 627	618 / 622
3 回目	623 / 627	618 / 622
Alpha 21264 (463MHz)		
言語	ANSI C	
プログラムサイズ	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	
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	572 / 578	586 / 594
2 回目	572 / 578	586 / 595
3 回目	572 / 578	586 / 594

巨大なバッファ領域を確保しているが、その領域を必要な分だけ取るように改変した。速度評価の主旨を違えるような改変は行っていないことは確認済み。

復号の処理時間を、暗号の処理時間と比べると、Pentium III と Alpha 21264 においては数%大きくなり、UltraSPARC Iii においては逆に数%少なくなった。これらは特に問題になるほどは大きくない。

設計者らにより Pentium III 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 Intel 8051 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。

### 3.4.7.5 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] 下山、屋並、横山、武仲、伊藤、矢嶋、鳥居、田中、共通鍵ブロック暗号 SC2000, 信学技報 ISEC2000-72, 2000.
- [3] 屋並、下山、共通鍵ブロック SC2000 の差分/線形探索、Proceedings of the SCIS 2001, 2001.
- [4] T. Shimoyama, H. Yanami, K. Yokoyama, M. Takenaka, K. Itoh, J. Yajima, N. Torii, H. Tanaka, Block Cipher SC2000, Proceedings of FSE 2001, pp.326-340, 2001.
- [5] 屋並、下山, SC2000 の差分/線形探索 (II), 信学技報 ISEC2001-10, 2001.
- [6] 矢嶋、武仲、小柴、鳥居, 共通鍵ブロック暗号 SC2000 の乱数性, 信学技報 ISEC2001-11, 2001.
- [7] H. Raddum and L.R. Knudsen, A Differetial Attack on Reduced-Round SC2000, Proceedings of SAC2001, pp.207-215, 2001.

## 3.5 スクリーニング評価対象暗号の評価

### 3.5.1 MUGI

#### 3.5.1.1 技術概要

MUGI はストリーム暗号向けの疑似乱数生成器であり、秘密鍵 128 ビット、初期ベクトル (公開値)128 ビットをパラメータに持つ。

提案者の主張によると MUGI は、1998 年に Daemen と Clapp が提案した PANAMA を参考に設計されている。PANAMA は、疑似乱数生成器の設計法の 1 つである線形フィードバックシフトレジスタではなく、ブロック暗号と同じ原理に基づく設計を行っている。このため、ブロック暗号の設計、評価手法を適用しやすいと考えられる。また、基本的なアイデアが単純であり、同様の構造を持つバリエーションを設計しやすいことも特徴として挙げられる。しかし、一方で、PANAMA は従来にない設計を採用しており、安全性解析が困難であるという課題がある。

このため、設計者は、MUGI の設計方針として、PANAMA と同様の構造を持ち、安全性が、既存のブロック暗号の解析手法がより適用しやすいような設計と評価を目指した、と主張している。

設計思想の 1 つに既存の評価された暗号技術を再利用することを掲げている。特に MUGI では、評価が十分に行われている AES の構成要素 (例えば置換表 S-box) を採用している。



### 3.5.1.2 スクリーニング評価結果

■**結論** 安全性について、いまのところ問題は見つかっていない。1998年に発表された PANAMA の改良であるとはいえ、発表されてから日が浅く、さらなる安全性及び実装性の評価が必要と考えられる。

#### ■スクリーニング評価の要点 評価において

1. Test Vector の不一致
2. 自己評価書における 2000 年度詳細評価対象暗号に対する特長が不十分

の 2 点が問題となった。このため、検討事項として

- (1) Test Vector の不一致の理由は何か? Technical か Editorial ??
- (2) ストリーム暗号としての 2000 年度の詳細評価対象 (MULTI-S01 及び TOYOCRYPT-HS1) に対する特長、とくに内部で乱数生成アルゴリズム PANAMA を使っていることから、MULTI-S01 との比較・優位性、差別化の検討。

があげられた。

■**スクリーニング評価の経緯** 暗号ワークショップにて、上記検討事項 (1)(2) に対する提案者のコメントがあった。特に (1) に関しては、編集上の記載ミスであるとの回答があった。

さらに、SCIS2002 会議にて、提案社により、さらなる自己評価が発表された [1]。ここでは MUGI に対して、差分解読法、線形解読法を用いた再同期攻撃を適用することが検討され、主に  $\rho$  関数の差分、線形特性の評価が行われている。この結果でも、線形解読法に対する耐性を厳密に評価するためにさらなる解析の必要性を提案社自身示唆しているが、安全性について問題があるという結果には至っていない。

提案社の安全性解析をみると、MUGI は、PANAMA 以上に既存のブロック暗号の解析手法がより適用しやすい設計であり、この点は暗号設計評価上、重要な優位性と評価でき、検討課題 (2) の 1 つの答えと判断できる。

また、最近 MUGI を含むある種のストリーム暗号系に対して、解析手法が提案された [2]。この論文では、MUGI への攻撃可能性は示唆されているものの、詳細な検討・実際の解析は与えられていない。

■**スクリーニング評価の結論** このように、MUGI の安全性について、いまのところ問題は見つかっていない。しかし、MUGI に関しては、上記の攻撃をふくむさらなる安全性及び、実装性の評価が必要と考えられる。

## 参考文献

- [1] 渡辺 他, 鍵ストリーム生成器 MUGI の安全性評価 (1) Proc. SCIS2002
- [2] Don Coppersmith, Shai Halevi, Charanjit Jutla, “Cryptanalysis of stream ciphers with linear masking” ePrint@IACR February 16, 2002, available at <http://eprint.iacr.org/>

### 3.5.1.A 各評価者の総合コメント (そのまま抜粋)

■ReportA 詳細評価を実施し, 安全性の評価を行なうべきと判断します.

1. 平成 12 年度詳細評価対象暗号に対する特長について提案者 (会社) が同じである”MULTI-S01”は, 本提案と同じく PANAMA を利用したストリーム暗号ですので, それとの比較 (差別化) は必要です. また, ブロック暗号の段関数に相当する関数を利用していることから, ブロック暗号との性能 (速度) 比較も必要です. AES と同程度とは記述されていますが, 数値比較は行なわれていません.
2. 想定するアプリケーション明記されていませんが, 秘匿に関するほとんどの用途に適用可能と考えられます.

■ReportB 仕様は実装可能なレベルで記述されている。また、設計方針などについても明確に記述されている。安全性評価については、一般的な攻撃法から、構造特有の攻撃まで、様々な観点から行われており、統計的評価も FIPS140-1 をカスタマイズして行われている。実装性評価については、ソフトウェアハードウェア共に、妥当な検討がされており、処理速度や、リソース量、ハード規模などについても妥当な値であると考えられる。ただし、仕様書についてはタイプミスと思われる記述の抜けや誤りがいくつか見られる。また、当方の計算機環境でリファレンスコードを実行したところ、仕様書に記載されているテストベクトルの一つについて、出力値が一致しなかった。平成 12 年度詳細評価対称暗号に対する特長については応募書類に明確な記述がない。平成 12 年度詳細評価対称暗号 MULTI-S01 も PANAMA の構造を採用しているが、本技術では内部に AES の関数を使用している点が特長であると思われる。

■ReportC 本提案方式は、ソフトウェア・ハードウェアのいずれのプラットフォームにおいても高速、または軽量な実装が可能な暗号方式として提案されている。ストリーム暗号でありながら、64 ビット長のブロック単位の処理によって長周期性と高速性の両立を図っている。単にブロック単位の処理を行っているのであれば長周期性に疑問もあるが、データ攪拌部を備えていることから特に問題ないと思われる。ただし、その設計においては十分な注意が必要である。

■ReportD PANAMA の改良を行っている点は理解できるが、安全性等の議論において、詳細評価の必要あり。また、SW,HW 実装においても PANAMA との比較データが欠如している。この点も詳細評価で検討すべき。

## 第4章

# ハッシュ関数の評価

### 4.1 評価方法

#### 4.1.1 ハッシュ関数の評価方法

ハッシュ関数は、ほぼ任意のビット長  $m$  を一定長  $n$  に圧縮する関数である。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ことを許しているため、完全に無衝突であることはあり得ない。そこで、現実的な計算量で衝突が発見されなかった場合、無衝突性を持つと判断する。今回は、ハッシュ関数の応募はなかったため、広く使用されていると判断された技術について、文献等の調査によりその安全性評価を実施した。

### 4.2 総評

#### 4.2.1 ハッシュ関数の評価

##### 4.2.1.1 詳細評価

■**評価対象** 2001年度は以下のハッシュ関数に対し、評価委員会は詳細な評価が必要と判断し評価を行った。

- draft SHA-256, draft SHA-384, draft SHA-512

■**評価内容** 暗号学的ハッシュ関数が満たすべき性質である、一方向性、無衝突性の検討を行う。また、本暗号方式に特化した新たな攻撃による評価、SHA-1やMD型ハッシュとの安全性の観点からの比較、発表された攻撃結果などの文献調査も行う。

■**評価結果** 本評価は評価委員会だけでなく、それぞれ国内外併せて3者程度の研究者に依頼した。その結果「安全性について、いまのところ問題は見つかっていない。」との結論を得た。

#### 4.2.1.2 継続評価

対象のハッシュ関数は RIPEMD-160、SHA-1 である。

表 4.1: 各ハッシュ関数の特徴

	RIPEMD-160	SHA-1
特徴	メッセージダイジェスト長	
	160 ビット	160 ビット
	基本処理単位のビット長	
	512 ビット	512 ビット
	総処理ステップ数	
	160	80
	最大入力可能メッセージ長	
	$2^{64} - 1$ ビット	$2^{64} - 1$ ビット

■**安全性** RIPEMD-160、SHA-1 に対する実用的な安全性を脅かす攻撃方法は報告されていないため、これらのハッシュ関数は暗号の応用分野で使うのに十分安全であると考えられる。しかし、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  $n$  ビットである場合、Birthday 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ります。最近の研究では少なくとも  $n = 160$  ビット以上必要であると考えられている。

### 4.3 詳細評価対象暗号 (個別暗号) の評価

#### 4.3.1 draft SHA-256/384/512

##### 4.3.1.1 draft SHA-256/384/512 の技術概要

NIST は 1992 年に 160 ビットハッシュ関数 SHA と 1994 年にその改定版の SHA-1 を FIPS-180 で提案している [1]。さらに、NIST は SHA-1 とほぼ同様な設計原理に基づいているが、SHA-1 よりも長いハッシュ関数、256 ビットハッシュ関数 SHA-256、384 ビットハッシュ関数 SHA-384、512 ビットハッシュ関数 SHA-512 の三種類を提案している。

新しい標準ハッシュ関数として三種類のハッシュ関数 SHA-256、SHA-384、SHA-512 を導入した最大の理由は、これらのハッシュ関数に対する衝突攻撃のセキュリティレベルは各々 128、192、256 ビットであるので、最近採用された、三種類のブロック暗号 AES-128、AES-192、AES-256 と対応させるためである。

## 4.3.1.2 SHA-256 の技術仕様

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

$$M = x || 1 || 0^k || \ell \quad (4.1)$$

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)

$$M^{(i)} = M_0^{(i)} || M_1^{(i)} || \cdots || M_{15}^{(i)} \quad (4.2)$$

からなる。

- (iii) バッファ変数を初期ハッシュ値で初期値化

$$\left. \begin{array}{l} a_0 = H_1^{(0)}, \quad b_0 = H_2^{(0)} \\ c_0 = H_3^{(0)}, \quad d_0 = H_4^{(0)} \\ e_0 = H_5^{(0)}, \quad f_0 = H_6^{(0)} \\ g_0 = H_7^{(0)}, \quad h_0 = H_8^{(0)} \end{array} \right\} \quad (4.3)$$

する。ただし、初期ハッシュ値  $H_j^{(0)}$  ( $1 \leq j \leq 8$ ) は

$$\left. \begin{array}{l} H_1^{(0)} = 6A09E667, \quad H_2^{(0)} = BB67AE85 \\ H_3^{(0)} = 3C6EF372, \quad H_4^{(0)} = A54FF53A \\ H_5^{(0)} = 510E527F, \quad H_6^{(0)} = 9B05688C \\ H_7^{(0)} = 1F83D9AB, \quad H_8^{(0)} = 5BE0CD19 \end{array} \right\} \quad (4.4)$$

である。

- (iv)  $0 \leq t \leq 63$  に対して以下の計算を繰り返す。

$$\left. \begin{array}{l} T1_t = h_t + \Sigma_1^{256}(e_t) + \text{CH}(e_t, f_t, g_t) + K_t^{256} + W_t \\ T2_t = \Sigma_0^{256}(a_t) + \text{Maj}(a_t, b_t, c_t) \\ h_{t+1} = g_t \\ g_{t+1} = f_t \\ f_{t+1} = e_t \\ e_{t+1} = d_t + T1_t \\ d_{t+1} = c_t \\ c_{t+1} = b_t \\ b_{t+1} = a_t \\ a_{t+1} = T1_t + T2_t \end{array} \right\} \quad (4.5)$$

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。

また、 $\text{CH}(\cdot)$ ,  $\text{Maj}(\cdot)$ ,  $\Sigma_0^{256}(\cdot)$ ,  $\Sigma_1^{256}(\cdot)$  は各々次式で定義する、32 ビット変数を入力、出力とする関数

$$\left. \begin{array}{l} \text{CH}(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z) \\ \text{Maj}(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \Sigma_0^{256}(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x) \\ \Sigma_1^{256}(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x) \end{array} \right\} \quad (4.6)$$

であり、 $S^n(x)$  は 32 ビット word  $x$  を  $n$  ビット右へ巡回シフトすることを意味する。また、 $K_t^{256}$ , ( $0 \leq t \leq 63$ ) は定数 (FIPS-180 参照) であり、拡張メッセージ  $W_t$  ( $0 \leq t \leq 63$ ) は、次式で定義する SHA-256 メッセージスケジュール関数

$$W_t = \begin{cases} M_t^{(i)}, & 0 \leq t \leq 15 \\ \sigma_1^{256}(W_{t-2}) + W_{t-7} + \sigma_0^{256}(W_{t-15}) + W_{t-16}, & 16 \leq t \leq 63 \end{cases} \quad (4.7)$$

で計算される。ただし、 $\sigma_0^{256}(\cdot)$ ,  $\sigma_1^{256}(\cdot)$  は次式で定義する、32 ビット word を入力変数、出力変数とする関数

$$\left. \begin{aligned} \sigma_0^{256}(x) &= S^7(x) \oplus S^{18}(x) \oplus R^3(x) \\ \sigma_1^{256}(x) &= S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x) \end{aligned} \right\}$$

であり、 $R^n(x)$  は 32 ビット word  $x$  を  $n$  ビット右へシフトすることを意味する。

(v) 中間ハッシュ値を

$$\left. \begin{aligned} H_1^{(i)} &= a_{64} + H_1^{(i-1)}, & H_2^{(i)} &= b_{64} + H_2^{(i-1)} \\ H_3^{(i)} &= c_{64} + H_3^{(i-1)}, & H_4^{(i)} &= d_{64} + H_4^{(i-1)} \\ H_5^{(i)} &= e_{64} + H_5^{(i-1)}, & H_6^{(i)} &= f_{64} + H_6^{(i-1)} \\ H_7^{(i)} &= g_{64} + H_7^{(i-1)}, & H_8^{(i)} &= h_{64} + H_8^{(i-1)} \end{aligned} \right\} \quad (4.8)$$

で計算する。8 個の 32 ビットハッシュ値をパディングした 256 ビット中間ハッシュ値を

$$H^{(i)} = H_1^{(i)} || H_2^{(i)} || \cdots || H_8^{(i)} \quad (4.9)$$

と表し、上記 (iii)-(iv) を SHA-256 圧縮関数  $C^{256}(\cdot)$  で表すと、上記手続きは以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}^{256}(H^{(i-1)}), \quad 1 \leq i \leq N \quad (4.10)$$

で纏めて表現できる。 $H^{(N)}$  がメッセージ  $M$  のハッシュ値である。

#### 4.3.1.3 SHA-512 の技術仕様

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 ビット長のパディングされたメッセージ

$$M = x || 1 || 0^k || \ell \quad (4.11)$$

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) バッファ変数を初期ハッシュ値で初期値化

$$\left. \begin{aligned} a_0 &= H_1^{(0)}, & b_0 &= H_2^{(0)} \\ c_0 &= H_3^{(0)}, & d_0 &= H_4^{(0)} \\ e_0 &= H_5^{(0)}, & f_0 &= H_6^{(0)} \\ g_0 &= H_7^{(0)}, & h_0 &= H_8^{(0)} \end{aligned} \right\} \quad (4.12)$$

する。ただし、初期ハッシュ値  $H_j^{(0)}$  ( $1 \leq j \leq 8$ ) は

$$\left. \begin{aligned} H_1^{(0)} &= 6A09E667F3BCC908, & H_2^{(0)} &= BB67AE8584CAA73B \\ H_3^{(0)} &= 3C6EF372FE94F82B, & H_4^{(0)} &= A54FF53A5F1D36F1 \\ H_5^{(0)} &= 510E527FADE682D1, & H_6^{(0)} &= 9B05688C2B3E6C1F \\ H_7^{(0)} &= 1F83D9ABFB41BD6B, & H_8^{(0)} &= 5BE0CD19137E2179 \end{aligned} \right\} \quad (4.13)$$

である。

(iv)  $0 \leq t \leq 79$  に対して以下の計算を繰り返す。

$$\left. \begin{aligned} T1_t &= h_t + \Sigma_1^{512}(e_t) + \text{CH}(e_t, f_t, g_t) + K_t^{512} + W_t \\ T2_t &= \Sigma_0^{512}(a_t) + \text{Maj}(a_t, b_t, c_t) \\ h_{t+1} &= g_t \\ g_{t+1} &= f_t \\ f_{t+1} &= e_t \\ e_{t+1} &= d_t + T1_t \\ d_{t+1} &= c_t \\ c_{t+1} &= b_t \\ b_{t+1} &= a_t \\ a_{t+1} &= T1_t + T2_t \end{aligned} \right\} \quad (4.14)$$

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。

また、 $\text{CH}(\cdot)$ ,  $\text{Maj}(\cdot)$ ,  $\Sigma_0^{512}(\cdot)$ ,  $\Sigma_1^{512}(\cdot)$  は各々次式で定義する、64 ビット word を入力変数、出力変数とする関数

$$\left. \begin{aligned} \text{CH}(x, y, z) &= (x \wedge y) \oplus (\neg x \wedge z) \\ \text{Maj}(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\ \Sigma_0^{512}(x) &= S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x) \\ \Sigma_1^{512}(x) &= S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x) \end{aligned} \right\} \quad (4.15)$$

であり、 $S^n(x)$  は 64 ビット word  $x$  を  $n$  ビット右へ巡回シフトすることを意味する。また、 $K_t^{512}$  ( $0 \leq t \leq 63$ ) は定数 (FIPS180-2 参照) であり、拡張されたメッセージ  $W_t$  ( $0 \leq t \leq 79$ ) は、次式で定義する SHA-512 メッセージスケジュール関数

$$W_t = \begin{cases} M_t^{(i)}, & 0 \leq t \leq 15 \\ \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16}, & 16 \leq t \leq 79 \end{cases} \quad (4.16)$$

で計算される。ただし、 $\sigma_0^{512}(\cdot)$ ,  $\sigma_1^{512}(\cdot)$  は次式で定義する、64 ビット変数を入力、出力とする関数

$$\left. \begin{aligned} \sigma_0^{512}(x) &= S^1(x) \oplus S^8(x) \oplus R^7(x) \\ \sigma_1^{512}(x) &= S^{19}(x) \oplus S^{61}(x) \oplus R^6(x) \end{aligned} \right\}$$

であり、 $R^n(x)$  は 64 ビット word  $x$  を  $n$  ビット右へシフトすることを意味する。

(v) 中間ハッシュ値を

$$\left. \begin{aligned} H_1^{(i)} &= a_{64} + H_1^{(i-1)}, & H_2^{(i)} &= b_{64} + H_2^{(i-1)} \\ H_3^{(i)} &= c_{64} + H_3^{(i-1)}, & H_4^{(i)} &= d_{64} + H_4^{(i-1)} \\ H_5^{(i)} &= e_{64} + H_5^{(i-1)}, & H_6^{(i)} &= f_{64} + H_6^{(i-1)} \\ H_7^{(i)} &= g_{64} + H_7^{(i-1)}, & H_8^{(i)} &= h_{64} + H_8^{(i-1)} \end{aligned} \right\} \quad (4.17)$$

で計算する。

8 個の 64 ビットハッシュ値をパディングした 512 ビット中間ハッシュ値を

$$H^{(i)} = H_1^{(i)} || H_1^{(i)} || \dots || H_8^{(i)} \quad (4.18)$$

と表し、上記 (iii)-(iv) を併せて SHA-512 圧縮関数  $C^{512}(\cdot)$  で表すと、上記手続きは簡単に以下の漸化式

$$H^{(i)} = H^{(i-1)} + C_{M^{(i)}}^{512}(H^{(i-1)}), \quad 1 \leq i \leq N \quad (4.19)$$

で纏めて表現できる。 $H^{(N)}$  がメッセージ  $M$  のハッシュ値である。

#### 4.3.1.4 SHA-384 の技術仕様

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  $H^{(0)}$  は SHA-256、SHA-512 のそれと異なり、以下のように

$$\left. \begin{array}{l} H_1^{(0)} = \text{CBBB9D5DC1059ED8}, H_2^{(0)} = \text{629A292A367CD507} \\ H_3^{(0)} = \text{9159015A3070DD17}, H_4^{(0)} = \text{152FEC8F70E5939} \\ H_5^{(0)} = \text{67332667FFC00B31}, H_6^{(0)} = \text{8EB44A8768581511} \\ H_7^{(0)} = \text{DB0C2E0D64F98FA7}, H_8^{(0)} = \text{47B5481dBEFA4FA4} \end{array} \right\} \quad (4.20)$$

に変更する。

2. SHA-512 の 512 ビットの最終ハッシュ値  $H^{(N)}$  の左 384 ビットで打ち切った 384 ビットハッシュ値

$$H^{(N)} = H_1^{(N)} || H_2^{(N)} || \dots || H_6^{(N)} \quad (4.21)$$

を最終ハッシュ値  $H^{(N)}$  として採用するものである。

#### 4.3.1.5 安全性評価

1. SHA-256

- (a) 評価者 1

評価結果概要: SHA-256 の安全性に関して以下に掲げる観点に関して評価を行った。

- i. MD タイプのハッシュ関数に対する Dobbertin[2, 3, 4] や Chabaud-Joux[5] の攻撃法はいずれも SHA-256 には適用することは難しい。
- ii. SHA-1 と比較して繰り返し段数が少ないように見えるし、またその上、公式文書が何も無く僅かな仕様から設計選択に関する選択基準や暗号用変数を再構成することは困難ではあるが、SHA-256 の基本構成部分の最大の特長は、既存ハッシュ関数に対するよりもかなり高い暗号強度を提供している。
- iii. 内蔵している圧縮関数の差分特性に関して調査した結果、考えられる繰り返し特性も各段での圧縮関数に共通な特性のいずれも見出すことはできなかった。
- iv. 段数定数が 2 分割に関して互いに対称となるような簡略版の SHA-256 (modified SHA-256、詳細 (vi) 参照) は安全ではない。

以上の調査結果から、現在既知のあらゆる攻撃法のいずれも SHA-256 には適用できないし、原像 (preimage)、弱い意味の原像 (second preimage) の計算複雑



度を  $2^{256}$  以下に、また通常の Birthday 攻撃である  $2^{256/2} = 2^{128}$  の複雑度以下に帰着可能とする攻撃法は見当たらない。

詳細:

- i. 主に三つの安全性評価基準:
  - (i) 衝突困難性 (collision resistance)
  - (ii) 原像計算困難性<sup>\*1</sup>、一方向性 (preimage resistance、one-wayness)
  - (iii) 弱原像計算困難性<sup>\*2</sup>、弱衝突困難性 (second preimage resistance、weak collision-resistance)
 に関して検討を行った結果、いずれも問題が見付からなかった。
- ii. SHA-1 のアルゴリズムと SHA-256 のそれとの主な相違点は以下の通りである。
  - A. メッセージスケジュール計算において排他的論理和  $\oplus$  の代わりに加法演算  $+$  を採用して計算を複雑にした結果、
    - (i) 差分パターンが線形符号にはなりえないので、解析を困難にしている、
    - (ii) SHA-1 の特性が強化された。
    - (iii) SHA-0、SHA-1 で観測されていた入力語の回転不変性が見られなくなった。
    - (iv) メッセージスケジュール長対作業変数用レジスタ長比 (各圧縮関数計算毎の作業変数の全回転の数) は劣化している。  
この劣化による安全性度の具体的評価法は定かではないが、これは、一見すると SHA-256 の安全性劣化につながると思われる。一方、この比の低下は SHA-256 の作業変数の更新の複雑さの向上と SHA-1 と異なり、二レジスタ変数が各段で修正されていることを考え併せれば補償されているともいえる。
  - B. 状態レジスタ更新関数において、32 ビットの 8 レジスタ ( $a_t, b_t, c_t, d_t, e_t, f_t, g_t, h_t$ ) を使用する SHA-256 は、32 ビットの 5 レジスタ ( $a_t, b_t, c_t, d_t, e_t$ ) を使用する SHA-0 や SHA-1 と類似しているものの以下の点で異なる。
    - (i) ラウンド関数が複雑になり、強力でかつ高速な拡散性を有する。すなわち、多数決関数  $\text{Maj}(\cdot)$  および選択関数  $\text{CH}(\cdot)$  等の非線形関数の両方が各段で適用され、また、二つのレジスタ変数が各段で修正されている
    - (ii) SHA-0、SHA-1 で見られた状態レジスタ更新関数の低い一様性が更に低くなっている。これは少し安全性向上につながるかも知れない。
  - C. 多数決関数および選択関数等の非線形関数、シグマ関数  $\sigma_0(\cdot)$ 、 $\sigma_1(\cdot)$ 、 $\Sigma_0(\cdot)$ 、 $\Sigma_1(\cdot)$ 、定数  $K_t$  等は適切に設計されている。
- iii. ハッシュ関数に対する既存の攻撃法: (i) Dobbertin の衝突探索法、(ii) Chaubaud & Joux の差分攻撃による衝突探索法、に関して検討した結果、メッセージ拡張処理が複雑になっているのでいずれの攻撃法も適用できないので、設計変更は安全性の向上に役立っていることが判った。
- iv. 差分攻撃: 圧縮関数の差分特性を検討した結果、4 段での差分特性は、確率  $2^{-8}$  以下であり、64 段全体では  $2^{(-8) \times 16} = 2^{-128}$  以下となる。これは 256 ビットハッシュ関数の衝突確率と同程度に低くなるので、圧縮関数に関する差分攻撃は適用できないと結論付けられる。
- v. 繰り返し差分攻撃に関して検討した結果、SHA-256 に対して適用できないことが判った。

\*1 ハッシュ値  $h(M)$  が与えられたとき、 $h(M) = h(M')$  を満たす  $M'$  を計算する困難性

\*2 メッセージ  $M$  とそのハッシュ値  $h(M)$  が与えられたとき、 $h(M) = h(M')$  を満たす  $M'$  を計算する困難性

- vi. 極度に対称性のある初期ハッシュ値、定数 ( $H_0^{(0)} = H_1^{(0)} = \dots = H_7^{(0)} \in \Omega_{32}$ ) を選択し、加法演算  $+$  を排他的論理和  $\oplus$  に変更した、簡略化 SHA-256 (modified SHA-256) の場合、衝突困難性が無くなるので安全性が損なわれる。ただし、 $\Omega_{32}$  は

$$\Omega_{32} = \{C \in \{0, 1\}^{32} \mid \exists c \in \{0, 1\}^{16}, C = c \| c\} \quad (4.22)$$

で定義される対称 32 ビット word の集合を意味する。

(b) 評価者 2

評価結果概要:

- i. SHA-256、512 のアルゴリズムは SHA-1 のそれと以下の主な点でかなり異なる。
  - (i) メッセージ拡張処理が複雑になっているので、安全性の向上に役立っている。
  - (ii) 1 ステップで更新される変数が二つになっているので既存の攻撃法の適用を困難にしている。
- ii. Draft FIPS180-2 への公開コメント (Jonsson, Kelsey's comments) に言及している。
- iii. 現在の所、SHA-256 の致命的な欠点は勿論、安全性を疑わせるような点は見当たらない。提案されて日が浅いので、今後も引き続き安全性を検討する必要がある。
- iv. SHA-1 より設計変更した点は幾つかの攻撃に対して安全性の向上に役立っていることが確認できた。
- v. Jonsson のコメントにあるように、NIST が SHA-256、384、512 の安全性評価を公開し、SHA-1 と異なる設計にした理由を明らかにすることが望まれる。

以上の結果、現在の所 SHA-256 の致命的な欠点や安全性を疑わせる点も見当たらないが、提案されて日が浅いので、安全性が十分に研究されているとはいえないので、引き続き安全性を検討する必要がある。

詳細:

- i. 主に四つの安全性評価基準: (i) ランダム性、(ii) Birthday Paradox 攻撃法、(iii) 衝突困難性、(iv) 一方向性に関して検討を行った結果、いずれも問題が見付からなかった。
- ii. 並列処理による高速化の観点から安全性の検討を行った結果、問題が見付からなかった。すなわち、SHA-256/383/512 では、並列化アルゴリズムに基づく Birthday Paradox 攻撃法による安全性の低下が難しくなるように SHA-1 は設計変更されているといえる。
- iii. ハッシュ関数に対する既存の攻撃法: (i) Dobbertin の衝突探索法、(ii) Chaubaud & Joux の差分攻撃による衝突探索法、(iii) 簡略版 (1-round) ハッシュ関数の衝突探索法に関して検討した結果、メッセージ拡張処理が複雑になっているのでいずれの攻撃法も適用できないので、設計変更は安全性の向上に役立っていることが判った。
- iv. ハッシュ関数を鍵付ハッシュのメッセージ認証に利用する場合、extension property 問題を解決する Kelsey 法の安全性の証明がなされるまで、Bellare、Canetti、Krawczyk による、効率的かつ安全性が証明可能な鍵付ハッシュ関数の構成法を利用することに言及している。

2. SHA-384、SHA-512

(a) 評価者 1

評価結果概要: SHA-384、SHA-512 の安全性に関して以下に掲げる観点に関して評価を行った。

- i. MD 型ハッシュ関数に対する Dobbertin[2, 3, 4] や Chaubaud-Joux[5] の攻撃法はいずれも SHA-384 や SHA-512 には適用することは難しい。

- ii. SHA-1 と比較して繰り返し段数が少ないように見えるし、またその上、公式文書が何も無く僅かな仕様から設計選択に関する選択基準や暗号用変数を再構成することは困難ではあるが、SHA-384 や SHA-512 の基本構成部分の最大の特長は、既存ハッシュ関数に対するよりもかなり高い暗号強度を提供している。
  - iii. 内蔵している圧縮関数の差分特性に関して調査した結果、考えられる繰り返し特性も各段での圧縮関数に共通な特性のいずれも見出すことはできなかった。
  - iv. 段数定数が 2 分割に関して互いに対称となるような簡略版の SHA-384/512 (modified SHA-384/512、詳細 (vi) 参照) は安全ではない。
- 以上の調査結果から、現在既知のあらゆる攻撃法のいずれも SHA-384 や SHA-512 には適用できないし、SHA-384 や SHA-512 は原像 (preimage)、弱い意味の原像 (second preimage) の計算複雑度を各々  $2^{384}$  や  $2^{512}$  以下に、また通常の Birthday 攻撃の複雑度 (各々  $2^{384/2} = 2^{192}$  や  $2^{512/2} = 2^{256}$ ) 以下の複雑度に帰着可能とする攻撃法は見当たらない。

詳細:

- i. 主に三つの安全性評価基準:
  - (i) 衝突困難性 (collision resistance)
  - (ii) 原像計算困難性、一方向性 (preimage resistance, one-wayness)
  - (iii) 弱原像計算困難性、弱衝突困難性 (second preimage resistance, weak collision-resistance)

に関して検討を行った結果、いずれも問題が見付からなかった。
- ii. SHA-1 のアルゴリズムと SHA-382 や SHA-512 のそれとの主な相違点は以下の通りである。
  - A. メッセージスケジュール計算において排他的論理和  $\oplus$  の代わりに加法演算  $+$  を採用して計算を複雑にした結果、
    - (i) 差分パターンが線形符号にはなりえないので、解析を困難にしている、
    - (ii) SHA-1 の特性が強化された。
    - (iii) SHA-0、SHA-1 で観測されていた入力語の回転不変性が見られなくなった。
    - (iv) メッセージスケジュール長対作業変数用レジスタ長比 (各圧縮関数計算毎の作業変数の全回転の数) は劣化している。

この劣化による安全性度の具体的評価法は定かではないが、これは、一見すると SHA-382 や SHA-512 の安全性劣化につながると思われる。一方、この比の低下は SHA-382 や SHA-512 の作業変数の更新の複雑さの向上と SHA-1 と異なり、二レジスタ変数が各段で修正されていることを考え併せれば補償されているともいえる。
  - B. 状態レジスタ更新関数において、64 ビットの 8 レジスタ ( $a_t, b_t, c_t, d_t, e_t, f_t, g_t, h_t$ ) を使用する SHA-382 や SHA-512 は 32 ビットの 5 レジスタ ( $a_t, b_t, c_t, d_t, e_t$ ) を使用する SHA-0 や SHA-1 と類似しているものの以下の点で異なる。
    - (i) ラウンド関数が複雑になり、強力でかつ高速な拡散性を有する。すなわち、多数決関数  $\text{Maj}(\cdot)$  および選択関数  $\text{CH}(\cdot)$  等の非線形関数の両方が各段で適用され、また、二つのレジスタ変数  $T1_t, T2_t$  が各段で更新されている。
    - (ii) SHA-0、SHA-1 で見られた状態レジスタ更新関数  $\text{TEMP}_t$  の低い一様性が更に低くなっている。これは少し安全性向上につながるかも知れない。
  - C. 多数決関数  $\text{Maj}(\cdot)$  および選択関数  $\text{CH}(\cdot)$  等の非線形関数、シグマ関数

- $\sigma_0(\cdot), \sigma_1(\cdot), \Sigma_0(\cdot), \Sigma_1(\cdot)$ 、定数  $K_t$  等は適切に設計されている。
- iii. ハッシュ関数に対する既存の攻撃法: (i) Dobbertin の衝突探索法、(ii) Chaubaud & Joux の差分攻撃による衝突探索法、に関して検討した結果、メッセージ拡張処理が複雑になっているのでいずれの攻撃法も適用できないので、設計変更は安全性の向上に役立っていることが判った。
  - iv. 差分攻撃: 圧縮関数の差分特性を検討した結果、4段での差分特性は、確率  $2^{-8}$  以下であり、80段全体では  $2^{(-8) \times 20} = 2^{-160}$  以下となる。これは512あるいは384ビットハッシュ関数の衝突確率よりも低くはないが、同一の低い重み特性をつなぎ併せることによりこの限界値に漸近する全体の差分確率を構成することはできそうにない。また、差分特性は最初の段階での擬衝突 (pseudocollision) を検出だけを意味し、また実際には多重の衝突を検出しなければならないこと等を考え併せれば、SHA-382やSHA-512の圧縮関数に関する差分攻撃は適用できないと結論付けられる。
  - v. 繰り返し差分攻撃に関して検討した結果、SHA-382やSHA-512に対して適用できないことが判った。
  - vi. 極度に対称性のある初期ハッシュ値、定数 ( $H_0^{(0)} = H_1^{(0)} = \dots = H_7^{(0)} \in \Omega_{64}$ ) を選択し、加法演算  $+$  を排他的論理和  $\oplus$  に変更した、簡略化 SHA-512 (modified SHA-512) の場合、衝突困難性が無くなるので安全性が損なわれる。ただし、 $\Omega_{64}$  は

$$\Omega_{64} = \{C \in \{0, 1\}^{64} \mid \exists c \in \{0, 1\}^{32}, C = c \parallel c\} \quad (4.23)$$

で定義される対称64ビット word の集合を意味する。

(b) 評価者2

評価結果概要: SHA-384、512の安全性に関して以下に掲げる観点に関して評価を行った。

- i. SHA-384はSHA-512と本質的に同じであるからSHA-512だけに言及。
- ii. SHA-256と同じコメントで同一の評価。

#### 4.3.1.6 SHA-256/384/512の総合評価結果

1. SHA-256の総合評価結果:

現在広く使用されている160ビットのハッシュ値を出力するSHA-1の安全性が損なわれるとの報告は今のところないが、長期間の使用および今後の計算機向上を見越してSHA-1を設計変更したSHA-256に関しては、その設計基準は明確ではないが、メッセージ拡張処理が複雑になり、ハッシュ関数に対する既存の攻撃法がいずれも適用できない。

また、提案されて日が浅いので今後も引き続き安全性の検討が必要であるものの、SHA-256は現在の所安全であると結論付けられる。

2. SHA-384/512の総合評価結果:

現在広く使用されている160ビットのハッシュ値を出力するSHA-1の安全性が損なわれるとの報告は今のところないが、長期間の使用および今後の計算機向上を見越してSHA-1を設計変更したSHA-384/512に関しては、その設計基準は明確ではないが、メッセージ拡張処理が複雑になり、またハッシュ関数に対する既存の攻撃法がいずれも適用できない。

また、提案されて日が浅いので今後も引き続き安全性の検討が必要であるものの、SHA-384/512は現在の所安全であると結論付けられる。

## 参考文献

- [1] SHA-1: National Institute of Standards and Technology (NIST) draft FIPS 180-1: Secure Hash Standards, April 1994.
- [2] H. Dobbertin, *Cryptanalysis of MD4*, in *Journal of Cryptology*, **11-4**, Autumn, 1998.
- [3] H. Dobbertin, *Cryptanalysis of MD5 Compress*, Presented at the rump session of Eurocrypt'96, May 14, 1996.
- [4] H. Dobbertin, *The status of MD5 after a recent attack*, *CryptBytes*, **2-2**, 1996, pp3-6.
- [5] F. Chaubaud and A. Joux, *Differential Collisions in SHA-0*, extended abstract, in CRYPTO'98, LNCS 1462, pp.56-71, 1998.

## 4.4 監視状態の暗号の評価

### 4.4.1 RIPEMD-160

#### 4.4.1.1 技術概要

RIPEMD-160 は Dobbertin、Bosselaers、Preneel により提案されたハッシュ関数であり、ヨーロッパの RIPE (Race Integrity Primitive Evaluation) プロジェクトの成果の一つである。その後、SHA-1 や RIPEMD-128 などと共に 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 ビットの倍数になるようにパディングされた任意のメッセージを入力として 160 ビットのハッシュ値を出力する。

#### 4.4.1.2 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  $X[0] \sim X[15]$  は定められた順番によって右ラインと左ラインに入力される。

##### (2) 圧縮関数

圧縮関数の計算には 5 つの連鎖変数  $(A, B, C, D, E)$  を用いる。  $A, B, C, D$  の初期値は MD5 と同じ値であり、新しく  $E$  の初期値が定められている。  $(A, B, C, D, E)$  の初期

値  $IV = (h_1, h_2, h_3, h_4, h_5)$  を以下に示す。

$$h_1 = 0x67452301$$

$$h_1 = 0xefcdab89$$

$$h_1 = 0x98badcfe$$

$$h_1 = 0x10325476$$

$$h_1 = 0xc3d2e1f0$$

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。

$$f(x, y, z) = x \oplus y \oplus z$$

$$g(x, y, z) = (x \wedge y) \vee (\bar{x} \wedge z)$$

$$h(x, y, z) = (x \wedge \bar{y}) \oplus z$$

$$k(x, y, z) = (x \wedge y) \vee (y \wedge \bar{z})$$

$$l(x, y, z) = x \oplus (y \vee \bar{z})$$

ここで、記号  $\wedge, \vee$  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。

$$K_L[j] = 0x00000000, \quad K_R[j] = 0x50a28be6, \quad (1 \leq j \leq 16)$$

$$K_L[j] = 0x5a827999, \quad K_R[j] = 0x5c4dd124, \quad (17 \leq j \leq 32)$$

$$K_L[j] = 0x6ed9eba1, \quad K_R[j] = 0x6d703ef3, \quad (33 \leq j \leq 48)$$

$$K_L[j] = 0x8f1bbcdc, \quad K_R[j] = 0x7a6d76e9, \quad (49 \leq j \leq 64)$$

$$K_L[j] = 0xa953fd4e, \quad K_R[j] = 0x00000000, \quad (65 \leq j \leq 80)$$

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  $X \lll s$  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 ( $1 \leq j \leq 16$ )

$FF_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) :$

$$A_L = (A_L + f(B_L, C_L, D_L) + X[i] + K_L[j]) \lll s_L[j] + E_L, \quad C_L = C_L \lll 10$$

$LL_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) :$

$$A_R = (A_R + l(B_R, C_R, D_R) + X[i] + K_R[j]) \lll s_R[j] + E_R, \quad C_R = C_R \lll 10$$

2 ラウンド ( $17 \leq j \leq 32$ )

$GG_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) :$

$$A_L = (A_L + g(B_L, C_L, D_L) + X[i] + K_L[j]) \lll s_L[j] + E_L, \quad C_L = C_L \lll 10$$

$KK_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) :$

$$A_R = (A_R + k(B_R, C_R, D_R) + X[i] + K_R[j]) \lll s_R[j] + E_R, \quad C_R = C_R \lll 10$$

3 ラウンド ( $33 \leq j \leq 48$ )

$HH_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) :$

$$A_L = (A_L + h(B_L, C_L, D_L) + X[i] + K_L[j]) \lll s_L[j] + E_L, \quad C_L = C_L \lll 10$$

$HH_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) :$

$$A_R = (A_R + h(B_R, C_R, D_R) + X[i] + K_R[j]) \lll s_R[j] + E_R, \quad C_R = C_R \lll 10$$

4 ラウンド ( $49 \leq j \leq 64$ )

$KK_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) :$

$$A_L = (A_L + k(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10}$$

$GG_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) :$

$$A_R = (A_R + g(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}$$

5 ラウンド

$LL_L(A_L, B_L, C_L, D_L, E_L, X[i], s_L[j], K_L[j]) :$

$$A_L = (A_L + l(B_L, C_L, D_L) + X[i] + K_L[j])^{<<s_L[j]} + E_L, \quad C_L = C_L^{<<10}$$

$FF_R(A_R, B_R, C_R, D_R, E_R, X[i], s_R[j], K_R[j]) :$

$$A_R = (A_R + f(B_R, C_R, D_R) + X[i] + K_R[j])^{<<s_R[j]} + E_R, \quad C_R = C_R^{<<10}$$

### (3) 出力

出力は基本的に MD5 同様、最後の段階で求められた連鎖変数の値と初期値 IV を加えた後、 $(A, B, C, D, E)$  の 5 つの変数を結合することでハッシュ値を出力するが、2 とのラインを使うため以下のように計算する。

$$A = h_2 + C_L + D_R$$

$$B = h_3 + D_L + E_R$$

$$C = h_4 + E_L + A_R$$

$$D = h_5 + A_L + B_R$$

$$E = h_1 + B_L + C_R$$

#### 4.4.1.3 その他

RIPEND-160 はこれより先の 1995 年に RIPE プロジェクトに提案された RIPEND 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 RIPEND-160 は MD4 をベースとするハッシュ関数のひとつでもある。

#### 4.4.1.4 評価結果

■**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。RIPEND-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。

■**RIPEND-160 固有の攻撃** RIPEND-160 の前身である RIPEND 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 RIPEND-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。RIPEND に対する攻撃は RIPEND-160 には適用できず、RIPEND-160 に対する固有の攻撃はまだ報告されていない。

■**入力値探索の手間** ハッシュ値の長さが  $n$  ビットのハッシュ関数が出力する値のパターンは  $2^n$  通りしか存在しない。従って、特定のハッシュ値を出力する入力値を探索しようとした場合、異なるハッシュ値を出力する  $2^n$  通りの入力値をあらかじめ用意しておけば指定されたハッシュ値に一致する入力値を得ることができる。RIPEMD-160 では  $n = 160$  であり、この方法を適用した場合  $2^{160}$  通りの入力値が必要となるが、これは現在の技術では用意不可能なほど大きい数であると考えられている。

■**衝突発見の手間** ハッシュ値の長さが  $n$  ビットの場合、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、ハッシュ値を十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。

#### 4.4.1.5 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、2000 年度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]

#### 4.4.1.6 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 techniques – 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>



## 4.4.2 SHA-1

### 4.4.2.1 技術概要

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用いて作用させる。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、160 ビットのハッシュ値を出力する。

### 4.4.2.2 技術仕様

SHA-1 は、任意長のメッセージを入力し、160 ビットのハッシュ値を出力する。入力されたデータは 512 ビット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。

#### Step 1: 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  $448 \bmod 512$  ビットとなるように加工する。このとき  $448 \bmod 512$  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。

#### Step 2: 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 をパディングビットのあとに 64 ビットの長さで付加する。ここまでの、元のメッセージは 512 ビット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 ビットのブロック系列  $Y_0, Y_1, \dots, Y_{L-1}$  のように表現でき、合計ビット長は  $L \times 512$  ビットとなる。

#### Step 3: 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 ビットレジスタを使って  $(A, B, C, D, E)$  と表現する。この 160 ビットのバッファは、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。

$$A = 0x67452301$$

$$B = 0xEFCDAB89$$

$$C = 0x98BADCFE$$

$$D = 0x10325476$$

$$E = 0xC3D2E1F0$$

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。

#### Step 4: 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  $f_1, f_2, f_3, f_4$  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  $A, B, C, D, E$  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  $\text{mod } 2^{32}$  で加算される。

Step 5: **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ビットバッファ内にあるハッシュ値を出力する。

#### 4.4.2.3 安全性評価結果

SHA-1のアルゴリズムで示したようにSHA-1の解析のためには圧縮関数だけでなく、入力メッセージを拡張する部分も分析しなければならない。SHA-1の前のバージョンであるSHAは入力の拡張部分が排他的論理和だけで構成されており、その分析に基づいて圧縮関数に対する衝突が発見できた。しかし、このSHAに対する攻撃はメッセージ拡張の部分で1ビットの左巡回シフトを用いるSHA-1には適用できないことが報告されている。現在、SHA-1に対する実用的な攻撃は報告されていないため、暗号の応用分野で使うには安全であると考えられる。しかし、全数探索攻撃に対する安全性は確保しなければならないため、ハッシュ値の長さが  $n$  ビットである場合、Birthday 攻撃により  $2^n$  個のメッセージに対するハッシュ値の中で衝突が見つかる可能性から、ハッシュ値を十分長くする必要があり、ハッシュ値が160ビットであるSHA-1は  $2^{80}$  個のハッシュ値に対して衝突が発見される可能性があることから、将来にわたって安全であるとは保証できない。

## 第5章

# 擬似乱数生成系の評価

## 5.1 評価方法

### 5.1.1 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。

## 5.2 総評

### 5.2.1 擬似乱数生成系

#### 5.2.1.1 スクリーニング評価

■**評価対象** 2001 年度の以下の応募暗号技術に対しスクリーニング評価を行った。

- Creation of intrinsic random numbers with Clutter Box
- FSRansu
- High security ultra mini random number generator

- TAO TIME Cognition Algorithm

■**評価内容** 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。

- 記述の有無、記述内容の論理的整合性/自己完結性の確認。
- 書面上で容易に判明するような欠点の検査。
- 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の確認。

#### ■**評価結果**

**Creation of intrinsic random numbers with Clutter Box** 特殊なハードウェアを必要とする方式であり、提出書類をベースにした評価が困難である。乱数生成のアルゴリズムに関する十分な情報が記載されていない。

**FSRansu** 評価に必要な、参照プログラム、テストベクタ生成プログラムがない。

**High security ultra mini random number generator** 特殊なハードウェアを必要とする方式であり、評価を行うことは困難である。参照プログラムは乱数系列を観測するプログラムであり、提出書類が評価を行うための条件を満足していない。

**TAO TIME Cognition Algorithm** 提案方式は CRYPTREC が公募している擬似乱数生成系と考えられない。

#### 5.2.1.2 継続評価

対象は Pseudo-Random Number Generator based on SHA-1 である。

■**特徴** 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。

■**評価結果** 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ため、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、2002 年以降に FIPS として規格化されるであろう、160 ビット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。

### 5.3 監視状態の暗号の評価

#### 5.3.1 PRNG based on SHA-1

(FIPS186-2: DSS Appendix 3.Random number Generator for the DSA)

### 5.3.1.1 技術概要

Digital Signature Algorithm (DSA) では、メッセージ  $M$  に対してユーザの秘密鍵  $x$  およびメッセージ署名毎の秘密鍵  $k$  や次式の  $r, s$

$$\left. \begin{aligned} r &= (g^k \bmod p) \bmod q, \\ s &= (k^{-1}(\text{SHA-1}(M) + xr)) \bmod q \end{aligned} \right\} \quad (5.1)$$

が必要となる。ただし、 $p, q, g$  は公開パラメータ、 $(kk^{-1}) \bmod q = 1$ ,  $0 < k^{-1} < q$  であり、SHA-1(M) は、FIPS180-1 で規定した Secure Hash Algorithm (SHA) の 160 ビット出力値である。

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 X9.17 の Appendix C の “Financial Institution Key Management (Wholesale)” による 160 ビットの一方方向性関数  $G(t, c)$  ( $t$  は 160 ビット、 $c$  は  $b$  ビットである。 $G(\cdot)$  が SHA-1 による場合、 $160 \leq b \leq 512$ 、 $G(\cdot)$  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 (各々次節の技術仕様の III と IV, I, II で記述)。

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 (May 11, 1993) 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 message digest 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ェア実装、テストの応用等のために、Random Number Generation and Testing を Web 公開している。

### 5.3.1.2 技術仕様

- (I) (I) 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  $\omega_{xkey}$  を選択する。
  - (2) SHS での 512 ビットの初期ハッシュ値  $H_0 || H_1 || \dots || H_4$  を

$$\left. \begin{aligned} H_0^{(0)} &= 67452301 \\ H_1^{(0)} &= \text{EFCDAB89} \\ H_2^{(0)} &= 98BADCFE \\ H_3^{(0)} &= 10325476 \\ H_4^{(0)} &= \text{C3D2E1F0} \end{aligned} \right\} \quad (5.2)$$

とする。

- (3)  $0 \leq j \leq m-1$  として以下の (a)-(d) を繰り返す。  
 (a)  $\omega_j$  を選択 (ユーザーオプション) する。  
 (b)  $c_j = (\omega_{\text{xkey}} + \omega_j) \bmod 2^b$ ,  $160 \leq b \leq 512$   
 (c)  $x_j = G(t, c_j) \bmod q$   
 (d)  $\omega_{\text{xkey}} = (1 + \omega_{\text{xkey}} + x_j) \bmod 2^b$

(II) 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  $\omega_{\text{kkey}}$  を選択する。  
 (2) SHS での 512 ビットの初期ハッシュ値

$$H_0 || H_1 || \cdots || H_4 = 67452301 || \text{EFCDAB89} || 98\text{BADCFE} || 10325476 || \text{C3D2E1F0} \quad (5.3)$$

をシフトした

$$t = \text{EFCDAB89} || 98\text{BADCFE} || 10325476 || \text{C3D2E1F0} || 67452301 \quad (5.4)$$

を選択する。

- (3)  $0 \leq j \leq m-1$  として以下の (a)-(d) を繰り返す。  
 (a)  $k = G(t, \omega_{\text{kkey}}) \bmod q$   
 (b)  $k_j^{-1} = k^{-1} \bmod q$  を計算する。  
 (c)  $r_j = (g^k \bmod p) \bmod q$   
 (d)  $\omega_{\text{kkey}} = (1 + \omega_{\text{kkey}} + k) \bmod 2^b$   
 (4)  $m$  個のメッセージを  $M_0, M_1, \dots, M_{m-1}$  として、 $0 \leq j \leq m-1$  に対して以下の (a)-(c) を繰り返す。  
 (a)  $h = \text{SHA-1}(M_j)$      $\text{SHA-1}(\cdot)$  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。

(III) FIPS186-2 の Appendix 3.3 の SHA-1 による一方向性関数  $G(t, c)$  の技術仕様  $G(t, c)$  は、下記 (III-II) の Secure Hash Standard (SHS) の技術仕様 (FIPS180-1, 1995, April 11) の Sec.7 の手順 (a)-(e) または、(III-III) の Sec.8 の手順 (a)-(d) 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  $c$  のパディング化を以下の手順で行う。

(III-I)  $\{H_j\}$  の初期化と  $c$  のパディング化

- (i) 160 ビットの  $t$  の 32 ビット分割  $t = t_0 || t_1 || \cdots || t_4$  に対し

$$H_j^{(0)} = t_j, (0 \leq j \leq 4) \quad (5.5)$$

とおく。

- (ii)  $M_1 = c || 0^{512-b}$

(III-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 16 words、512 ビット) を  $M_1, M_2, \dots, M_n$  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  $A, B, C, D, E$  と 2 番目のバッファ群  $H_0, H_1, \dots, H_4$  および 80 個の words 群  $W_0, W_1, \dots, W_{79}$  に対して、

SHS での 512 ビットの初期ハッシュ値

$$\left. \begin{aligned} H_0^{(0)} &= 67452301 \\ H_1^{(0)} &= \text{EFCDAB89} \\ H_2^{(0)} &= 98BADCFE \\ H_3^{(0)} &= 10325476 \\ H_4^{(0)} &= \text{C3D2E1F0} \end{aligned} \right\} \quad (5.6)$$

を用いる。

$M_i$  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

$$M_i = W_0 || W_1 || \cdots || W_{15} \quad (5.7)$$

(b) メッセージスケジュール関数

$$W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}), \quad 16 \leq t \leq 79. \quad (5.8)$$

を計算する。ただし、 $\oplus$  は排他的論理和を意味し、 $S^n(X)$  は word  $X$  の左  $n$  ( $0 \leq n \leq 32$ ) ビット巡回シフト関数  $S^n(X) = (X \ll n) \cup (X \gg 32 - n)$  を意味する。

(c) 5 個のバッファを

$$\left. \begin{aligned} A_0 &= H_0^{(0)}, & B_0 &= H_1^{(0)}, \\ C_0 &= H_2^{(0)}, & D_0 &= H_3^{(0)}, \\ E_0 &= H_4^{(0)} \end{aligned} \right\} \quad (5.9)$$

と初期値化する。ただし、 $H_k^{(0)}$  ( $0 \leq k \leq 4$ ) は初期ハッシュ値

$$\left. \begin{aligned} H_0^{(0)} &= 67452301, & H_1^{(0)} &= \text{EFCDAB89}, \\ H_2^{(0)} &= 98BADCFE, & H_3^{(0)} &= 10325476, \\ H_4^{(0)} &= \text{C3D2E1FD} \end{aligned} \right\} \quad (5.10)$$

である

(d)  $0 \leq t \leq 79$  に対し、以下の計算をする。

64 ビット以下のビット長  $\ell \leq 2^{64} - 1$  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

$$X = x || 1 || 0^m || \ell \quad (5.11)$$

を行う。ただし、 $m + \ell + 1 = 448 \bmod 512$  である。

$$TEMP_t = S^5(A_t) + f(B_t, C_t, D_t) + E_t + W_t + K_t \quad (5.12)$$

$$\left. \begin{aligned} E_{t+1} &= D_t \\ D_{t+1} &= C_t \\ C_{t+1} &= S^{30}(B_t) \\ B_{t+1} &= A_t \\ A_{t+1} &= TEMP_t \end{aligned} \right\} \quad (5.13)$$

ただし、 $f(B, C, D)$  は以下で定義される関数

$$f(B, C, D) = \begin{cases} (B \cap C) \cup (\bar{B} \cap D), & (0 \leq t \leq 19) \\ B \oplus C \oplus D, & (20 \leq t \leq 39) \\ (B \cap C) \cup (B \cap D) \cup (C \cap D), & (40 \leq t \leq 59) \\ B \oplus C \oplus D, & (60 \leq t \leq 79) \end{cases} \quad (5.14)$$

である。ただし、 $\bar{\cdot}$ ,  $\cap$ ,  $\cup$  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

$$K_t = \begin{cases} 5A827999, & (0 \leq t \leq 19) \\ 6ED9EBA1, & (20 \leq t \leq 39) \\ 8F1BBCDC, & (40 \leq t \leq 59) \\ CA62C1D6, & (60 \leq t \leq 79) \end{cases} \quad (5.15)$$

(e)

$$\left. \begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + A_{80} \\ H_1^{(i)} &= H_1^{(i-1)} + B_{80} \\ H_2^{(i)} &= H_2^{(i-1)} + C_{80} \\ H_3^{(i)} &= H_3^{(i-1)} + D_{80} \\ H_4^{(i)} &= H_4^{(i-1)} + E_{80} \end{aligned} \right\} \quad (5.16)$$

最終メッセージ  $M_n$  を処理後、160 ビットの

$$G(t, c) = H_0^{(n)} \| H_1^{(n)} \| \cdots \| H_4^{(n)} \quad (5.17)$$

をメッセージダイジェストとして出力する。

(III-III) SHS の技術仕様の Sec.8 の手順

Sec.7 の手続き中の 80 個の words 群  $W_0, W_1, \dots, W_{79}$  を以下のように節約できる。MASK = 0000000F とおく。 $M_i$  を処理するために、以下の手続き (a)-(d) の手順を実行する。

(a)  $M_i$  を 16 個の words に分割

$$M_i = W_0 \| W_1 \| \cdots \| W_{15} \quad (5.18)$$

(b) 5 個のバッファを

$$\left. \begin{aligned} A_0 &= H_0^{(0)}, & B_0 &= H_1^{(0)}, \\ C_0 &= H_2^{(0)}, & D_0 &= H_3^{(0)}, \\ E_0 &= H_4^{(0)} \end{aligned} \right\} \quad (5.19)$$

と初期値化する。

(c)  $0 \leq t \leq 79$  に対し、以下の計算をする。

$$s = t \cap \text{MASK} \quad (5.20)$$

$$W_s = S^1(W_{(s+13) \cap \text{MASK}} \oplus W_{(s+8) \cap \text{MASK}} \oplus W_{(s+2) \cap \text{MASK}} \oplus W_s) \text{ if } t \geq 16 \quad (5.21)$$

$$\text{TEMP}_t = S^5(A_t) + f(B_t, C_t, D_t) + E_t + W_s + K_t \quad (5.22)$$

$$\left. \begin{aligned} E_{t+1} &= D_t \\ D_{t+1} &= C_t \\ C_{t+1} &= S^{30}(B_t) \\ B_{t+1} &= A_t \\ A_{t+1} &= \text{TEMP}_t \end{aligned} \right\} \quad (5.23)$$

(d)

$$\left. \begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + A_{80} \\ H_1^{(i)} &= H_1^{(i-1)} + B_{80} \\ H_2^{(i)} &= H_2^{(i-1)} + C_{80} \\ H_3^{(i)} &= H_3^{(i-1)} + D_{80} \\ H_4^{(i)} &= H_4^{(i-1)} + E_{80} \end{aligned} \right\} \quad (5.24)$$



最終メッセージ  $M_n$  を処理後、160 ビットの

$$G(t, c) = H_0^{(n)} || H_1^{(n)} || \cdots || H_4^{(n)} \quad (5.25)$$

をメッセージダイジェストとして出力する\*<sup>1</sup>。

- (IV) FIPS186-2 の Appendix 3.4 の DES による一方向性関数  $G(t, c)$  の技術仕様  
 $a_1, a_2, b_1, b_2$  を 32 ビット文字列とし、 $b'_1$  を  $b_1$  の下位 24 ビット文字列とする。

$$\left. \begin{aligned} K &= b'_1 || b_2 \\ A &= a_1 || a_2 \end{aligned} \right\} \quad (5.26)$$

に対して記号  $DES_K(A)$  を

$$DES_K(A) = DES_{b_1, b_2}(a_1, a_2) \quad (5.27)$$

で定義すると、 $DES_K(A)$  は、通常の 64 ビットブロック  $A$  に対する 56 ビット鍵  $K$  を有する DES の暗号文を意味する。160 ビットの  $t, c$  に対する一方向性関数  $G(t, c)$  を以下の手順で計算する。

- (a)  $t, c$  を各々 32 ビットに分割

$$\left. \begin{aligned} t &= t_1 || t_2 || t_3 || t_4 \\ c &= c_1 || c_2 || c_3 || c_4 \end{aligned} \right\} \quad (5.28)$$

する。

- (b)  $1 \leq i \leq 5$  に対して

$$x_i = t_i \oplus c_i \quad (5.29)$$

を計算する。

- (c) (c)  $1 \leq i \leq 5$  に対して

$$\left. \begin{aligned} b_1 &= c_{((i+3) \bmod 5)+1} \\ b_2 &= c_{((i+2) \bmod 5)+1} \\ a_1 &= x_i \\ a_2 &= x_{(i \bmod 5)+1} \oplus x_{((i+3) \bmod 5)+1} \\ y_{i,1} || y_{i,2} &= DES_{b_1, b_2}(a_1, a_2) \end{aligned} \right\} \quad (5.30)$$

を計算する。ただし、 $y_1, y_2$  は 32 ビットである。

- (d)  $1 \leq i \leq 5$  に対して

$$z_i = y_{i,1} \oplus y_{((i+1) \bmod 5)+1,2} \oplus y_{((i+2) \bmod 5)+1,1} \quad (5.31)$$

を計算する。

- (e)

$$G(t, c) = z_1 || z_2 || z_3 || z_4 \quad (5.32)$$

をメッセージダイジェストとして出力する。

\*<sup>1</sup> Sec.7 で得られるメッセージダイジェストと Sec.8 のそれとは同一であるが、Sec.8 の方法では使用するメモリを節約できるが、その代わり計算時間が長くなる。

### 5.3.1.3 その他

#### 1. 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た。

#### 2. DSA flaw

ベル研の Daniel Bleichenbauer 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。

#### 3. Cryptographic Toolkit

Cryptographic Toolkit によると、乱数生成法は一般に、Random Number Generator (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 規格は存在せず。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ィ技術を選定する際の標準化、勧奨、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。

#### 4. 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。

### 5.3.1.4 安全性評価

■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を考える。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、Birthday 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 年以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 FIPS-186 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。

## 参考文献

- [1] CNN.com.SCI-TECH, Cryptologists sees digital signature flaw, fix: <http://www.cnn.com/2001/TECH/internet/02/06/DSA.flaw.idg/index.html>
- [2] Cryptographic Toolkit: <http://csrc.nist.gov/encryption/>
- [3] Secure Hashing: <http://csrc.nist.gov/encryption/tkhash.html>
- [4] Random Number Generation: <http://csrc.nist.gov/encryption/tnrng.html>
- [5] New hashing algorithms (SHA-256, SHA-384, and SHA-512): Descriptions of SHA-256, SHA-384, and SHA-512
- [6] FIPS Pub 186-2, Digital Signature Standard (DSS) (2000 January 27 更新) Appendix3: Random Number Generation for the DSA
- [7] NIST Special Publication 800-22 (Dec. 2000 年更新) A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications
- [8] FIPS140-1, Security Requirements for Cryptographic Modules, Cryptographic Module Validation (CMV) Program: <http://csrc.nist.gov/cryptval/cmvp.html>

## 5.4 スクリーニング評価対象暗号の評価

### 5.4.1 TAO TIME

#### 5.4.1.1 技術解説

応募者が提出した暗号技術仕様書によれば、TAO TIME 認知アルゴリズムは、ネットワーク上の任意の 2 地点に存在する Client (アクセス側コンピュータ) と Server (被アクセス側コンピュータ) がお互いを「認知」する認知アルゴリズムであるとされている。一方、参照プログラム仕様書によれば、TAO TIME 認知アルゴリズムは擬似乱数生成アルゴリズムであるとされている。前者の仕様書で「乱数」という用語を使って参照されている唯一の関数は  $a(n)$  であり、後者の仕様書においては、 $a(n)$  の関数が定義されている。このことから、応募書の軽微な変更によって、今回応募中の擬似乱数生成系に該当し得る可能性のあるものは、 $a(n)$  のみである。

Client は、この  $a(n)$  により生成される数値と、Server から以前に受け取ったデータを使って、漸化式により次に Server に送るべきデータを算出し、その算出結果を Server に送る。Server は Client から以前に受け取ったデータを使って、漸化式により次に Client に送るべきデータを算出し、その算出結果を Client に送る。このデータのやり取りを連続して行うことを通じて Client と Server がお互いを「認知」とされている。

暗号技術仕様書では、「認知」という用語は「認証」とは違うという意味のことが記述されている。しかし、これらがどのように違うのかは明確でない。つまり、この Client と Server の間でのデータのやり取りによって、我々が情報セキュリティ分野で普段使っている「認証」をも含んで実現されると主張されているのか、このような「認証」とは無関係の応募者独自の想定による何か「認知」という別物が実現されると主張されているのかどうかは明確ではない。

#### 5.4.1.2 スクリーニング評価結果

まず、Client と Server との間でやりとりされるデータは漸化式によって算出された値であり、ネットワーク上に露出される。例えば、クライアント識別子  $C(n)$  やサーバ識別子  $S(n)$ 、および、スクランブル値  $X(n)$  等はネットワーク上に露出される。その後、このネットワーク上に露出されたデータから、漸化式を逆に用いることにより、ネットワーク上に露出されなかったデータも  $a(n)$  を含め簡単に算出することができる。よって  $C(n)$ 、 $S(n)$ 、 $X(n)$ 、 $a(n)$  他漸化式に現れるデータは、秘密鍵の生成用途等で機密性をも必要とするような擬似乱数として用いることは安全上問題がある。つまり、これらのデータを出力するような本方法は、電子政府で使うため今回応募した擬似乱数生成系としては安全性が不十分である。

さらに、 $a(n)$  に限っていえば、 $a(n) - a(n-1)$  を 10 進表記したとき、下 5 桁は常に物理タイムとなり、これだけを見ても擬似乱数生成系としては安全性が不十分である。

また、Client と Server がお互いを「認知」する手段としての本方法について、もし、「認知」の意味を、我々が情報セキュリティ分野で普段使っている「認証」とは無関係の別物として使っているとしたら、CRYPTREC としては何らコメントを持たない。

しかし、電子政府用暗号システムにおいて、セキュリティが必要とされる端末認証手段としては、上記の理由、つまり、ネットワーク上に露出されなかった漸化式中のデータも後で簡単に分かってしまうことから、通常の「認証」で必要とされる秘密性を保持するデータの存在が見当たらないので、現仕様のままでは十分な安全性を有するとはいえないと考えられる。

## 第6章

# SSL プロトコルに関する暗号技術 の評価

### 6.1 総評

#### 6.1.1 調査の目的

SSL(Secure Socket Layer) はインターネットにおいて最も普及しているセキュリティプロトコルである。SSL は Web 上の暗号化や認証機能などを実現するものであり、インターネットを用いた電子商取引に広く用いられている。また、SSL とほぼ同じ仕様が TLS(Transport Layer Security) という名称でインターネット標準として検討されている。

本調査では、SSL/TLS プロトコルおよびそこで用いられる暗号の安全性を調査・評価する。本報告書の結果が、電子政府のセキュリティシステムの調達者・設計者・ユーザに対して SSL/TLS に用いられる暗号とプロトコルの安全性について正しい理解を与え、SSL/TLS が適切に利用されることを期待する。

SSL/TLS プロトコルの安全性調査としては SSL の仕様上および一般に普及している実装ソフトウェア上のセキュリティホールを調査する。SSL に含まれる暗号技術の安全性評価は可能な限り他の電子政府暗号候補に対するものと同様のレベルで行ない、SSL/TLS で用いられる暗号を他の暗号と比較することを図る。

本調査のまとめとして、電子政府において SSL を運用する場合の留意点について述べる。

#### 6.1.2 調査の対象と範囲

SSL とは、OSI 参照モデルのうちセッション層に位置するプロトコル (通信手順) であり、Web ブラウズやファイル転送といったアプリケーションによらない汎用的なセキュリティを実現することができる。SSL3.0[15] は米国 Netscape Communications 社によって規定されたプロトコルである。一方、TLS(Transport Layer Security) Ver1.0 [7] とは、インターネット技術の標準活動を行なっている IETF (Internet Engineering Task Force) が SSL3.0 を引き継いで RFC2246 として規定したものである。SSL3.0 と TLS1.0

の差異と IETF で行なわれている TLS の拡張作業についても調査する。

暗号技術検討会から以下の暗号の安全性を SSL の利用環境のもとで評価するように依頼があった。

**共通鍵暗号** RC2 (40, 128 ビット)、RC4 (40, 128 ビット)、DES (40, 56, 168 ビット)  
**公開鍵暗号** RSA 暗号

RC4 については現在評価中のため本報告の対象からはずした。

### 6.1.3 調査の方法

暗号評価に関しては内外の暗号研究者(組織)に評価を依頼し、暗号技術評価委員会では結果のとりまとめを行なった。各暗号の評価者数は、DES 1 名(組織)、RC2 2 名(組織)、RSA 5 名(組織)である。また評価期間は平成 13 年 10 月より 14 年 2 月までである。

### 6.1.4 調査結果

#### 6.1.4.1 SSL/TLS プロトコル脆弱性の調査

SSL/TLS プロトコルについて、暗号方式の安全性、プロトコルとしての安全性、実装に関する安全性、運用上の安全性に分類して調査したところ次のような結果が得られた。

■**暗号方式に関わる安全性** SSL に用いられる DSA 署名に用いられる乱数が一様にはならず偏りがあるという問題が指摘されている。また、RSA 公開鍵暗号方式の暗号化には PKCS#1 と呼ばれる実装仕様が適用されるが、PKCS#1V1.5 に対する適用的選択暗号文攻撃の存在が指摘されており、改良版である PKCS#1V2.0 の利用が推奨される。さらに PKCS#1V2.0 にも別のセキュリティホールが指摘されており、改善策が V2.1 において実現されている。

■**プロトコルに関する安全性** SSL には相互認証、サーバ認証のみ、匿名の 3 つの認証モードがあるが、このうち匿名認証モードにおいては 2 者間の通信の間に不正者が介在する man-in-the-middle 攻撃が存在し、情報の盗聴・改ざんの攻撃を受ける可能性があるため利用することは推奨されない。その他、攻撃者が SSL3.0 に対応しているサーバ、クライアントに対して SSL2.0 やそれ以下の Version で通信を行うように強制する攻撃法である version rollback 攻撃などがある。

■**実装に関わる安全性** SSL/TLS の公開鍵証明書を使った認証に関して、証明書検出機構が実装されていないケース、不正な証明書に対して警告を出さないケース、認証動作を迂回する攻撃が可能となるケースなどが報告されている。また、セッション鍵に用いる疑似乱数生成器の内部状態が暴露する攻撃も報告されている。実装に関するセキュリティホールについては、バグを改修した最新版や修正プログラムを用いることで攻撃を回避できる。

■**運用に関わる安全性** SSL/TLS にはサーバ、クライアント側で運用時に設定するパラメータがいくつか存在し、鍵や証明書を格納するファイル、セッション鍵のライフタイム、乱数生成方法、使用する暗号アルゴリズム、警告メッセージの表示可否などが設定可能である。このためこれらのパラメータの意味を十分理解し適切に設定する必要がある。不用意な設定がセキュリティホールになる可能性がある。

■**SSL/TLS の比較調査** 総じて TLS は SSL に比較して、鍵、初期値、MAC 生成に関して安全性の根拠を明確にし、署名の構造を若干修正したという点でセキュリティ上の差異があるが、SSL についても実用上問題のないレベルであると考えられる。

#### 6.1.4.2 SSL/TLS に用いられる暗号の評価

SSL/TLS で用いられる暗号の安全性評価について次のような調査結果が得られた。

##### ■DES(6.3.2.4 節参照) (暗号単独の安全性評価)

鍵長 40 ビットの DES は鍵総当りにより現実的な時間で解読可能である。鍵長 56 ビットの DES も現実的に解読可能な領域に達しつつある。鍵長が 168 ビットの 3-key Triple DES であれば当面の間の使用は問題ないといえる。

(SSL/TLS における安全性評価)

いずれの鍵長においても DES はデータ秘匿の目的に用いられる。ブロック暗号モードとしては CBC モードが用いられる。 $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化すると平文 1 ビットの情報がもれる可能性がある。

##### ■RC2(6.3.3.3 節参照) (暗号単独の安全性評価)

鍵長 40 ビットの RC2 は鍵総当りにより現実的な時間で解読可能である。鍵長 128 ビットの RC2 も現実的に解読可能な領域に達しつつある。最新の暗号解読理論を適用したときに鍵総当りより効率的な解読が知られている。学術的意味において解読可能である。

(SSL/TLS における安全性評価)

鍵長は SSL2.0 においては 40 ビットおよび 128 ビットが選択可能であり、SSL3.0 においては 40 ビットのみが選択可能である。40 ビットの鍵は現実的な計算機環境において数時間もあれば全数探索可能である。なお、DES と同じく、 $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化すると平文 1 ビットの情報がもれる可能性がある。

##### ■RSA(2.3.1 節、2.4.4 節、6.3.1.2 節参照) (暗号単独の安全性評価)

512 ビットの法は現実的に素因数分解可能であり安全ではない。2001 年時点では、1024 ビット以上の法を用いれば安全であると考えられる。

(SSL/TLS における安全性評価)

SSL および TLS における RSA 暗号を利用した基本的な鍵共有法および署名法の安全性について調べた。その結果、単純な技術の組み合わせの上にもっとも基本的なスキームを採用しており、暗号プロトコルとしてはセキュリティホールが潜む余地はほとんどないと

考えられる。

### 6.1.5 SSL/TLS の運用と利用についての注意点

SSL/TLS を安全に利用するためには、適切な運用が不可欠である。以下に運用と利用にあたっての注意事項を示す。

#### ■SSL/TLS プロトコルに関して (6.2 節参照)

- SSL3.0 を利用するにあたっては、セキュリティホールを含む SSL2.0 の利用を不可とするなど既知のセキュリティホールを十分認識した上での設定をすべきである。
- 市販の SSL ソフトウェアを利用する場合、セキュリティホールに対してパッチのあてられた最新版を利用すべきである。
- 市販のブラウザである Internet Explorer および Netscape Navigator においては CRL (公開鍵証明書無効化リスト) の管理は行なわれていない。従って CRL を不正に消去した上で不正な証明書を用いて認証を欺くという攻撃がありうる。このようなことがないように証明書を格納するファイルは厳密なアクセス管理のもとに管理すべきである。
- 情報の盗聴・改ざんの攻撃を受ける可能性があるため、匿名認証モードの利用を推奨しない。
- SSL3.0 では利用する暗号方式については変更できない。一方、TLS1.0 においては新しい暗号技術を追加することが可能となっている。そのため既存の暗号技術に問題があった場合にも対応が可能である。
- なお、TLS は機能追加を目的として拡張作業が行なわれているが、これらの拡張に伴って新たなセキュリティホールが発生する可能性もあるため、今後とも TLS の動向に注目し、その安全性について継続的な調査・検討が必要である。

#### ■SSL/TLS で利用される暗号に関して (2.3.1 節、2.4.4 節、6.3.1.2 節、6.3.2.4 節、6.3.3.3 節参照)

- RC2 であろうと DES であろうと鍵長 40 ビットの暗号は鍵の全数探索法により現実的な時間で解読可能であるため、安全性が必要なシステムにおいては用いられるべきではない。
- 鍵長 56 ビットの DES はもはや現実的に解読可能な領域に達しており、このことを十分配慮した上で利用すべきである。鍵長 168 ビットの Triple DES は当面の間の使用は安全性上特に問題ないが Triple DES に代わる暗号があればそれに置きかえるほうが望ましい。
- 64 ビットブロック暗号である RC2、TripleDES は  $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化すると平文 1 ビットの情報がもれる可能性があるためセッション鍵の更新に注意すべきである。
- 鍵長 128 ビットの RC2 に対して、鍵の全数探索法よりも効率のよい解読方法が存在する。よって、新規に構築する電子政府システムにおいて鍵長 128 ビットの RC2 を採用することは勧めない。
- 512 ビットの法は現実的に素因数分解可能であり安全ではない。2001 年時点では、1024 ビット以上の法を用いれば安全であると考えられる。



## 6.2 SSL/TLS プロトコルの実装と運用方法の評価

SSL(Secure Socket Layer) は、インターネットにおいて最も利用されているセキュリティプロトコルである。Web 上の暗号化、認証機能として主に利用されており、SSL を用いオンラインクレジット決済など電子商取引におけるキーテクノロジーとなっている。しかし、単に SSL を使っているから「安全が保証されている」といった認識があるのも事実であり、具体的に SSL がどの程度の安全性を有するのかを客観的に調査・評価した試みはあまり見受けられず、利用者としては、安全性を熟知した上でその利用を推進することが望まれている。

本報告は、SSL と同等の仕様である TLS(Transport Layer Security) を含めて、SSL/TLS について様々な角度から安全性の調査を行ったものである。具体的には、(1) 暗号方式の安全性、(2) プロトコル(メカニズム)としての安全性、(3) 実装に関する安全性、(4) 運用上の安全性といった観点から、これまで報告されてきたセキュリティホールを調査するとともに、幾らかの考察も行った。ここで、(1) 暗号方式、(2) プロトコル上の安全性とは、方式の仕様上の問題でセキュリティホールとなりえるもので、その中でも純粋に暗号方式に関連するものを(1)に、エンドエンドの両エンティティが関与するプロトコル手順上の問題について(2)に分類した。また、(3)は、機能実装の際、生じる問題で、主にバグと考えられる。(4)については、SSL/TLS を動作・運用させる際に指定可能な構成情報やオペレーションなどに関する問題に類する。

本報告では、さらに、SSL/TLS の違いを明確化するとともに、IETF で検討が進められている TLS の拡張作業について、セキュリティに関わる拡張に限定して調査した。以下にその調査結果の概要を示す。なお、SSL の最新バージョンは 3.0、TLS の最新バージョンは 1.0 である。とくにことわりのない限り、SSL、TLS は、SSL3.0[15]、TLS1.0[7] をそれぞれ指すものとする。

### 6.2.1 暗号方式に関わる安全性について

暗号方式は、セキュリティプロトコルの安全性の基盤となるものである。SSL/TLS においては、署名方式、公開鍵暗号方式、共通鍵暗号方式、鍵つきハッシュ処理を利用しており、これらの方式のなかでも、DSA 署名方式における乱数生成の偏りの問題(Bleichenbacher の攻撃)、RSA 公開鍵暗号方式の暗号化に対する適応型選択暗号文攻撃の可能性(Million Message 攻撃や、Manger らの攻撃)が、指摘されている。しかしながら、これらの問題については、既に改善方式が提案されており、最新のソフトウェアを用いる限り問題はないといえる。

### 6.2.2 プロトコルメカニズムに関わる安全性について

プロトコルに対する攻撃としては、SSL/TLS に限らず、いくつかの代表的な攻撃が存在する。具体的には、二者間の通信の間に不正者がゲートウェイのように介在する Man-in-the-middle 攻撃、通信を傍受し、その傍受したデータおよびその一部を再送することにより、正当な通信になりすます Replay 攻撃、セキュリティ上に問題があった古いバージョンに意図的に fall back させる version rollback 攻撃、流れるデータを解析することによって、IP アドレスや、URL などを推測する Traffic Analysis 攻撃などがある。SSL/TLS においては、上記の種々の攻撃について対策がなされている。いくつかの攻撃

(Key-exchange algorithm rollback 攻撃, Dropping the change cipher spec message 攻撃)について、セキュリティメカニズム上、留意すべきものがあるが、実装手法や運用により回避できる。従って、SSL/TLS については、既存の種々の攻撃に耐性を有するセキュリティプロトコルと考えられる。

### 6.2.3 実装に関わる安全性について

実装上の問題は、いわゆるバグであるためさまざまな問題が考えられる。ここで、SSL/TLS のバグとしての傾向は、SSL/TLS が公開鍵証明書を使った認証を行っており、そのメカニズムの複雑性や、実装者の理解の不十分さがバグの原因と思われる場合がいくつかある。また、セッション鍵が乱数をベースに生成されるため、その擬似乱数生成プログラムの不備をついた攻撃も報告されている。実装に関するセキュリティホールについては、バグを改修した最新版や修正プログラム(パッチ)を用いることで攻撃を回避できる。

### 6.2.4 運用に関わる安全性について

SSL/TLS についても運用時に設定するパラメータがサーバ、クライアント側にいくつか存在する。ここで、調査対象としてはサーバに `mod_ssl` を、クライアントに Web ブラウザ (Netscape Navigator 4.7x, Internet Explorer 5.5) を用いた。調査の結果、鍵や証明書を格納するファイル、セッション鍵のライフタイム、乱数生成手法、使用する暗号アルゴリズム、警告メッセージの表示可否などが設定可能であることが判明した。このため、これらのパラメータの意味を十分に理解し、適切に設定する必要がある。さもなければ、不用意な設定がセキュリティホールとなりえるからである。具体的には、鍵や証明書を格納するファイルは厳密なアクセス制御のもとに管理すべきであり、セッション鍵のライフタイムは、攻撃可能な暗号文のサンプルが収集できないよう短期間に、乱数生成は乱数性の高い方式を、使用する暗号アルゴリズムは十分な鍵長 (128 ビット以上) を持つ安全な暗号を、警告メッセージは可能な限り表示要に設定することが望ましい。これらのパラメータを適切に設定するかどうかは、運用者あるいは利用者の責任に帰着する。特に、SSL/TLS は Web ブラウザで使われるため、セキュリティの知識が乏しい一般利用者が対象となる点に留意する必要がある。これらのパラメータを適切に設定することにより、運用上のセキュリティホールは回避できる。

### 6.2.5 SSL/TLS の比較調査

SSL と TLS の主な相違点としては、共有する鍵や初期値の生成手法、およびデータ完全性の検証を行うための MAC の計算手法、サポートする暗号アルゴリズムが異なる点である。鍵、初期値、MAC の計算について、SSL では、独自の鍵つきハッシュ処理を用いているのに対し、TLS では、RFC2014 で規定される HMAC を用いている。SSL の計算手法に問題があるというよりも、TLS ではハッシュ関数と同等の安全性が保証される標準的な HMAC を用いる点で、安全性の根拠をより明確にしている。暗号アルゴリズム(方式)の差異としては、SSL では、Netscape 独自の Fortezza を用いているが、TLS ではサポートしていない。また、TLS では、アラートメッセージが詳細化されており、障害原因の特定を容易にしている。また、細かな点として、SSL で安全性について疑義の指摘があった署名の構造を TLS では、若干修正している点、SSL でオーバスペックな署名処理を TLS では標準的な署名方式に軽量化している部分もある。

総じて、TLS は SSL と比較して、鍵、初期値、MAC 生成に関して、安全性の根拠を

明確化した、署名の構造を若干修正した、といった点でセキュリティ上の差異はあるが、SSL についても、実用上問題ないレベルと考えられる。

### 6.2.6 TLS の拡張作業

SSL は、1997 年以降改定は行われていないが、TLS は IETF において拡張作業が続けられている。その内容としては、A) ワイヤレス・モバイル環境への対応、B) 新規暗号アルゴリズムの追加、C) 認証・鍵交換手法のバリエーション追加、D) プロトコルの拡張に大別される。これらは、現状の TLS の安全性を向上するのが主目的ではなく、さまざまな利用環境を想定した機能追加 (向上) が主な目的と考えられる。なお TLS については、2002 年 6 月に Draft Standard として IESG に仕様が提出される予定となっている。

### 6.2.7 総括

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

## 6.3 SSL/TLS で利用されている暗号技術の評価

### 6.3.1 RSA(1024,2048) を用いた鍵共有法および署名法の脆弱性に関する調査

本節では、SSL および TLS で用いられている RSA 暗号を利用した鍵共有法、署名法の安全性を評価した結果について述べる。本評価では、鍵共有法および署名法を抽象的に表現したものについての安全性評価を主に行なっている。したがって、その表現においてデータ表現法 (ヘッダのあるビットが何を表しているか等) は、プロトコル表現に含まれていない。安全性評価としては、プロトコルを用いることにより、暗号自体を解読せずに共有された秘密情報の漏洩や、署名の偽造等の不正ができないかどうか、についての議論を中心に行なっている。そしてその評価の結果、特に対象とする方式に問題は見られないということがわかった。

6.3.1.1 節では、現在広く用いられているプロトコルである SSL 3.0[15] および TLS 1.0[7] における、RSA 暗号を用いた鍵共有法、署名法について概説する。6.3.1.2 節ではこれら方式について、プロトコルでの使用における安全性について議論する。最後に 6.3.1.3 節で本評価の結論をまとめる。

### 6.3.1.1 RSA 暗号を用いた鍵共有法、署名法

本章では RSA を用いた鍵共有法、署名法について簡単に述べる。それぞれ SSL 3.0 と TLS 1.0 について説明を行なう。プロトコルにおいて、鍵共有法を用いて共有されるのは、

- 通信路暗号化に用いられる共通鍵暗号の秘密鍵 (SSL 3.0, TLS 1.0)
- データの完全性を保証するための MAC 生成、検証用鍵 (TLS 1.0)

を生成するための「プレマスタシークレット」と呼ばれる秘密情報である。

■**RSA を用いた鍵共有法** RSA 暗号を用いた鍵共有法において、SSL 3.0, TLS 1.0 の間でプロトコルに違いは存在しない。そのプロトコルは以下の通りである。

#### 1. サーバ→クライアント

サーバは RSA 暗号の暗号化用公開鍵とその鍵の証明書をクライアントへ送る。送る方法には以下のような2種類がある。

- (a) 署名検証用の鍵とその証明書を送り (Certificate メッセージ)、続けて RSA 暗号の暗号化用公開鍵とその鍵に関する情報への署名 (署名で使った鍵は証明されたもの) を送る (ServerKeyExchange メッセージ)。
- (b) RSA 暗号の暗号化用公開鍵とその証明書を送る (Certificate メッセージ)。

ここでその鍵に関する情報への署名とは、

- クライアントが生成した乱数
- サーバが生成した乱数
- RSA の公開鍵 (合成数  $n$  と乗数  $e$ )

を並べた情報に対し、MD5 関数をかけたもの、SHA-1 関数をかけたものにさらに署名したものである。この際 RSA 署名が用いられることもある。

#### 2. クライアント

クライアントはどちらの場合も、公開鍵の証明書 (やサーバの署名) を用いて、その鍵に関する情報への署名 RSA 暗号化用公開鍵の正当性を検証する。

#### 3. クライアント→サーバ

クライアントは鍵の正当性検証に成功した場合、プロトコルバージョン (2 バイト、SSL 3.0 の場合は 0x30、TLS 1.0 の場合は 0x31) と自分で生成した乱数 (46 バイト) を結合し、48 バイトのプレマスタシークレットを作成する。そしてそれを RSA 暗号を用いてサーバの公開鍵で暗号化し、サーバに送る (ClientKeyExchange メッセージ)。

#### 4. サーバ

サーバは送られてきた暗号文を復号し、プレマスタシークレットを得る。

以上でサーバ、クライアントはプレマスタシークレットを共有する。

■**RSA を用いた署名法** SSL 3.0, TLS 1.0 において、RSA 署名法は様々な場面で用いられるが、それらの中でプロトコルに違いは存在しない。以下それを列挙する。

#### 1. 署名検証用公開鍵証明書 (CA によるサーバのための)

- Certificate メッセージ (サーバ→クライアント)
- サーバの署名検証用公開鍵を証明するために、CA が RSA 署名をしたもの
  - クライアントは、CA(複数の場合あり)の公開鍵を用いて CA の署名を検証
  - 検証成功でサーバの署名検証用公開鍵が認証
- 2. 暗号化用公開鍵証明書 (CA によるサーバのための)
  - Certificate メッセージ (サーバ→クライアント)
  - サーバの暗号化用公開鍵を証明するために、CA が RSA 署名をしたもの
    - クライアントは CA (複数の場合あり)の公開鍵を用いて CA の署名を検証
    - 検証成功でサーバの暗号化用公開鍵が認証
- 3. 暗号化用公開鍵証明書 (サーバによる)
  - ServerKeyExchange メッセージ (サーバ→クライアント)
  - サーバの暗号化用公開鍵を証明するために、サーバが RSA 署名をしたもの
    - 署名データは前節 1 の下線部で表したデータ
    - クライアントは (CA(複数の場合あり)によって前もって証明された)サーバの署名検証用公開鍵を用いて署名を検証
    - 検証成功でサーバの暗号化用公開鍵が認証
- 4. 署名検証用公開鍵証明書 (CA によるクライアントのための)
  - Certificate メッセージ (クライアント→サーバ、送られるのは稀)
  - クライアントの署名検証用公開鍵を証明するために、CA が RSA 署名をしたもの
    - サーバは CA(複数の場合あり)の公開鍵を用いて CA の署名を検証
    - 検証成功でクライアントの署名検証用公開鍵が認証
- 5. クライアント認証用署名メッセージ (クライアントによる)
  - CertificateVerify メッセージ (クライアント→サーバ、送られるのは稀)
  - クライアントを認証するために、クライアントが RSA 署名をしたもの
    - 署名対象データは、それまでの通信メッセージに対し、MD5, SHA-1 関数をかけたものを結合したデータ
    - サーバは (CA (複数の場合あり)によって前もって証明された)クライアントの署名検証用公開鍵を用いて署名を検証
    - 検証成功でクライアントを認証

### 6.3.1.2 RSA 暗号を用いた鍵共有法、署名法の安全性

本章では SSL 3.0, TLS 1.0 で用いられている、RSA 暗号を用いた鍵共有法、署名法の安全性について議論する。

**■RSA 暗号を用いた鍵共有法の安全性** 前章で見たように、鍵共有法自体は RSA 暗号の暗号化通信方式そのものである。したがって、その安全性は RSA 暗号自体の安全性に依拠している。この部分については依頼内容とは異なるため、本報告では評価を行わない。

適切に用いた場合 RSA 暗号による暗号化通信は安全であると仮定したとき、プロトコルにおいて安全性を脅かすと考えられる部分は、

1. 署名検証用公開鍵証明書の偽造可能性
  - 不正者が証明書で証明された公開鍵の持ち主になりすます可能性
2. 暗号化用公開鍵証明書の偽造可能性
  - 不正者が証明書で証明された公開鍵の持ち主になりすます可能性

3. 証明書によって証明される公開鍵 (暗号化用、署名検証用) に対応する秘密鍵の推測可能性  
不正者が証明書で証明された公開鍵の持ち主になりすます可能性
4. 署名 (ServerKeyExchange, CertificateVerify メッセージ中) の偽造可能性  
不正者が署名生成用秘密鍵の持ち主になりすます可能性
5. プレマスタシークレットとして使用されるデータの偽造あるいは予測可能性  
不正者が以降行なわれる暗号化通信の内容を得る可能性

といった部分である。以下それぞれの場合について詳しく述べる。

1. 署名検証用公開鍵証明書の偽造可能性  
単純な CA による (Root CA までの) 署名の連鎖により、公開鍵証明書は成り立っている。連鎖されている署名全てが偽造不可能であれば、証明書の偽造は不可能であると言える。これは作成時に使用される署名方式と、CA が署名の際に用いた鍵の安全性により決まる。大多数の CA により使用されている署名方式は DSA(DSS), RSA である。今回の評価では、これらの署名方式は安全であるとの仮定を置いているため、鍵の露呈による偽造の可能性についてのみ考えれば良い。またこのとき、公開鍵や署名から秘密鍵は得られないと仮定していることに注意 (以降でも同様に仮定)。  
さて、このような仮定のもとで不正者が鍵の露呈による偽造を行なうには、CA の秘密鍵自体を得る必要がある。プロトコル (とそれが用いている PKI のシステム) において、署名生成に用いられる CA の秘密鍵は、システムが適切に運営されていれば、その CA 以外知ることができない。またその秘密鍵が含まれた情報 (それが何らかの暗号で暗号化された情報等) は通信路上に流れることはない。したがって CA の秘密鍵を得ることはできない。よって、署名検証用公開鍵証明書の偽造は不可能であると言える。
2. 暗号化用公開鍵証明書の偽造可能性  
この偽造可能性は、署名検証用公開鍵証明書の場合と同様である。したがって安全性についても同様である。
3. 証明書によって証明される公開鍵に対応する秘密鍵の推測可能性  
本評価では、サーバあるいはクライアントによって、RSA 暗号の鍵対は適切に生成されていると仮定している。またプロトコルにおいて、サーバあるいはクライアントの秘密鍵は、システムが適切に運営されていれば、その保持者以外知ることができない。またその秘密鍵が含まれた情報 (それが何らかの暗号で暗号化された情報等) は通信路上に流れることはない。さらに仮定より、公開鍵と署名、あるいは公開鍵と暗号文から秘密鍵を得ることはできない。よって、証明される公開鍵に対応する秘密鍵を推測することはできないと言える。
4. 署名 (ServerKeyExchange, CertificateVerify メッセージ中) の偽造可能性  
たとえ署名を偽造できたとしても、プレマスタシークレットが得られなければ、以降の通信で正しい返答ができなくなり、なりなりすましが露呈するようになっている。もちろん通信の攪乱を防ぐ意味でも、このような不正ができないことが望ましい。これまで述べてきたような仮定より、プレマスタシークレットと署名者の秘密鍵が得られなければ、署名の偽造はできない。他の項で行なったような議論と同様、やはり不正者は署名者の秘密鍵を得ることができないと言える。よって署名の偽造は不可能であると言える。
5. プレマスタシークレットとして使用されるデータの偽造あるいは予測可能性  
本評価では、用いられている暗号技術の安全性、プレマスタシークレットとして使用されるデータの乱数性を仮定している。プロトコルにおいて、プレマスタシークレット (あるいは鍵共有法として DH 法を用いる場合、それを構成するための情報) を含む情報は暗号化されたもののみである。したがって予測はできないと言える。また暗号化されたデータからそれらの情報を得ることは、やはり仮定よりできないと言え

る。よって、プレマスタシークレットとして使用されるデータの偽造、あるいは予測はできないと言える。

**■RSA 暗号を用いた署名法の安全性** 前章で見たように、署名法自体は RSA 暗号の署名方式そのものである。したがって、その安全性は RSA 暗号自体の安全性に依拠している。この部分については依頼内容とは異なるため、本報告では評価を行わない。

本評価においては、適切に用いていれば RSA 暗号による署名法は安全であると仮定している。6.3.1.1 で述べたように、RSA 署名法が用いられる可能性があるのは以下の場面である。

1. 署名検証用公開鍵証明書 (CA によるサーバのための)
2. 暗号化用公開鍵証明書 (CA によるサーバのための)
3. 暗号化用公開鍵証明書 (サーバによる)
4. 署名検証用公開鍵証明書 (CA によるクライアントのための)
5. クライアント認証用署名メッセージ (クライアントによる)

考えるべき不正は署名の偽造である。1-5 について、前節の議論 (署名方式や秘密鍵の安全性) を用いることにより、署名の偽造ができないことが示せる。

### 6.3.1.3 結論

SSL 3.0, TLS 1.0 での RSA 暗号を利用した鍵共有法、署名法は、単純な技術の組合せの上にもっとも基本的なスキームを採用しており、セキュリティホールが潜む余地はほとんどないと考えられる。ただし、実装時に用いられるデータの形式や、乱数の発生方法、サーバのエラーメッセージの返し方等、セキュリティホールが潜む可能性のある部分は数多く残っている。特にサーバ以外のエンティティ (CA 等) が関係しているため、その選定 (利用している技術等に基づき) にあたっては十分検討をすべきであろう。

## 6.3.2 DES (40bit/56bit-key DES, 168bit-key Triple DES)

### 6.3.2.1 技術概要

DES (Data Encryption Standard) 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共通鍵ブロック暗号である [14]。また、Triple DES[19] は、1979 年に IBM の Tuchman により提案された DES 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 [1, 14]。

### 6.3.2.2 技術仕様

Triple DES は、Feistel 型暗号である DES を 3 回繰り返す構造をとっているため、DES と同じく 64 ビット入出力サイズの共通鍵ブロック暗号に分類される。平文を  $P$ 、暗号文

を  $C$  鍵を  $K$ 、鍵  $K$  による暗号化及び復号処理を  $E_K$  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**  $C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$

**復号**  $P = D_{K_1}(E_{K_2}(D_{K_3}(C)))$

この時、鍵  $K_1$ 、 $K_2$ 、 $K_3$  の取り方で次の三種類のオプションがとられる [1]。

- (1)  $K_1$ 、 $K_2$ 、 $K_3$  が独立
- (2)  $K_1$  と  $K_2$  が独立で、 $K_1 = K_3$
- (3)  $K_1 = K_2 = K_3$

特に (3) は、3つの鍵を全て同じとすることで、通常の『Single DES』との間で互換性がとられている。一般に (1) は、『3key Triple DES』、(2) は、『2key Triple DES』と呼ばれる。DES の鍵サイズが 56 ビットであることから、(1)～(3) の鍵サイズは各々、168 ビット、112 ビット、56 ビットである。

Triple DES の利用モードは ANSI X9.52 の中で、ISO 8372 で規定される 64 ビットブロック暗号の利用モード (ECB、CBC、CFB、OFB) をベースに拡張した利用モード (TECB、TCBC、TCFB、TOFB) と、その他 (TCBC-I、TCFB-P、TOFB-I) の計 7 つの利用モードが規定されている [1]。

### 6.3.2.3 その他

発表当初から DES の 56 ビットという鍵長は短かく、鍵の総当た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 [21]。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 段でなく 3 段にしているのは、中間一致攻撃 (meet-in-the-middle attack) [4] を避けるためである。実際に 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る [22]。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 ユーザーの間でも Triple DES に移行する動きは更に拡大することが予想される。

### 6.3.2.4 評価結果

■**安全性** Triple DES ((1)3-key Triple DES, (2)2-key Triple DES, (3)Single DES) についてこれまでに報告されてきた主な安全性の評価結果を、表 6.1 に示す。Single 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、Single 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-II) で約 22 時間で解読に成功したという報告もあり [22]、もはや現実的な意味で解読可能な領域に達していると言える。2-key Triple DES および 3-key Triple DES は代表的な Short Cut Method である差分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 2-key Triple DES 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、3-key Triple DES 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、3-key Triple DES であれば当面の間の使用は問題ないと言える。

表 6.1: 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 [Lucks による攻撃]		$2^{57}$ (選択平文数 $2^{56}$ ) [—]	$2^{112}$ (選択平文数 $2^{56}$ ) [ $2^{108.2}$ ]
Oorshot-Wiener 既知平文攻撃	—	$2^{120 - \log_2 N}$ 既知平文数 $N$	—
● Short Cut Method			
差分解読法	$2^{37}$ (選択平文数 $2^{47}$ )	最大差分特性確率 $2^{-162.3}$ 以下 (2)	(同左)
線形解読法	$2^{42}$ (選択平文数 $2^{43}$ )	最大線形特性確率 $2^{-134.7}$ 以下 (3)	(同左)
関連鍵攻撃 (4)	—	—	$2^{56} \sim 2^{72}$ (選択平文 1) (選択鍵ペア 1)

- (1) 解読に必要な Triple DES(または DES) の暗号化または復号処理の回数  
(2) TripleDES 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) TripleDES 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 Triple DES, 3-key Triple DES) 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。Single 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 [22]、もはや十分な安全性があるとは言えないようになった。一方、Triple DES は組合せ暗号であるため、ある条件の下では、鍵長が拡大するほどには実質的な安全性は向上しないことが示されている。代表的な例として、Merkle と Hellman が提案した選択平文攻撃では、2-key Triple DES の場合は、 $2^{57}$  (全数探索法では  $2^{112}$ )、3-key Triple DES 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 [13]。ただし、本解読法においては、解読成功確率を 50 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  $4.03 \times 10^{10}$  G ビットと膨大になるほか、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 [10]。なお、Lucks は、Merkle と Hellman による選択平文攻撃の処理回数を削減する解読方法を提案し

ており、3-key Triple DES に対して、 $2^{108}$  程度の計算量で解読できると報告している [11]。ただし、Lucks による解読法もまた、必要となる記憶媒体等から現実の脅威となる可能性は現時点では低いと考えられている。また、2-key Triple DES に関しては、Oorschot と Wiener が Merkle と Hellman による選択平文攻撃をもとに拡張した既知平文攻撃を提案し、既知平文数  $N$  に対して  $120 - N$  ビットの記憶媒体を用意すれば  $2^{120 - \log_2 N}$  という計算量で解読できると指摘している [20]。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 [11]。

(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 [3]。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、3-key Triple DES が Merkle と Hellman の選択平文攻撃よりも少ない計算量によって解読可能であるとの研究成果が報告されている [8]。具体的には、1 組の選択平文・暗号文ペアとある特定の関係を有する 1 組の鍵ペアを利用することによって、 $2^{56} \sim 2^{72}$  回程度の計算量で解読できることが示されている。しかしこの攻撃法は、2-key Triple DES には適用できないほか、攻撃可能な環境は極めて限定されているため、実際の脅威となるとの見方は少ないと言える。

■ソフトウェア (SW) 実装評価 Triple DES の SW 実装評価として、表 6.2、6.3 に CRYPTREC2000 での評価結果を示す。これによれば PC 環境 (Pentium III) での Triple DES のデータランダム化部速度は最速値で 854 [cycles/block] を達成する。またその後、2002 年の SCIS2002 において、CRYPTREC2000 における評価とほぼ同様な PC 環境 (Pentium III) で、様々な高速化実装技術を駆使することで、データランダム化部速度において最速値 763 [cycles/block] を達成したとの研究報告が発表されている [2]。

表 6.2: データランダム化部速度測定結果 (単位 [cycles/block])

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		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	854 / 856	854 / 856
2 回目	854 / 857	854 / 856
3 回目	854 / 856	854 / 857

■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 [6]。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 0.35  $\mu$ ルール ASIC ライブラリ)

表 6.3: 鍵スケジュール部+データランダム化部速度測定結果 (単位 [cycles/block])

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		
	暗号化 (最速値/平均値)	復号 (最速値/平均値)
1 回目	1963 / 1967	1971 / 1975
2 回目	1967 / 1971	1971 / 1975
3 回目	1963 / 1967	1971 / 1975

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): 148147 ゲート (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]

**■SSL/TLS における DES の安全性** SSL/TLS における DES は、40bit-key (Single) DES、56bit-key (Single) DES、168bit-key Triple DES (3-key Triple DES) の 3 種類がデータの秘匿の目的で利用される。尚、40bit-key (Single) DES とは、(Single) DES において鍵長を通常の 56 ビット鍵から 40 ビット鍵に短縮化したものである。いずれもブロック暗号利用モードとしては CBC モードが用いられる。

まず、Single DES に関しては、6.3.2 節で述べたように全数探索法に対して、もはや十分な安全性があるとは言えなくなった。従って、SSL/TLS における DES として、40bit-key (Single) DES と 56bit-key (Single) DES の利用は安全性の面から避けるべきである。

次に、Triple DES に関しては、まず SSL/TLS の bulk encryption として Triple DES を選択する際に注意しなければならないのは、 $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化することである。SSL/TLS において Triple DES は CBC モードで利用され、かつブロック長は 64 ビットであるため、 $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化すれば、暗号文一致攻撃により暗号文から平文に関する 1 ビットの情報が漏れる可能性が高くなる。ただし、ブロックサイズ 64 ビットの  $2^{32}$  ブロック分は 32G バイトであるため、セッション鍵の更新を適宜行うことによりこの問題は避けられる。

また 6.3.2 節で述べたように 168bit-key(3-key) Triple DES は  $2^{56}$  組の選択平文暗号文対を用いて学術的な意味での解読が可能である。そのため、次に注意しなければならないのは  $2^{56}$  ブロック以上を同じセッション鍵を用いて暗号化することであるが、この攻撃に要する計算量は  $2^{108.2}$  程度と非常に大きく、また、ブロックサイズ 64 ビットの  $2^{56}$  ブロック分も 512P バイトと非常に大きいためこの攻撃に対する脅威は無視してよいであろう。

### 6.3.3 RC2(40,128)

#### 6.3.3.1 技術概要

RC2 は、RSA Data Security, Inc.\*<sup>1</sup>の Ron Rivest により 1989 年に設計されたメッセージ長 64 ビットの共通鍵ブロック暗号である。

#### 6.3.3.2 技術仕様

RC2 の仕様は 1997 年 Internet Draft [18] として公開されている。アルゴリズムは暗号化処理と復号処理からなり、各処理は鍵スケジュール部とデータ攪拌部からなる。

鍵スケジュール部は 128 バイト以下の任意長さ\*<sup>2</sup>の秘密鍵を入力とし、16 ビットの部分鍵を 64 個出力する。データ攪拌部は鍵スケジュール部から出力された拡大鍵と 64 ビットの平文を入力とし、64 ビットの暗号文を出力する。データ攪拌部は MIX と MASH の 2 種類の段関数から構成され、処理は MIX 5 段、MASH 1 段、MIX 6 段、MASH 1 段、MIX 5 段の順で進められる。64 個ある部分鍵は MIX の各段において 4 つずつ利用される。

#### 6.3.3.3 評価結果

■**安全性** RC2 は代表的な Short Cut Method である差分解読法に対して、(鍵長が 60 ビットより大きい場合) 鍵の全数探索法よりも効率のよい解読法が存在する (すなわち学術的な意味で解読可能である)[16]。[9] では MIX 15 段+ MASH 2 段における  $2^{-58}$  の差分特性パスが見つかっており、[16] によりその差分確率が  $2^{-56.7}$  になると見積もられている。攻撃に要する選択平文暗号文対は  $2^{60}$  組であり、計算量は RC2 の暗号化あるいは復号  $2^{60}$  回程度である。

線形解読法に関しては [16] において MIX 3 段で  $2^{-15}$  となる線形特性パスが見つかっており、その線形確率は  $2^{10.3}$  になると見積もられている。しかしながら、それらをフルラウンドの RC2 の解読に結びつけるには到っていない。また、RC2 では巡回シフト演算と算術加算演算が有効に働いているため、線形解読をフルラウンドの RC2 に適用することは容易でないと予想されている [16, 17]。

高階差分攻撃に対しては、[16] において平文の 64 ビット (16 ビット × 4 ブロック) の内 3 ブロックを定数とし、残りの 1 ブロックを変数とした場合の  $r$  段の出力の代数次数が調べられている。その結果、出力の各ビットの代数次数は MIX 3 段以降で最大次数である 16 次となることが示された。RC2 は MIX 16 段と MASH 2 段により構成されているため、高階差分攻撃を RC2 に適用することは難しいと考えられる。

以上、まとめると RC2 は高階差分攻撃や線形解読法などに対しては耐性を持っているが、差分解読法に対しては学術的な意味で解読可能であり、どのような環境で利用された

\*<sup>1</sup> 現在の RSA Security, Inc.。

\*<sup>2</sup> SSL ver.2 では 40 ビットおよび 128 ビットが選択可能であり、SSL ver.3 および TLS ver.1 では 40 ビットのみが選択可能である。

としても十分な強度を持つことが要求される電子政府向け暗号には薦められない。

■**SSL/TLS における RC2 の安全性** SSL/TLS における RC2 はデータの秘匿の目的で利用される。ブロック暗号利用モードとしては CBC モードが利用される。鍵長は SSL ver.2 では 40 ビットおよび 128 ビットが選択可能であり、SSL ver.3 および TLS ver.1 では 40 ビットのみが選択可能である。

SSL/TLS の bulk encryption として RC2 を選択する際に最も注意しなければならないのは、40 ビット鍵を選択してはならないということである。(これは、SSL ver.3 および TLS ver.1 において RC2 を選択すべきでないことを意味する。) 40 ビットの鍵は現実的な計算機環境において数時間もあれば全探索可能であるため、このオプションの選択は安全性の面からは絶対に薦められない。

次に注意しなければならないのは、 $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化することである。SSL/TLS において RC2 は CBC モードで利用され、かつブロック長は 64 ビットであるため、 $2^{32}$  ブロック以上を同じセッション鍵を用いて暗号化すれば、暗号文一致攻撃により暗号文から平文に関する 1 ビットの情報が漏れる可能性が高くなる。ただし、ブロックサイズ 64 ビットの  $2^{32}$  ブロック分は 32G バイトであるため、セッション鍵の更新を適宜行うことによりこの問題は避けられる。

次に注意しなければならないのは、 $2^{60}$  ブロック以上を同じセッション鍵を用いて暗号化することである。安全性節で述べたように、RC2 は差分解読法を用いて  $2^{60}$  組の選択平文暗号文対から拡大鍵全てを求めることができる。そのため、 $2^{60}$  ブロック以上を同じセッション鍵を用いて暗号化した場合、差分解読法による解読の可能性が出てくる。ただし、ブロックサイズ 64 ビットの  $2^{60}$  ブロック分は 8E バイトと非常に大きいため、セッション鍵の更新を適当なタイミングで行うことでこの問題は避けられる。

最後に、SSL ver.3 および TLS ver.1 における RC2 は、かつて米国の輸出規制対象外であった鍵長 40 ビットの RC2 との互換性を保つ目的で残されている可能性がある。安全性節でも述べた通り、鍵長 128 ビットの RC2 は鍵の全数探索法よりも効率のよい解読方法が存在するため、理想的な暗号であるとは必ずしも言えない。これから新規に構築する電子政府システムにおいては、旧バージョンの互換性に配慮する必要は無く、安全性の観点から、SSL ver.2 でのみ利用可能な鍵長 128 ビットの RC2 を採用することは薦めない。

### 6.3.4 RC4(40,128) および Arcfour(128)

現在、暗号技術評価委員会で RC4 を評価中。

#### 参考文献

- [1] American National Standards Institute, *Triple Data Encryption Algorithm Modes of Operation (X9.52-1998)*, 1998.
- [2] 青木 和麻呂, Pentium III 上の Triple DES 実装最適化記, 2002 年暗号と情報セキュリティシンポジウム, SCIS2002, 12C-2, 2002.
- [3] E. Biham and A. Shamir, Differential cryptanalysis of the full 16-round DES, In *Advances in Cryptology -Proceedings of CRYPTO92*, Vol. 740 of *LNCS*, pp. 487–496. Springer-Verlag, 1993.
- [4] W. Diffie and M. E. Hellman, Exhaustive cryptanalysis of the NBS data encryp-

- tion standard, *Computer*, Vol. 10, No. 6, pp. 74–84, June 1977.
- [5] S. Fluhrer, I. Mantin, and A. Shamir, Weaknesses in the key scheduling algorithm of RC4, In *Selected Areas in Cryptography 8th Annual International Workshop, SAC 2001*, Vol. 2259 of *Lecture Notes in Computer Science*, pp. 1–24. Springer-Verlag, 2001.
  - [6] T. Ichikawa, T. Kasuya, and M. Matsui, Hardware evaluation of the AES finalists, In *The Third AES Candidate Conference*, pp. 279–285. the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14 2000.
  - [7] IETF, *The TLS Protocol Version 1.0, RFC2246*, 1999. <http://www.ietf.org/rfc/rfc2246.txt>.
  - [8] J. Kelsey, B. Schneier, and D. Wagner, Key-schedule cryptanalysis of IDEA, GDES, GOST, SAFER, and triple DES, In *Advances in Cryptology CRYPTO96*, Vol. 1109 of *LNCS*, pp. 237–251. Springer-Verlag, 1996.
  - [9] L. R. Knudsen, V. Rijmen, R. L. Rivest, and M. J. B. Robshaw, On the design and security of rc2, In *Fast Software Encryption, 5th International Workshop (FSE '98)*, Vol. 1372 of *LNCS*, pp. 206–221. Springer-Verlag, 1998.
  - [10] K. Kusuda and T. Matsumoto, *A Strength Evaluation of the Data Encryption Standard*, No. 97-E-5 in IMES Discussion Paper. Institute for Monetary and Economic Studies, Bank of Japan, 1997.
  - [11] S. Lucks, Attacking triple DES, In *proceedings of Fast Software Encryption '98*, Vol. 1372 of *LNCS*, pp. 239–253, 1998.
  - [12] M. Matsui, Linear cryptanalysis method for DES cipher, In *Advances in Cryptology -Proceedings of EUROCRYPT'93*, Vol. 765 of *LNCS*, pp. 386–397. Springer-Verlag, 1994.
  - [13] R. C. Merkle and M. Hellman, On the security of multiple encryption, *Communications of the ACM*, Vol. 24, No. 7, pp. 465–467, 1981.
  - [14] National Institute of Standards and Technology, *Data Encryption Standard (Federal Information Processing Standards Publication 46-3)*, 1999.
  - [15] Netscape Communications, *SSL 3.0 SPECIFICATION*, 1996. <http://home.netscape.com/eng/ssl3/draft302.txt>.
  - [16] RC2 外部評価者 1, Analysis of RC2, 平成 13 年度暗号技術評価委員会 外部評価報告書, 2002.
  - [17] RC2 外部評価者 2, RC2 の安全性詳細評価報告, 平成 13 年度暗号技術評価委員会 外部評価報告書, 2002.
  - [18] R. L. Rivest, A description of the RC2<sup>TM</sup> encryption algorithm, RFC 2268, 1997.
  - [19] W. Tuchman, Hellman presents no shortcut solutions to DES, *IEEE Spectrum*, Vol. 16, No. 7, pp. 40–41, 1979.
  - [20] P. C. van Oorschot and M. J. Wiener, A known plaintext attack on two-key triple encryption, In *Advanced in Cryptology -Proceedings of EUROCRYPT'90*, Vol. 473 of *LNCS*, pp. 318–325. Springer-Verlag, 1990.
  - [21] 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.
  - [22] 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.

## 第7章

# 2002 年度評価予定暗号の知的財産 情報・ライセンス方針・問い合わせ 先一覧

本章の内容は、応募暗号については、応募者から修正願いがでて、受け付けられたものを除いて、すべて 2001 年 10 月時点で応募者から提出された応募書類からの抜粋である。従って、問い合わせ窓口担当者等に変更のある場合もある。

その他評価が必要とされる暗号技術については、暗号仕様などを入手するのに役立つと考えられる情報を掲載した。

### 7.1 監視状態の暗号

#### 7.1.1 公開鍵暗号技術

##### 7.1.1.1 DSA

仕様 ANSI X9.30 Part 1 (<http://www.x9.org/> から入手可能) で規定されたもの。

参照 URL <http://csrc.nist.gov/encryption/tkdigsigns.html>

##### 7.1.1.2 ECDSA (ANSI X9.62)

仕様 ANSI X9.62 (<http://www.x9.org/> から入手可能) で規定されたもの。

##### 7.1.1.3 ECDSA (Elliptic Curve Digital Signature Algorithm) in SEC1

公開ホームページ URL

和文 [http://www.labs.fujitsu.com/theme/crypto/public\\_key2001.html](http://www.labs.fujitsu.com/theme/crypto/public_key2001.html)

英文 [http://www.labs.fujitsu.com/theme/crypto/public\\_key2001\\_e.html](http://www.labs.fujitsu.com/theme/crypto/public_key2001_e.html)

**学会発表等** SECG(Standards for Efficient Cryptography Group)、“SECG standards”  
ウェブページで公開 <http://www.secg.org/>、2000年9月20日

**調達窓口**

**担当者氏名** 下江 達二

**所属・役職** ソフトウェア事業本部運用管理ソフトウェア事業部 第二開発部 担当  
部長

**企業・団体名** 富士通株式会社

**所在地** 〒222-0033 神奈川県横浜市港北区新横浜 3-9-18 (TECH ビル)

**TEL** 045-474-1925

**FAX** 045-474-1953

**e-mail** shimoe@jp.fujitsu.com

**関連する特許権** SECG member patent letters(下記アドレス)をご参照ください。  
[http://www.secg.org/collateral/certicom\\_secg\\_patent.pdf](http://www.secg.org/collateral/certicom_secg_patent.pdf)

**関連する著作権** License to copy the document is granted provided it is identified  
as “Standards for Efficient Cryptography (SEC)”, in all material mentioning or  
referencing it.

**関連する特許とその扱い** 特許の扱いについては、下記の SECG Patent Policy をご参照  
ください。 [http://www.secg.org/patent\\_policy.htm](http://www.secg.org/patent_policy.htm)

**電子政府で使用する際のライセンス方針** 富士通株式会社の所有する特許は、合理的な条  
件で、提供先を差別することなく、実施権を供与する。

#### 7.1.1.4 RSA-PKCS#1 v1.5

**仕様** PKCS#1 RSA Cryptography Standard (Ver.2.0)

**参照 URL** <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html>

#### 7.1.1.5 RSA Public-Key Cryptosystem with Probabilistic Signature Scheme (RSA-PSS)

**公開ホームページ URL**

**和文** <http://www.rsasecurity.com/rsalabs/submissions/index.html>

**英文** <http://www.rsasecurity.com/rsalabs/submissions/index.html>

**学会発表等** Phillip Rogaway, “PSS/PSS-R (an encoding method for RSA or RW  
signatures)” IEEE P1363 Working Group, 1998年8月

**調達窓口**

**担当者氏名** 荒井 英治

**所属・役職** デベロッパ営業本部 部長

**企業・団体名** RSA セキュリティ株式会社

**所在地** 〒100-0005 東京都千代田区丸の内 1-3-1 東京銀行協会ビルディング 13F

**TEL** 03-5222-5210

**FAX** 03-5222-5270

**e-mail** earai@rsasecurity.com

**関連する特許権**

**関連する著作権** 応募提出物にあるサンプルコードは RSA Security が著作権を所有して  
います。

**関連する特許とその扱い**

**電子政府で使用する際のライセンス方針** RSA-PSS アルゴリズムに関する特許ライセン  
スを有しません。



### 7.1.1.6 RSA Public-Key Cryptosystem with Optimal Asymmetric Encryption Padding (RSA-OAEP)

#### 公開ホームページ URL

和文 <http://www.rsasecurity.com/rsalabs/submissions/index.html>

英文 <http://www.rsasecurity.com/rsalabs/submissions/index.html>

学会発表等 M. Bellare and P. Rogaway, “Optimal asymmetric encryption – How to encrypt with RSA” Eurocrypt’94, 1994 年

#### 調達窓口

担当者氏名 荒井 英治

所属・役職 デベロッパ営業本部 部長

企業・団体名 RSA セキュリティ株式会社

所在地 〒100-0005 東京都千代田区丸の内 1-3-1 東京銀行協会ビルヂング 13F

TEL 03-5222-5210

FAX 03-5222-5270

e-mail earai@rsasecurity.com

#### 関連する特許権

関連する著作権 応募提出物にあるサンプルコードは RSA Security が著作権を所有しています。

#### 関連する特許とその扱い

電子政府で使用する際のライセンス方針 RSA-OAEP アルゴリズムに関する特許ライセンスを有しません。

### 7.1.1.7 DH

仕様 次の文献に記載されているもの

W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, vol. IT-22, pp. 644-654, Nov. 1976

あるいは、第 2.5.3 節を参照のこと。

### 7.1.1.8 ECDH (Elliptic Curve Diffie-Hellman Scheme) in SEC1

#### 公開ホームページ URL

和文 [http://www.labs.fujitsu.com/theme/crypto/public\\_key2001.html](http://www.labs.fujitsu.com/theme/crypto/public_key2001.html)

英文 [http://www.labs.fujitsu.com/theme/crypto/public\\_key2001\\_e.html](http://www.labs.fujitsu.com/theme/crypto/public_key2001_e.html)

学会発表等 SECG(Standards for Efficient Cryptography Group)、“SECG standards” ウェブページで公開 <http://www.secg.org/>、2000 年 9 月 20 日

#### 調達窓口

担当者氏名 下江 達二

所属・役職 ソフトウェア事業本部運用管理ソフトウェア事業部 第二開発部 担当部長

企業・団体名 富士通株式会社

所在地 〒222-0033 神奈川県横浜市港北区新横浜 3-9-18 (TECH ビル)

TEL 045-474-1925

FAX 045-474-1953

**e-mail** shimoe@jp.fujitsu.com

**関連する特許権** SECG member patent letters(下記アドレス) をご参照ください。  
[http://www.secg.org/collateral/certicom\\_secg\\_patent.pdf](http://www.secg.org/collateral/certicom_secg_patent.pdf)

**関連する著作権** License to copy the document is granted provided it is identified as “Standards for Efficient Cryptography (SEC)”, in all material mentioning or referencing it.

**関連する特許とその扱い** 特許の扱いについては、下記の SECG Patent Policy をご参照ください。[http://www.secg.org/patent\\_policy.htm](http://www.secg.org/patent_policy.htm)

**電子政府で使用する際のライセンス方針** 富士通株式会社の所有する特許は、合理的な条件で、提供先を差別することなく、実施権を供与する。

## 7.1.2 共通鍵暗号技術

### 7.1.2.1 CIPHERUNICORN-E

#### 公開ホームページ URL

**和文** <http://www.hnes.co.jp/products/security/index.html>

**英文** <http://www.hnes.co.jp/products/security/index-e.html>

**学会発表等** 日本電気株式会社、“登録番号 19、登録日 1998.7.6、アルゴリズム公開登録” ISO/IEC 9979 暗号アルゴリズム登録制度、1998年7月6日

日本電気株式会社、“統計的手法により安全性が評価された暗号” 1998年暗号と情報セキュリティシンポジウム、SCIS'98-4.2.B、1998年1月29日

#### 調達窓口

**担当者氏名** セキュリティ技術センター

**所属・役職** インターネットソフトウェア事業部

**企業・団体名** 日本電気株式会社

**所在地** 〒108-8557 港区芝浦 2-11-5

**TEL** 03-5476-1913

**FAX** 03-6576-1678

**e-mail** sec@isd.nec.co.jp

#### 関連する特許権

1. **特許出願番号 (公開番号)** 出願平 9-213274

**名称** 暗号装置及び暗号装置を実現するプログラムを記録したコンピューターが読みとり可能な記録媒体

**出願日** 平成9年(1997)8月7日

**関連する著作権** 著作物 CIPHERUNICORN-E のプログラム

商標 登録番号 第 4221077 号

**関連する特許とその扱い** 現時点までに発行されている特許公報からは、関連する他社先行特許は発見できませんでした。

**電子政府で使用する際のライセンス方針** 民間企業等による営利目的の使用の場合を除き、無償とする考えです。

### 7.1.2.2 Hierocrypt-L1

#### 公開ホームページ URL

**和文** <http://www.toshiba.co.jp/rdc/security/hierocrypt>

**英文** <http://www.toshiba.co.jp/rdc/security/hierocrypt>

**学会発表等** 大熊建司、“ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 に対する強

度／性能評価” 電子情報通信学会 信学技法 ISEC2000-71 pp.71-100、2000年9月29日

#### 調達窓口

**担当者氏名** 大熊 建司  
**所属・役職** 研究開発センター コンピュータ・ネットワーク ラボラトリー 主任研究員  
**企業・団体名** (株) 東芝  
**所在地** 〒212-8582 神奈川県川崎市幸区小向東芝町 1  
**TEL** 044-549-2156  
**FAX** 044-520-1841  
**e-mail** kenji.ohkuma@toshiba.co.jp

#### 関連する特許権

1. **特許出願番号 (公開番号)** 特願 2000-210484  
**名称** 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに演算装置」  
**出願日** 2001/03/06
2. **特許出願番号 (公開番号)** 特願 2000-211686  
**名称** 「暗号化装置、復号装置及び拡大鍵生成装置、拡大鍵生成方法並びに記録媒体」  
**出願日** 2000/07/12
3. **特許出願番号 (公開番号)** 特願 2000-212175  
**名称** 「パラメータ決定装置、パラメータ決定方法、暗号化装置、および復号装置」  
**出願日** 2000/07/13
4. **特許出願番号 (公開番号)** 特願 2001-68742  
**名称** 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに記憶媒体」  
**出願日** 2001/06/30

#### 関連する著作権

**関連する特許とその扱い** 非排他的かつ妥当な条件で実施を許諾します。

**電子政府で使用する際のライセンス方針** 非排他的かつ妥当な条件で実施を許諾します。

### 7.1.2.3 MISTY1

#### 公開ホームページ URL

**和文** <http://www.security.melco.co.jp/misty>

**英文** <http://www.security.melco.co.jp/misty>

**学会発表等** 松井 充、“ブロック暗号アルゴリズムMISTY” 電子情報通信学会 ISEC 研究会、1996年7月22日

#### 調達窓口

**担当者氏名** 小松田 敏二  
**所属・役職** インフォメーションシステム事業推進本部 情報セキュリティ推進センター 副センター長  
**企業・団体名** 三菱電機株式会社  
**所在地** 〒100-8301 東京都千代田区丸の内 2-2-3 (三菱電機ビル)  
**TEL** 03-3218-3221  
**FAX** 03-3218-3638  
**e-mail** Binji.Komatsuda@hq.melco.co.jp

#### 関連する特許権

1. **特許出願番号 (公開番号)** 特許第 3035358 号  
**名称** データ変換装置及びデータ変換方法

**出願日** 2000.2.18

上記特許は、PCT/JP96/01254 (PCT 出願 1996年7月31日) にも出願中。

**関連する著作権** 提出書類の著作権は三菱電機株式会社に帰属します。

**関連する特許とその扱い** 弊社の知る範囲では、関連する他社特許はありません。

**電子政府で使用する際のライセンス方針** 上記特許に関しては、応募暗号技術 (MISTY1) を使用するものに対して相互主義の下に、非排他的に、無償で実施許諾する方針です。

#### 7.1.2.4 Triple DES

**参照 URL** <http://csrc.nist.gov/encryption/tkencryption.html>

#### 7.1.2.5 Camellia

##### 公開ホームページ URL

**和文** <http://info.isl.ntt.co.jp/camellia/>

**英文** <http://info.isl.ntt.co.jp/camellia/>

**学会発表等** 神田雅透、「128ビットブロック暗号 Camellia」 ISEC 研究会、2000年5月25日

##### 調達窓口

**担当者氏名** 中尾 昌善

**所属・役職** 情報流通プラットフォーム研究所 情報セキュリティプロジェクト (セ制 G) グループリーダー

**企業・団体名** 日本電信電話株式会社

**所在地** 〒239-0847 横須賀市光の丘 1-1-609A

**TEL** 0468-59-3334

**FAX** 0468-59-3365

**e-mail** nakao@isl.ntt.co.jp

**担当者氏名** 豊嶋 淳

**所属・役職** NTT 事業部第一部長

**企業・団体名** 三菱電機株式会社

**所在地** 〒104-6212 中央区晴海 1-8-12 オフィスタワー Z13 階

**TEL** 03-6221-2634

**FAX** 03-6221-2770

**e-mail** toshima@npd.hon.melco.co.jp

##### 関連する特許権

1. **特許出願番号 (公開番号)** 2000-064614

**名称** データ変換装置及びデータ変換方法及びデータ変換方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体

**出願日** 2000年3月9日

上記特許は、PCT/JP01/01796 (PCT 出願, 2001年3月8日) および第 90105464 号 (台湾, 2001年3月8日) にも出願中。

**関連する著作権** 「(2) 暗号技術仕様書」「(3) 自己評価書」「(7) 応募暗号説明会発表資料」の著作権は日本電信電話株式会社および三菱電機株式会社に、「(5) 参照プログラムおよびその仕様書, テストベクトル生成プログラムおよびその仕様書」の著作権は三菱電機株式会社に帰属します。

**関連する特許とその扱い** 弊社の知る範囲では、関連する他社の特許はありません。

**電子政府で使用する際のライセンス方針** 上記特許に関しては、応募暗号技術 (Camellia)

を使用するものに対して、相互主義の下に、非排他的に、無償で実施許諾する方針です。

### 7.1.2.6 CIPHERUNICORN-A

#### 公開ホームページ URL

和文 <http://www.hnes.co.jp/products/security/index.html>

英文 <http://www.hnes.co.jp/products/security/index-e.html>

学会発表等 日本電気株式会社、“信学技報 Vol.100 No.76 pp23-46, ISEC2000-5 「A New 128-bit Block Cipher CIPHERUNICORN-A」” 電子情報通信学会 ISEC 研究会、2000年5月26日

#### 調達窓口

担当者氏名 セキュリティ技術センター

所属・役職 インターネットソフトウェア事業部

企業・団体名 日本電気株式会社

所在地 〒108-8557 港区芝浦 2-11-5

TEL 03-5476-1913

FAX 03-6576-1678

e-mail [sec@isd.nec.co.jp](mailto:sec@isd.nec.co.jp)

#### 関連する特許権

1. 特許出願番号(公開番号) 出願平 9-213274

名称 暗号装置及び暗号装置を実現するプログラムを記録したコンピューターが読みとり可能な記録媒体

出願日 平成9年(1997)8月7日

関連する著作権 著作物 CIPHERUNICORN-A のプログラム

商標 登録番号 第 4221077 号

関連する特許とその扱い 現時点までに発行されている特許公報からは、関連する他社先行特許は発見できませんでした。

電子政府で使用の際のライセンス方針 民間企業等による営利目的の使用の場合を除き、無償とする考えです。

### 7.1.2.7 Hierocrypt-3

#### 公開ホームページ URL

和文 <http://www.toshiba.co.jp/rdc/security/hierocrypt>

英文 <http://www.toshiba.co.jp/rdc/security/hierocrypt>

学会発表等 大熊建司、“ブロック暗号 Hierocrypt-3 および Hierocrypt-L1 に対する強度/性能評価” 電子情報通信学会 信学技法 ISEC2000-71 pp.71-100、2000年9月29日

#### 調達窓口

担当者氏名 大熊 建司

所属・役職 研究開発センター コンピュータ・ネットワーク ラボラトリー 主任研究員

企業・団体名 (株) 東芝

所在地 〒212-8582 神奈川県川崎市幸区小向東芝町 1

TEL 044-549-2156

FAX 044-520-1841

e-mail [kenji.ohkuma@toshiba.co.jp](mailto:kenji.ohkuma@toshiba.co.jp)

**関連する特許権**

1. **特許出願番号 (公開番号)** 特願 2000-210484  
**名称** 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに演算装置」  
**出願日** 2001/03/06
2. **特許出願番号 (公開番号)** 特願 2000-211686  
**名称** 「暗号化装置、復号装置及び拡大鍵生成装置、拡大鍵生成方法並びに記録媒体」  
**出願日** 2000/07/12
3. **特許出願番号 (公開番号)** 特願 2000-212175  
**名称** 「パラメータ決定装置、パラメータ決定方法、暗号化装置、および復号装置」  
**出願日** 2000/07/13
4. **特許出願番号 (公開番号)** 特願 2001-68742  
**名称** 「暗号化装置及び暗号化方法、復号装置及び復号方法並びに記憶媒体」  
**出願日** 2001/06/30

**関連する著作権**

**関連する特許とその扱い** 非排他的かつ妥当な条件で実施を許諾します。

**電子政府で使用する際のライセンス方針** 非排他的かつ妥当な条件で実施を許諾します。

**7.1.2.8 RC6 Block Cipher****公開ホームページ URL**

**和文** <http://www.rsasecurity.com/rsalabs/submissions/index.html>

**英文** <http://www.rsasecurity.com/rsalabs/submissions/index.html>

**学会発表等** Ron Rivest, "The RC6 Block Cipher, algorithm specification" The First AES Candidate Conference (AES1), 1998年8月

**調達窓口**

**担当者氏名** 荒井 英治

**所属・役職** デベロッパ営業本部 部長

**企業・団体名** RSA セキュリティ株式会社

**所在地** 〒100-0005 東京都千代田区丸の内 1-3-1 東京銀行協会ビルヂング 13F

**TEL** 03-5222-5210

**FAX** 03-5222-5270

**e-mail** earai@rsasecurity.com

**関連する特許権**

1. **特許出願番号 (公開番号)** Patent number: 5,724,428  
**名称** Block Encryption Algorithm with Data-Dependent Rotations  
**出願日** November 1, 1995
2. **特許出願番号 (公開番号)** Patent number: 5,835,600  
**名称** Block Encryption Algorithm with Data-Dependent Rotations  
**出願日** April 21, 1997
3. **特許出願番号 (公開番号)** Serial number: 09/094,649  
**名称** Enhanced Block Encryption Algorithm with Data-Dependent Rotations  
**出願日** June 15, 1998
4. **特許出願番号 (公開番号)** Serial number: PCT/US99/13358  
**名称** Enhanced Block Encryption Algorithm with Data-Dependent Rotations  
**出願日** June 15, 1999

## 5. 特許出願番号 (公開番号) 特願 2000-555387

名称 整数乗算、データ依存および各回の固定ローテーション回数を有するブロック暗号

出願日

関連する著作権 応募提出物にあるサンプルコードは RSA Security が著作権を所有しています。

関連する特許とその扱い

電子政府で使用する際のライセンス方針 電子政府のためのアプリケーションに利用される場合には、RC6 アルゴリズムの使用に関しての特許使用に対するライセンス料は請求いたしません。

## 7.1.2.9 SC2000

公開ホームページ URL

和文 <http://www.labs.fujitsu.com/theme/crypto/sc2000.html>

英文 <http://www.labs.fujitsu.com/theme/crypto/sc2000.html>

学会発表等 下山 武司、“共通鍵ブロック暗号 S C 2 0 0 0 (ISEC2000 - 72)” 電子情報通信学会情報セキュリティ研究会、2000 年 9 月 29 日

調達窓口

担当者氏名 下江 達二

所属・役職 ソフトウェア事業本部運用管理ソフトウェア事業部 第二開発部 担当部長

企業・団体名 富士通株式会社

所在地 〒222-0033 神奈川県横浜市港北区新横浜 3-9-18 (TECH ビル)

TEL 045-474-1925

FAX 045-474-1953

e-mail [shimoe@jp.fujitsu.com](mailto:shimoe@jp.fujitsu.com)

関連する特許権

## 1. 特許出願番号 (公開番号) 2001-018016

名称 暗号設計装置、暗号設計プログラム及び記録媒体

出願日 2000.1.26

## 2. 特許出願番号 (公開番号) 2000-212813

名称 F 関数内部に SPN 構造を用いた演算装置及び演算方法

出願日 2000.7.13

## 3. 特許出願番号 (公開番号) 2000-212814

名称 Feistel 構造と SPN 構造とを組み合わせた演算装置及び演算

出願日 2000.7.13

## 4. 特許出願番号 (公開番号) 2000-212482

名称 拡大鍵生成装置及び記録媒体

出願日 2000.7.13

関連する著作権 富士通株式会社

関連する特許とその扱い なし

電子政府で使用する際のライセンス方針 上記特許、及び著作権は、合理的な条件で、提供先を差別することなく、実施権を供与する

## 7.1.2.10 SEED

参照 URL <http://www.kisa.or.kr/seed/index.html>

### 7.1.2.11 MULTI-S01

#### 公開ホームページ URL

和文 <http://www.sdl.hitachi.co.jp/crypto/s01/index-j.html>

英文 <http://www.sdl.hitachi.co.jp/crypto/s01/index.html>

学会発表等 古屋 聡一、“MULTI-S01 のパディングと安全性についての考察” 電子情報通信学会 ISEC 研究会、2000年9月29日

#### 調達窓口

担当者氏名 原野 紳一郎

所属・役職 (株)日立製作所ソフトウェア開発本部 推進部長

企業・団体名 (株)日立製作所ソフトウェア開発本部

所在地 〒244-8555 神奈川県横浜市戸塚区戸塚町 5030 番地

TEL (045)866-8140

FAX (045)865-9036

e-mail [harano@itg.hitachi.co.jp](mailto:harano@itg.hitachi.co.jp)

#### 関連する特許権

1. 特許出願番号(公開番号) 特願 2000-108334 号(特開 2001-007800)  
名称 「暗号化装置および方法」  
出願日 2000年4月10日
2. 特許出願番号(公開番号) 特願 2000-070994 号  
名称 「共通鍵暗号方法及び装置」  
出願日 2000年3月9日
3. 特許出願番号(公開番号) 特願 2000-210690 号  
名称 「共通鍵暗号方法及び装置」  
出願日 2000年7月6日

関連する著作権 本暗号技術 MULTI-S01 を応募するにあたり提出される文書、ソースコードなどの著作権はすべて日立製作所に帰属します。

関連する特許とその扱い 弊社は、MULTI-S01 の技術に関し、弊社による上記の出願特許および登録特許が関連すると考えられます。弊社と致しましては、MULTI-S01 が本提案に採用された場合には、上記特許を非差別的、かつ適正な対価条件でライセンス致します。ただし、相手方が相手方の MULTI-S01 関連特許を非差別的、かつ適正な対価条件でライセンスしない場合はこの限りではございません。

電子政府で使用する際のライセンス方針 上記の内容と同じです。

### 7.1.2.12 RIPEMD-160

参照 URL <http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html>

### 7.1.2.13 SHA-1

参照 URL <http://csrc.nist.gov/encryption/tkhash.html>

### 7.1.2.14 draft SHA-256/384/512

参照 URL <http://csrc.nist.gov/encryption/tkhash.html>



### 7.1.2.15 PRNG based on SHA-1

参照 URL <http://csrc.nist.gov/encryption/tkrng.html>

## 7.2 2002 年度詳細評価対象暗号候補

### 7.2.1 公開鍵暗号技術

#### 7.2.1.1 ESIGN(電子署名法に係る指針)

##### 公開ホームページ URL

和文 <http://info.isl.ntt.co.jp/>

英文 <http://info.isl.ntt.co.jp/>

学会発表等 岡本 龍明、“安全性の証明のついたデジタル署名 TSH-ESIGN および楢門 Okamoto-Schnorr” NTT R&D、1999 年 10 月

##### 調達窓口

担当者氏名 中尾 昌善

所属・役職 NTT 情報流通プラットフォーム研究所主幹研究員

企業・団体名 日本電信電話株式会社 (NTT)

所在地 〒239-0847 神奈川県横須賀市光の丘 1-1-609A

TEL 0468-59-3334

FAX 0468-59-3365

e-mail [nakao@isl.ntt.co.jp](mailto:nakao@isl.ntt.co.jp)

##### 関連する特許権

1. 特許出願番号 (公開番号) 60-42052  
名称 署名文書通信方式  
出願日 1985 年 3 月 4 日
2. 特許出願番号 (公開番号) 59-052696  
名称 署名文書通信方式  
出願日 1984 年 3 月 19 日
3. 特許出願番号 (公開番号) US 4625076  
名称 Signed Document Transmission System  
出願日 1985 年 3 月 11 日
4. 特許出願番号 (公開番号) Canada 1255784  
名称 Signed Document Transmission System  
出願日 1985 年 3 月 14 日
5. 特許出願番号 (公開番号) EP 0157258  
名称 Signed Document Transmission System  
出願日 1985 年 3 月 15 日

関連する著作権 「(2) 暗号技術仕様書」「(3) 自己評価書」「(5) 参照プログラムおよびその仕様書、テストベクトル生成プログラムおよびその仕様書」「(7) 応募暗号説明会発表資料」の著作権は日本電信電話株式会社に帰属します。

関連する特許とその扱い 弊社の知る範囲では、関連する他社の特許はありません。

電子政府で使用される際のライセンス方針 上記特許に関しては、応募暗号技術 (ESIGN) を使用するものに対して、相互主義の下に、非排他的に、無償で実施許諾する方針です。

### 7.2.1.2 ECIES (Elliptic Curve Integrated Encryption Scheme) in SEC1

#### 公開ホームページ URL

和文 [http://www.labs.fujitsu.com/theme/crypto/public\\_key2001.html](http://www.labs.fujitsu.com/theme/crypto/public_key2001.html)

英文 [http://www.labs.fujitsu.com/theme/crypto/public\\_key2001\\_e.html](http://www.labs.fujitsu.com/theme/crypto/public_key2001_e.html)

学会発表等 SECG(Standards for Efficient Cryptography Group)、“SECG standards”  
ウェブページで公開 <http://www.secg.org/>、2000年9月20日

#### 調達窓口

担当者氏名 下江 達二

所属・役職 ソフトウェア事業本部運用管理ソフトウェア事業部 第二開発部 担当  
部長

企業・団体名 富士通株式会社

所在地 〒222-0033 神奈川県横浜市港北区新横浜 3-9-18 (TECH ビル)

TEL 045-474-1925

FAX 045-474-1953

e-mail [shimoe@jp.fujitsu.com](mailto:shimoe@jp.fujitsu.com)

関連する特許権 SECG member patent letters(下記アドレス)をご参照ください。  
[http://www.secg.org/collateral/certicom\\_secg\\_patent.pdf](http://www.secg.org/collateral/certicom_secg_patent.pdf)

関連する著作権 License to copy the document is granted provided it is identified  
as “Standards for Efficient Cryptography (SEC)”, in all material mentioning or  
referencing it.

関連する特許とその扱い 特許の扱いについては、下記の SECG Patent Policy をご参照  
ください。 [http://www.secg.org/patent\\_policy.htm](http://www.secg.org/patent_policy.htm)

電子政府で使用する際のライセンス方針 富士通株式会社の所有する特許は、合理的な条  
件で、提供先を差別することなく、実施権を供与する。

### 7.2.1.3 HIME(R) (High Performance Modular Squaring Based Public Key Encryption (Revised version))

#### 公開ホームページ URL

和文 <http://www.sdl.hitachi.co.jp/crypto/>

英文 <http://www.sdl.hitachi.co.jp/crypto/>

学会発表等 佐藤 尚宜、“効率的かつ安全性証明可能な公開鍵暗号方式 HIME(R)” 電子  
情報通信学会 ISEC 研究会、2001年5月18日

#### 調達窓口

担当者氏名 金野 千里

所属・役職 (株)日立製作所 情報・通信グループ統括本部 チーフマネージャー

企業・団体名 (株)日立製作所

所在地 〒140-8572 東京都品川区南大井 6-27-18 日立大森第二別館

TEL (03)5471-8922

FAX (03)5471-2565

e-mail [c-konno@itg.hitachi.co.jp](mailto:c-konno@itg.hitachi.co.jp)

#### 関連する特許権

1. 特許出願番号(公開番号) 特願 2001-284363

名称 「 $N=p^d q$  を法とする剰余環上のべき乗根計算方法及び公開鍵暗号方法お  
よび装置」

**出願日** 2001 年 9 月 19 日

**関連する著作権** 本暗号技術 HIME(R) を応募するにあたり提出される文書、ソースコードなどの著作権はすべて日立製作所に帰属します。

**関連する特許とその扱い** 弊社は、HIME(R) の技術に関し、弊社による上記の出願特許および登録特許が関連すると考えられます。弊社と致しましては、HIME(R) が本提案に採用された場合には、上記特許を非差別的、かつ適正な対価条件でライセンス致します。ただし、相手方が相手方の HIME(R) 関連特許を非差別的、かつ適正な対価条件でライセンスしない場合はこの限りではございません。

**電子政府で使用する際のライセンス方針** 上記記載の内容に同じです。

### 7.2.1.4 PSEC-KEM Key agreement

#### 公開ホームページ URL

**和文** <http://info.isl.ntt.co.jp/>

**英文** <http://info.isl.ntt.co.jp/>

**学会発表等** 岡本 龍明、「公開鍵暗号「EPOC」および「PSEC」」ISEC 研究会、2000 年 5 月 25 日

#### 調達窓口

**担当者氏名** 中尾 昌善

**所属・役職** NTT 情報流通プラットフォーム研究所主幹研究員

**企業・団体名** 日本電信電話株式会社 (NTT)

**所在地** 〒239-0847 神奈川県横須賀市光の丘 1-1-609A

**TEL** 0468-59-3334

**FAX** 0468-59-3365

**e-mail** nakao@isl.ntt.co.jp

#### 関連する特許権

1. **特許出願番号 (公開番号)** 10-320172  
**名称** ランダム関数利用公開鍵暗号の暗号装置、復号装置  
**出願日** 1998 年 11 月 11 日
2. **特許出願番号 (公開番号)** 2000-32461  
**名称** 暗号化装置、方法、復号装置、方法、暗号システム及びプログラムを記憶した記憶媒体  
**出願日** 2000 年 2 月 9 日

**関連する著作権** 「(2) 暗号技術仕様書」「(3) 自己評価書」「(5) 参照プログラムおよびその仕様書、テストベクトル生成プログラムおよびその仕様書」「(7) 応募暗号説明会発表資料」の著作権は日本電信電話株式会社に帰属します。

**関連する特許とその扱い** 弊社の知る範囲では、関連する他社の特許はありません。

**電子政府で使用する際のライセンス方針** 上記特許に関しては、応募暗号技術 (PSEC-KEM) を使用するものに対して、相互主義の下に、非排他的に、無償で実施許諾する方針です。

## 7.2.2 共通鍵暗号技術

### 7.2.2.1 MUGI

#### 公開ホームページ URL

**和文** <http://www.sdl.hitachi.co.jp/crypto/mugi/>

**英文** <http://www.sdl.hitachi.co.jp/crypto/mugi/index-e.html>

**学会発表等** 渡辺大、“PANAMA 型疑似乱数生成器の乱数性” 電子情報通信学会 ISEC 研究会、2001年9月17日

**調達窓口**

**担当者氏名** 原野 紳一郎

**所属・役職** (株)日立製作所ソフトウェア事業部 システム管理ソフトウェア本部 推進部長

**企業・団体名** (株)日立製作所ソフトウェア事業部 システム管理ソフトウェア本部

**所在地** 〒244-8555 神奈川県横浜市戸塚区戸塚町 5030 番地

**TEL** (045)862-8715

**FAX** (045)865-9010

**e-mail** harano@itg.hitachi.co.jp

**関連する特許権**

1. **特許出願番号 (公開番号)** 特願 2001-013959  
**名称** 疑似乱数生成方法またはそれを用いた暗号復号処理装置  
**出願日** 2000年1月23日
2. **特許出願番号 (公開番号)** 特願 2001-145783  
**名称** 疑似乱数生成方法またはそれを用いた暗号復号処理装置  
**出願日** 2000年5月16日
3. **特許出願番号 (公開番号)** 特願 2001-274433  
**名称** 疑似乱数生成方法またはそれを用いた暗号復号処理装置  
**出願日** 2000年9月11日

**関連する著作権** 本暗号技術 MUGI を応募するにあたり提出される文書、ソースコードなどの著作権はすべて日立製作所に帰属します。

**関連する特許とその扱い** 弊社は、MUGI の技術に関し、弊社による上記出願特許および登録特許が関連すると考えられます。弊社と致しましては、MUGI が本提案に採用された場合には、上記特許を非差別的、かつ適正な対価条件でライセンス致します。ただし、相手方が相手方の MUGI 関連特許を非差別的、かつ適正な対価条件でライセンスしない場合はこの限りではございません。

**電子政府で使用する際のライセンス方針** 同上

### 7.2.2.2 RC4

**問い合わせ先** RSA セキュリティ社 (<http://www.rsasecurity.co.jp/>)

**仕様** 次の文献に記載されているもの

S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4,” Lecture Notes in Computer Science 2259, pp.1-24, Springer-Verlag, 2001

## 第 8 章

# 評価暗号一覧

応募暗号技術の略称名を [ ] 内に記載。

### 1. 公開鍵暗号技術

#### (a) 署名

##### i. DSA

2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。

##### ii. ECDSA (ANSI X9.62)

2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。

##### iii. ECDSA (Elliptic Curve Digital Signature Algorithm) in SEC1 [ECDSA]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。

##### iv. ESIGN signature [ESIGN]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。

##### v. OK-ECDSA [OK-ECDSA]

2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 2 章を参照。

##### vi. RSA 署名 (PKCS#1 v1.5<sup>\*1</sup>)

2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。

##### vii. RSA Public-Key Cryptosystem with Probabilistic Signature Scheme (RSA-PSS) [RSA-PSS]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。

#### (b) 守秘

##### i. ECIES (Elliptic Curve Integrated Encryption Scheme) in SEC1 [ECIES]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 2 章を参照。

##### ii. EPOC-2 encryption [EPOC-2]

---

<sup>\*1</sup> 本方式は、規格書 RSA PKCS#1v1.5 で規定され、規格書 RSA PKCS#1v2.0 以降にも引き継がれているため、本報告書では、方式名として RSA-PKCS#1v1.5 と略記する。

- 2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。
- iii. HIME(R)(High Performance Modular Squaring Based Public Key Encryption (Revised version)) [HIME(R)]  
2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 2 章を参照。
  - iv. NTRU public key cryptosystem [NTRU]  
2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 2 章を参照。
  - v. RSA Public-Key Cryptosystem with Optimal Asymmetric Encryption Padding (RSA-OAEP) [RSA-OAEP]  
2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 2 章を参照。
- (c) 鍵共有
- i. Common private Complex Key System [Cock system]  
2001 年度応募暗号技術であり、スクリーニング評価を実施したが、さまざまなプラットフォームでの実装や互換性の実現が困難であり、かつ、安全性に関して鍵共有としての機能が達成されていないと判断されたため、以下の理由により評価を終了した。
    - 実数を扱うので、データの表現に無限のビット長が必要である。
    - 公開情報と通信路上の情報から秘密情報が容易に判明し、共有鍵も求まる。
  - ii. DH  
2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 2 章を参照。
  - iii. ECDH (Elliptic Curve Diffie-Hellman Scheme) in SEC1 [ECDH]  
2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 2 章を参照。
  - iv. OK-ECDH [OK-ECDH]  
2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 2 章を参照。
  - v. PSEC-KEM Key agreement [PSEC-KEM]  
2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 2 章を参照。
2. 共通鍵暗号技術
- (a) 64 ビットブロック暗号
    - i. CIPHERUNICORN-E [UNI-E]  
2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 3 章を参照。
    - ii. Hierocrypt-L1 [HC-L1]  
2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。
    - iii. MISTY1 [MISTY1]  
2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。
    - iv. RC2  
2001 年度暗号技術検討会から評価依頼があった暗号技術であり、詳細評価

を実施した。評価結果については本報告書第 6 章を参照。

v. Triple DES

2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。

(b) 128 ビットブロック暗号

i. Advanced Encryption Standard (AES)

2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。

ii. Camellia [Camellia]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。

iii. CIPHERUNICORN-A [UNI-A]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 3 章を参照。

iv. Hierocrypt-3 [HC-3]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。

v. RC6 Block Cipher [RC6]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。

vi. SC2000 [SC2000]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 3 章を参照。

vii. SEED

2001 年度暗号技術検討会から評価依頼があった暗号技術であり、詳細評価を実施した。評価結果については本報告書第 3 章を参照。

(c) ストリーム暗号

i. C4-1 [C4-1]

2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。

- 提出書類には、第三者実装が可能と考えられるだけの十分なアルゴリズム情報が記載されていない。
- 参照プログラムには応募暗号本体の記述がないことから、評価するための十分な書類が提出されていない。

ii. FSAnGo [FSAnGo]

2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。

- 評価に必要な、テキストファイル (ソースプログラム) の参照プログラム、テストベクタ生成プログラムが提出書類中がない。

iii. MUGI

2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 3 章を参照。

iv. MULTI-S01 [S01]

2000 年度評価実施暗号技術。2001 年度応募暗号技術であり、詳細評価を実施した。評価結果については本報告書第 3 章を参照。

- v. RC4  
2001 年度暗号技術検討会から評価依頼があった暗号技術であり、詳細評価を実施中。
3. ハッシュ関数
- (a) RIPEMD-160  
2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 4 章を参照。
- (b) SHA-1  
2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 4 章を参照。
- (c) draft SHA-256, 384, 512  
2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、詳細評価を実施した。評価結果については本報告書第 4 章を参照。
4. 擬似乱数生成系
- (a) Creation of intrinsic random numbers with Clutter Box [Clutter Box]  
2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。
- 特殊なハードウェアを必要とする方式であり、提出書類をベースにした評価を行うことは困難である。
  - 提出書類には乱数生成のアルゴリズムに関する十分な情報が記載されていない。
- (b) FSRansu [FSRansu]  
2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。
- 評価に必要な、テキストファイル (ソースプログラム) の参照プログラム、テストバクタ生成プログラムが提出書類中がない。
- (c) High security ultra mini random number generator [RNE]  
2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。
- 特殊なハードウェアを必要とする方式であり、提出書類をベースにした評価を行うことは困難である。
  - 参照プログラムは乱数系列を観測するプログラムであり、提出書類が評価を行うための条件を満足していない。
- (d) PRNG based on SHA-1  
2000 年度評価実施暗号技術。2001 年度暗号技術評価委員会で評価が必要と判断した暗号技術であり、監視状態の暗号として詳細評価を実施した。なお、評価結果については本報告書第 5 章を参照。
- (e) TAO TIME Cognition Algorithm [TAO TIME]  
2001 年度応募暗号技術であり、スクリーニング評価を実施した。評価結果については本報告書第 5 章を参照。
5. その他
- (a) The Security System for Information Telecommunication using the unconditional secrecy technology [CVCRT]  
2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。
- 提出書類の記載内容が要求仕様に対する単なる開発計画に過ぎず、詳細な技術情報が不足しており、基本的に評価不能である。
  - ランダム圧縮が「一方向性」に見えるのは、単に像から逆像に戻すための情報を潰しているからであって、例えば、衝突困難なハッシュ関数と比べて優



れていると主張する根拠は何もない。また、なぜ  $24 \times 24$  のマトリクスが適しているのかの合理的根拠もない。

- 攪乱信号を効率的に構成するためのアルゴリズムが明示されていない。また、攪乱がなぜ有効なのかの根拠がない。

(b) Security-up for Kana Kanji System Applied Mathematics Using Magic Squares [MKS]

2001 年度応募暗号技術であり、スクリーニング評価を実施したが、以下の理由により評価を終了した。

- 提出書類の記載内容が提出されたプログラムの機能と動作環境の説明であって、その機能の原理とアルゴリズムに関する記載がなく、基本的に評価不能である。
- 提出書類に記載されている安全性の根拠が合理的でない。
- 提案の内容がそもそも公開鍵暗号技術であるとは認められない。



# 索引

- A3 関数, 112
- AER 仮定, 42
- AER 問題, 42
- anomalous binary curve, 39
- ANSI X9.62, 37, 65
- ANSI X9.63, 70
- authenticated encryption, 127
  
- Brent の予測式, 28
- B 関数, 158
  
- CA, 198–201
- Carter-Wegman MAC, 128
- CBC モード, 205, 207
- CMA, 43, 44
- Convolution Modular Lattice, 68
- CRL, 194
- CRT, 69
  
- DDH 問題, 61
- Decisional Diffie-Hellman 問題, 64
- DES, 202
- DES Challenge, 203
- Deviation Parameter, 128
- DPA 攻撃, 65, 70
  
- ECC challenge, 31
- ECM, 23–25
- ElGamal 暗号, 71
- $e$  乗根近似問題, 40, 42–44
  
- FIPS 186-2, 34, 39
- FIPS-180, 166
- FIPS186-2, 182
- FIPS186-2 change notice, 34, 35, 39
- FL 関数, 139, 144
- Fortezza, 196
- forward-secrecy, 63
- Frobenius 写像, 39
  
- Gap-Diffie-Hellman 問題, 61
- generic DSA, 38
- generic group model, 36–38, 66
- generic group オラクル, 38
- generic model, 66
- GNFS, 23–25
- $G$  関数, 118
  
- IETF, 191
- IND-CCA2, 47, 61, 68, 69, 72
- IND-CPA, 68
- Integral Cryptanalysis, 140
- I 関数, 158
  
- KASUMI, 140
  
- Koblitz 曲線, 31, 36, 39, 61, 63
  
- lattice reduction technique, 35
- Lenstra-Verheul, 28, 32
- LFM, 24, 25
- LLL アルゴリズム, 43, 68
  
- man-in-the-middle 攻撃, 192, 195
- MASH, 206
- MIX, 206
- MT 関数, 112
  
- $n = p^2q$  型素因数分解問題, 42
- $N = p^d q$  型の素因数分解問題, 69
  
- OAEP, 69
  
- PANAMA, 124
- PKCS #1 v1.5, 48
- P 関数, 145
  
- Rabin-OAEP, 69
- Rabin-SAEP, 69
- Replay 攻撃, 195
- RIPEMD, 177
- RIPE プロジェクト, 175
- RSA-FDH, 48
- RSA-OAEP, 47, 69
- RSA-PSS, 47
- RSA 署名, 47
- RSA プリミティブ, 47
- R 関数, 158
  
- SECG, 60, 62, 72
- Single-Occurrence 適応的選択文書攻撃, 43
- SO-CMA, 43, 44
- SP 800-22, 131
- SPA 攻撃, 65, 70
- SSL, 191, 195
- S 関数, 145
  
- T0 関数, 112
- TLS, 191, 195
- Traffic Analysis 攻撃, 195
- T 関数, 112
  
- Vernam 暗号, 128
- version rollback 攻撃, 192, 195
  
- アバランシュ性評価, 77
- 暗号強度評価支援システム, 101, 111
- 暗号文一致攻撃, 205, 207
- 暗号文単独攻撃, 75
- 安全性余裕, 87
- 一時鍵生成部, 112

- 一方向性, 165
- 入れ子型 SPN 構造, 150
- 鍵カプセル化メカニズム, 71, 72
- 鍵衝突攻撃, 103
- 鍵導出関数, 64
- 学術的な解説, 88
- 頑強性, 47
- 既知平文攻撃, 75
- 強秘匿, 61, 69, 72
- 近隣ベクトル問題, 68
- クロス接続, 158
- 原像計算困難性, 171
- 公開鍵証明書, 196
- 公開鍵証明書無効化リスト, 194
- 高階差分攻撃, 83
- 攻撃が成功, 75
- 格子基底縮小アルゴリズム, 43
- 最大差分確率, 76
- 最大差分特性確率, 76
- 最大線形確率, 76
- 最大線形特性確率, 76
- サイドチャンネル攻撃, 66, 67, 70, 71, 155
- シード鍵, 112
- 指数関数時間, 24
- 衝突, 165
- 衝突困難性, 171
- 初等統計量評価, 102
- 署名オラクル, 43, 44
- 実際の安全性保証, 83
- 弱鍵, 103
- 受動的攻撃, 70
- 準指数関数時間, 23
- ストレート接続, 158
- 制御型高階差分攻撃, 146
- セキュリティプロトコル, 195
- セキュリティホール, 192, 193, 195
- セキュリティモデル, 72
- 選択平文攻撃, 75
- 存在的偽造不可, 40, 41, 43, 44
- タイミング攻撃, 65, 70
- 多項式時間, 24-26
- 楕円曲線パラメータ, 30, 39, 60, 62, 72
- 中間一致攻撃, 202
- 適応的選択暗号文攻撃, 47, 61, 69, 72, 192
- 適応的選択文書攻撃, 38, 40, 41, 43, 44
- 電力解析攻撃, 66, 71
- 匿名認証モード, 194
- トラップドア, 39
- 認知アルゴリズム, 189
- 能動的攻撃, 70
- ハイブリッド暗号, 72
- 平方根演算, 69
- 平方無縁, 26
- 平方無縁部分, 26
- 変形版  $e$  乗根近似問題, 45
- 補間攻撃, 83
- 無衝突性を持つ, 165
- メッセージスケジュール関数, 168
- モンゴメリ型, 66, 70
- モンゴメリ型楕円曲線, 65, 66, 70
- ラティスの最短ベクトル問題, 67, 68
- ランダムオラクルモデル, 35, 37, 40, 41, 43, 44, 61, 66, 69, 70, 72
- ランダム化射影座標, 65, 70
- 利用モード, 202
- ワイエルシュトラス型楕円曲線, 66, 70

# 付録 CD-ROM について <sup>\*1</sup>

付録 CD-ROM には

- (a) 応募暗号技術の技術情報
- (b) SSL 調査報告

が含まれています。

1. この CD-ROM に収録した応募暗号技術の技術情報は、2001 年度に評価の対象とした応募暗号に関する技術情報の詳細を参考情報として提供するものです。
2. この CD-ROM に収録した技術情報に対する著作権等の知的財産権は、応募者に帰属しています。
3. 本 CD-ROM に収録した自己評価書は、応募者の提出していただいた自己評価書をそのまま収録しています。従って、暗号技術評価委員会の見解とは異なることがあります。暗号技術評価委員会の評価結果については、CRYPTREC Report 2001 本編をご覧ください。
4. 本 CD-ROM に収録した技術資料は、2001 年の応募時点で、提出いただいた資料をそのまま収録しています。資料提出後の内容の改訂や誤記訂正については、CRYPTREC のホームページで公開してゆく予定です。

CD-ROM の詳細については `readme.txt` を御覧下さい。

---

<sup>\*1</sup> PDF 版には付属していません。



**不許複製 禁無断転載**

**発行日** 2002(平成 14) 年 3 月

**発行者**

- 〒113-6591

東京都文京区本駒込二丁目 28 番 8 号  
(文京グリーンコートセンターオフィス 16 階)  
情報処理振興事業協会 (セキュリティセンター)  
電話: 03-5978-7508, FAX: 03-5978-7518

INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN  
BUNKYO GREEN COURT CENTER OFFICE  
2-28-8 HONKOMAGOME, BUNKYO-KU  
TOKYO, 113-6591 JAPAN  
TEL: +81-3-5978-7508, FAX: +81-3-5978-7518 (IT SECURITY CENTER)

- 〒105-0014

東京都港区芝二丁目 31 番 19 号  
(バンザイビル)  
通信・放送機構 研究企画管理部研究企画課  
電話: 03-3769-6810, FAX: 03-5441-7584

TELECOMMUNICATIONS ADVANCEMENT ORGANIZATRION of JAPAN  
BANZAI Bld. 2-31-19 SHIBA, MINATO-KU  
TOKYO, 105-0014 JAPAN  
TEL: +81-3-3769-6810, FAX: +81-3-5441-7584