暗号アルゴリズム「ＲＣ６」
詳細評価（攻撃評価）レポート

２００１年1月１２日

目次

１．RC6 の概要

RC6 は、Ron Rivest, Matt Robshaw, Ray Sidney および Yiqun Yin により１９９７年に設計されたブロック暗号であり、ＡＥＳの最終審査対象５候補のひとつに選定されている。RC6 はブロック長、ラウンド数、鍵長がともに可変なスケーラブルブロック暗号である点がひとつの特徴である。このうちＡＥＳに投稿されたものは、ブロック長１２８ビット、段数２０段、鍵長１２８、１９２、２５６ビットのものであるので、本評価レポートにおいてもこのパラメータをもつ RC6 を評価対象とする。

RC6 は、これに先立つ１９９５年に設計されたスケーラブルブロック暗号 RC5 をもとに、ＡＥＳコンテスト向けに再設計されたものと考えてよい。RC5 の特長である記述の簡易性とスケーラビリティを受け継ぎ、さらにソフトウエアでの高速性を追求したものである。RC6 はテーブル参照を一切おこなわず、３２ビット単位の算術加算・乗算と、データ依存回転シフトによってデータの攪拌を行っている。これは３２ビットプロセッサではメモリアクセスの頻度のきわめて少ない高速な実装が可能なことを示している。

RC6 は、３２ビットプロセッサではＡＥＳの最終審査対象５候補のうち最も高速であると言っておおむねよいが、一方３２ビット乗算やデータ依存回転シフトの多用は、プラットフォームによっては速度の低下をまねくことがある。例えばローエンドＩＣカードむけの８ビットマイコンでは、RC6 は低速である。またハードウエアでは、RC6 のスループット（平文入力から暗号文出力までの時間）は非常に低速である。このように RC6 は結果的にＡＥＳの標準プラットフォームであった３２ビットプロセッサにターゲットを絞った暗号であるということができる。

RC6 はＡＥＳ候補として、その安全性についてはこれまで数多くの研究がなされてきた。そこでここでは安全性の評価項目ごとに、設計者の評価内容とこれまでの研究結果をふまえて、評価者の考察をのべることとする。

２．ＲＣ６の安全性評価

２．１　差分解読法に対する安全性

【自己評価書における記述】

自己評価書の添付資料には、非常に詳細に RC6 の差分解読法に対する強度評価の検討がなされている。これによれば RC6 の解読に必要な選択平文数は以下のとおりである。

段数	8	１２	１６	２０
選択平文数	2^{56}	2^{117}	2^{190}	2^{238}

これによって RC6 は、１６段で差分解読法に必要な平文数が理論上得ることの出来る平文数を超えることになり、したがって完全仕様の２０段の RC6 は十分安全であると結論を導き出している。

【考察】

RC6 の差分解読法に対する安全性は、その「親」のアルゴリズムである RC5 の時代から、幅広く研究されており、この自己評価書における評価もその方法論に沿ったものである。RC6 のようにデータ依存回転シフトを含むようなアルゴリズムでは、回転数によって差分の経路がかわるため、これら経路ごとの特性確率の和、すなわち差分確率(differential probability)の考察が不可欠である。自己評価書ではこの点も十分考慮されており、内容的にも十分信頼に足ると考えられる。

なおこの評価結果は、完全な形での差分確率の上からの評価(いわゆる差分解読法に対する provable security)を実現したものではないことに注意する必要がある。すなわち、真の差分確率の最大値はこの評価結果よりも大きくなり、したがって解読に必要な選択平文数は実際にはこれより少ない可能性が残っている。しかしながら、一般に与えられたブロック暗号アルゴリズムの差分確率の最大値を求めることはきわめて困難であり、差分確率の最大値が求められていないことを理由に評価が不足しているということはできない。

２．２　線形解読法に対する安全性

【自己評価書における記述】

自己評価書の添付資料には非常に詳細に RC6 の線形解読法に対する強度評価の検討がなされている。これによれば RC6 の解読に必要な既知平文数は以下のとおりである。

段数	8	１２	１６	２０
既知平文数	2^{47}	2^{83}	2^{119}	2^{155}

これによって RC6 は、１６段で線形解読法に必要な平文数が理論上得ることの出来る平文数とほぼ均衡することになり、したがって完全仕様の２０段の RC6 は十分安全であると結論を導き出している。

【考察】

RC6 の線形解読法に対する安全性は、その「親」のアルゴリズムである RC5 の時代から、幅広く研究されており、この自己評価書における評価もその方法論に沿ったものである。RC6 のようにデータ依存回転シフトを含むようなアルゴリズムでは、回転数によって近似の経路がかわるため、これら経路ごとの特性確率の和、すなわち線形確率(linear hull probability)の考察が不可欠である。自己評価書ではこの点も十分考慮されており、内容的にも十分信頼に足ると考えられる。

なおこの評価結果は、完全な形での線形確率の上からの評価(いわゆる線形解読法に対する provable security)を実現したものではないことに注意する必要がある。すなわち、真の線形確率の最大値はこの評価結果よりも大きくなり、したがって解読に必要な既知平文数は実際にはこれより少ない可能性が残っている。しかしながら、一般に与えられたブロック暗号アルゴリズムの線形確率の最大値を求めることはきわめて困難であり、線形確率の最大値が求められていないことを理由に評価が不足しているということはできない。

２．３　差分線形解読法に対する安全性

【自己評価書における記述】

（自己評価書添付資料からの引用）
差分線形暗号解析法は、Crypto'94 においてLangfordとHellmanによって紹介された。洗練されているといえるこの攻撃法では、暗号処理の過程を通じて、2つの文要素の差違を推測するのに差分を使っている。この差分を知ることで暗号処理過程の後段階で線形近似を使うことが可能になる。十分良い線形近似が得られた場合には、差分解析のみを使った場合よりも、暗号解読の可能性が高まることも時にはある。この方法に可能性があるとするなら、それはラウンド数が多い暗号を少ないデータ量で攻撃する場合である。この方法は、8ラウンドのDES に対しては、現在最も強力な攻撃法であると言えるが、適用範囲は広くない。攻撃の開始段階で有効な差分が存在すること、後段階では有効な線形近似が存在することが必要条件になっている。RC6 に関する我々の差分および線形暗号解析の研究では、これらの2つの条件が同時に成り立つことは、特に完全20ラウンドのRC6 に対して差分線形攻撃を企てようとする場合には、特にあり得ないことが明らかになっている。

【考察】

この文章は、結論は正しいものの若干説得力に欠けている。事の本質は、差分線形解読法に必要な選択平文数は、前半でえられた差分確率の２乗および後半でえられた線形偏差の４乗に反比例してしまうところにある。したがって、前半の差分確率および後半の線形偏差がそれぞれ非常に大きな（１に近い）値でなければ、差分線形解読法の優位性はあらわれない。このため通常、差分線形解読法は、段数の少ないアルゴリズムでのみ成立する。このような理由から、RC6に対する差分線形解読法が、差分解読法や線形解読法よりも効果があるとは考えられないのである。

２．４　高階差分解読法／補間攻撃に対する安全性

【自己評価書における記述】

（自己評価書添付資料からの引用）
微積分における微分演算の類推から考えられた高次差分（*higher-order differential*）攻撃がLaiによって検討された。Knudsenはこれに続いて、通常の差分攻撃には強いが高次差分攻撃には弱い暗号を作ることができることを実証した。同様に、補間（*interpolation*）攻撃がある種の暗号に対して有効な攻撃法であることも示された。けれども、これらの攻撃法は双方とも十分なラウンド数を使ったより高度な暗号に対しては幾分限界があるものと思われる。以上がRC6に関する我々の経験である。

【考察】

いかなるブロック暗号においても、平文と暗号文と鍵（あるいは副鍵）との関係を代数式(algebraic form)であらわすことが理論上可能である。そしてその代数式が簡易であればこの式を数学的に解くことによって解読できる可能性がある。これが、高階差分解読法や補間攻撃とよばれる解読法の基本的な考え方である。すなわち、これらの解読法が成立するためには、まず暗号アルゴリズムを記述する代数式が、明示的な形で書き下せる程度の複雑さであるかどうかが、第１の関門となる。

普通これらの解読法が成立するのは、「数学的に美しく」あるいは「簡単な論理式で」作られた参照テーブルと、論理演算だけから構成されるような暗号アルゴリズムの場合である。これに対してRC6には、３２ビット乗算、データ依存回転シフト、排他的論理和演算が含まれている。このように、算術演算と論理演算を組み合わせた暗号アルゴリズムを代数式で書き下すのは、式の項数が多すぎて困難である。したがって自己評価書にあるように、RC6は高階差分解読法や補間攻撃で解読されるとは考えにくい。

２．５　短縮差分解読法に対する安全性

【自己評価書における記述】

（自己評価書添付資料からの引用）
Knudsenは偏差分（*truncated differentials*）（正式には*partial differentials*）の考え方を導入した。想定されているのは、暗号解読者が128ビットすべての差分ではなく、一部の差分を予測しようとする場合である。事実、差分の一部の予測に成功すると、時には鍵の関連情報の再生につながる場合がある。偏差分を使うことで、暗号解読者には暗号攻撃にさらに自由度が与えられる可能性があるが、多くの場合その適用範囲には限界があるものと思われる。我々はしばしば我々の基本的な攻撃法の改良として偏差分の大まかな考え方を利用した。時には認識できる差分が発生する可能性を高めようとする試みにおいて、ある特性あるいは差分を持つ特定のワードの動きを予測する機会の利用は控えた。これによって、平文の必要量を減少させる結果につながるが、これは、偏差分によるフル攻撃というよりも、最適化の工夫と見るほうが良いであろう。 確かに、偏差分による解析に耐えない暗号をつくることはできるが、この攻撃法、あるいはそれに近い類型モデルが、さらに本気で作られた対抗暗号に対して依然として有効性を持つことは非常に少ないはずである。これに関する注目すべき事例がSaferとSkipjackによって紹介されている。

【考察】

短縮差分解読法とは、特性(characteristic)の遷移状態をビットよりも大きい単位で（例えばバイト単位で）扱うことによって、差分(differentials)を効率よく求め、これにより通常の差分解読法よりも効率のよい解読をめざすものである。この解読法が成立するためには、アルゴリズムの主要な部分がこの単位を基本とする演算で構成されており、しかも１ブロック内におけるこの単位の数が多いなど、強い構造をもつことが必要である。

RC6はアルゴリズム全体が３２ビット演算で構成されているため、短縮差分解読法の第１の前提条件は満足しているが、１ブロックが１２８ビットであるため単位数が１２８／３２＝４とすくなく第２の前提条件を満たしていない。したがって自己評価書にあるように、短縮差分解読法をそのままRC6に適用するのは困難であると考えられる。

２．６　不能差分攻撃に対する安全性

【自己評価書の記述】

（自己評価書添付資料からの引用）
また、*impossible differentials* による攻撃に対する強度を高めるには、データ依存の回転が効果を持つものと思われる。

【考察】

通常の差分解読法では差分特性確率が小さいという性質は安全性の観点から望ましいことであるが、不能差分攻撃では逆に確率０の差分経路がある場合、すなわち絶対に起こらない差分波及パターンが多ラウンドにわたって存在する場合に、このことを手がかりに拡大鍵の情報を絞りこむことができるというものである。例えばラウンド関数が全単射であるような５段のＦｅｉｓｔｅｌ暗号には、ラウンド関数の構造によらず必ず不能差分が存在することが知られている。

RC6 は Feistel とは異なった暗号構造をもっており、その不能差分攻撃に対する強度をしらべることは意味のないことではないと思われる。ただし現時点では RC6 の不能差分について考察された研究は知られていない。直感的には RC6 の一段の複雑さを考えると、１０段以上で不能差分があるとは考えにくいが、正確には今後の研究をまたなければならない。自己評価書の内容も、著者らが不能差分攻撃に対する安全性評価をまだ本格的には行っていないことを示しているように思われる。

２．７　ブーメラン攻撃に対する安全性

【自己評価書における記述】

（自己評価書添付資料からの引用）
また一部に存在する攻撃法である、*boomerang* や *mod n* などの暗号解析法はRC6には適用不可能であると思われる。

【考察】

ブーメラン攻撃は、差分解読法を応用した適応的選択暗号文攻撃手法であり、その原理は、ある差分値をもつ選択平文から得られた暗号文をもとに、そこから解読者が適当な差分値をもつ選択暗号文を新たに生成し、それを今度は復号して得られた平文の差分値を観測するというものである。この解読法は暗号アルゴリズムを例えば上半分と下半分に分割した場合に、上半分の最大平均差分確率と下半分の最大平均差分確率がともに非常に大きい場合に成立する。

実際この解読法に必要な平文の数は少なく見積もってもこれらの確率の積の逆数程度になり、この点差分線形解読と似ているともいえる。結論としては、自己評価書にあるようにRC6 に対してこのブーメランの攻撃が成立するということは考えにくいと言ってよい。

２．８　χ^2攻撃に対する安全性

【自己評価書における記述】

（自己評価書からの引用）
一種の統計的な手法であるχ^2解析も考えられてはいるが、デザイナによって存在が確認されているRC6 に対する線形暗号攻撃よりも有効ではない。

【考察】

自己評価書の記述とは異なり、このχ^2攻撃は現在知られているRC6に対する攻撃の中で、最も強力なものであると考えられており、この攻撃の存在がゆえにRC6は他のＡＥＳ最終審査対象候補に比べてセキュリティマージンが比較的少ないといわれている。RC6 に対するχ^2攻撃はこれまで３回発表されており、その最初のものは Vaudenay らによる第２回ＡＥＳ会議での報告であり、これを発展させたものがＦＳＥ２０００における Vaudenay らによる論文と Knudsen らによる論文の２本である。参考までに、Knudsen と Meier による RC6 のχ^2攻撃結果は以下のとおりである。

Distinguishing Attack

段数	7	9	11	13	15	17(*)
選択平文数	$2^{46.2}$	$2^{62.4}$	$2^{78.6}$	$2^{94.8}$	$2^{111.0}$	$<2^{118}$

(*) １７段の場合は2^{80}個に１つの鍵の割合で発生する

Key Recovery Attack

段数	１２	１４	１４	１６(*)	１５
選択平文数	2^{94}	2^{110}	2^{108}	2^{118}	2^{119}
計算量	2^{119}	2^{135}	2^{160}	2^{171}	2^{215}
メモリ	2^{42}	2^{42}	2^{74}	2^{74}	2^{138}

(*) １６段の場合は2^{60}個に１つの鍵の割合で発生する

自己評価書添付資料に記された線形解読法は、「少なくともこれだけの平文が線形解読に必要である」との下から評価であったのに対し、Knudsenの結果は、「これだけの平文数が集まればχ^2解読が可能である」といった上からの評価結果であったことに注意すべきである。基本的にχ^2攻撃は線形解読法のある種の一般化であり、自己評価書の線形解読法と類似の結果が得られることは不思議ではない。その意味でKnudsenの結果は、自己評価書の結果の解読者側からの精密化という見方ができる。

この解読法のもうひとつのポイントは鍵の値による強度の変化である。シフト数可変の回転シフト演算を含むブロック暗号では、鍵の値と選択平文の組み合わせによって回転シフト数のバラエティが減少し、解読の効率があがることがあり、上記の Knudsen, Meier の解読法はまさにこのことをとらえたものである。とくに Distinguishing Attack の１７段は特定の鍵についてのみ成り立つ攻撃であるが、設計者の予想より少ない選択平文数で成立している。広い意味でこのような鍵も一種の弱鍵と呼ばれることがある。

ただしこの１７段攻撃が可能になる鍵は2^{80}個に１個であり、例えば１２８ビット鍵ならば全部で2^{48}個程度しか存在しない。したがって１２８ビットの鍵をもつ RC6 ならば、この2^{48}個を総当りするほうがはるかに効率のよい攻撃となる。１９２ビット鍵の場合も同様である。すなわちこの１７段攻撃は２５６ビット鍵をもつRC6の場合にのみ、解読としての意味をもつものと考えてよい。

ところで、SCIS2000 で発表された竹内らの論文「共通鍵ブロック暗号 RC5, RC6 に対する correlation attack」では、RC5 に対するχ^2攻撃が考察されている。この論文の方法論は Knudsen, Meier らの方式と全く同じであるが、24段の RC5 に対する攻撃に成功している。具体的には2^{20}個に１個の割合の鍵で成立する distinguishing attack と、2^{45}個に１個の割合の鍵で成立する key recovery attack が示されている。いずれも解読に必要な選択平文数は2^{54}である。

自己評価書による RC6 の安全性の評価は、これまでの RC5 の安全性の評価をもとにしているため、竹内らのこの結果は、RC6 の安全性評価の根本にかかわるという見方もできる。いずれにしても RC6 における弱鍵を含めた統計評価研究の余地はまだ大いに存在していると考えられる。

（注）竹内らの論文中では24段のRC5が「完全仕様」と書かれているが，これは初期の完全仕様版であり，その後さまざまな強度評価の結果、現在では32段が推奨仕様になっている（Kaliski and Yin, “On the Security of the RC5 Encryption Algorithm”, TR-602, Sep, 1998 (ftp://ftp.rsasecurity.com/pub/rsalabs/rc5/rc5-report.pdf)。また現在知られている最も良い解読法は Eurocrypt’98で発表された Biryukov と Kushilevitz によるもので、24段のRC5を2^{44}個の選択平文で解読している。

２．９　mod n 攻撃に対する安全性

【自己評価書における記述】

（自己評価書添付資料からの引用）
また一部に存在する攻撃法である、*boomerang* や*mod n* などの暗号解析法はRC6には適用不可能であると思われる。

【考察】

mod n 攻撃は、暗号アルゴリズムの中間データの分布を mod n で見たときに出現する偏りを手がかりに拡大鍵の情報を推定するという手法である。この解読法は算術演算を多く用いているアルゴリズムにおいて有効になるが、算術演算と論理演算がともに利用されているアルゴリズムでは急速にその効果が失われることが知られている。この mod n 攻撃は特殊な暗号アルゴリズムでのみ成立する解読法と考えたほうがよい。

RC6 においては３２ビット加算と３２ビット排他的論理和がともに含まれているため、mod n 攻撃は成立しない。

２．１０　弱鍵の存在について

【自己評価書における記述】

（自己評価書添付資料からの引用）
*弱い鍵*という言葉は広く使われているが、いつも同じ様に使われるとは限らない。最も有名な弱い鍵の例はおそらく、暗号の構造的な性質をついたDESにおけるものであろう。DESにおいては、その存在がセキュリティにあたえる影響は限られている。弱い鍵が存在する他の暗号方式をあげるとすれば、それはIDEAとblowfishであろう。これらの方式では、弱い鍵の存在によって一般的には考えにくい非常に限られた状況の中での暗号文の解読が可能になる。公開されて以来、RC5では弱い鍵の例は報告されていない。構造的な弱点を露呈するものや、限定された解析攻撃を誘引するような例もない。RC6の鍵スケジュールは、重要な点ではすべてRC5のものと等価であるので、RC6においても同様に弱い鍵は存在しないものと期待される。

【考察】

自己評価書では、RC5では弱い鍵の例は報告されていないとあるが、χ^2攻撃の節でも示したように、SCIS2000 で発表された竹内らの論文「共通鍵ブロック暗号 RC5, RC6 に対する correlation attack」では、初期の完全仕様24段の RC5 に対する（広い意味での）弱鍵の存在が、すでに示されている。したがって自己評価書のように RC5 の弱鍵が存在しないことを根拠に RC6 の弱鍵の不存在を主張することは不可能である。

完全仕様の RC6 の弱鍵は見つかっていないが、Knudsen, Meier のχ^2攻撃結果（２５６ビット鍵に対する１７段 RC6 における弱鍵）は、すでに設計者の意図を超えたものであったことにも見られるように、RC6 における弱鍵を含めた統計評価研究の余地はまだ大いに存在していると考えられる。

２．１１　関連鍵攻撃に対する安全性

【自己評価における記述】

（自己評価書からの引用）
RC6は、その鍵スケジュールの構造とデザインのおかげで、*related-key* 攻撃と*slide* 攻撃に対しては特に強力だと考えられる。

（自己評価書添付資料からの引用）
専門家の注目を浴びている攻撃の種類に関連鍵攻撃がある。暗号処理の過程で使われる複数の鍵には、何らかの既知の関連性があり、平文と暗号文の関連性を調べることで、二つの未知の鍵情報を推測できる可能性があるというのがこの攻撃の大前提になっている。 再度同じ議論になるが、RC5の鍵スケジュールはこの種の攻撃に関する研究の対象としても数年間の実績をもち、その中でこの種の攻撃に関する報告は皆無であるので、この種の攻撃はRC6に対しても適用不可能であると思われる。というより、この鍵スケジュールはきわめて複雑であり、またさらに重要なことに、暗号処理の構造と若干矛盾があるように見せるようなデザインが可能なので、実はこの種の攻撃法が開発される可能性は非常に低いということを付け加えておくべきかもしれない。ユーザ鍵の差違を知ることで、暗号処理用の副次鍵の差違を知り、それを利用可能にするような変換法を想像するのは難しい。もちろん、この議論が本当に正しいかどうかは、今後の研究を待って判断されることである。

【考察】

自己評価書のこの記述はおおむね正しいと考えられる。関連鍵攻撃は鍵スケジュール部が非常に簡単な暗号アルゴリズムに対して成立する（あるいは意味をもつ）ものであって、鍵スケジュールが複雑なアルゴリズムに対する関連鍵攻撃は、それが仮に可能であったとしてもきわめて作為的な関連性を解読者が与えられなければならなくなり、解読としての意味が薄れてしまうと考えられる。

RC6 は複雑ではないが、関連鍵攻撃を無効化するに十分なロジックを鍵スケジュール部にもっている。したがって関連鍵攻撃が RC6 で成立するということは考えにくい。

２．１２　スライド攻撃に対する安全性

【自己評価における記述】

（自己評価書からの引用）
RC6は、その鍵スケジュールの構造とデザインのおかげで、*related-key* 攻撃と*slide* 攻撃に対しては特に強力だと考えられる。

【考察】

スライド解読法はデータ暗号化部のラウンドを一段あるいはそれ以上の段数をずらしたものと、もとのアルゴリズムとの間に、拡大鍵を含めた等価なロジックが現れる場合に有効となる解読法であり、一種の関連鍵解読法と考えることができる。この解読法が成立するためには、１段あるいはそれ以上の段数ごとに同じ拡大鍵が周期的に各ラウンド関数に入力されることが必要となる。RC6 の鍵スケジュール部はこのような構造をもっていないので、RC6 にスライド解読法を適用するのは困難と考えられる。

２．１３　タイミング攻撃に対する安全性

【自己評価における記述】

（自己評価書からの引用）
RC6は、タイミング攻撃を防ぐ様に実装することが容易である。最近のプロセサには、回転や乗算の命令の処理時間が一定なものが多い。プロセサのなかには回転やシフトの処理時間が回転の量に比例するものもあるが、通常この場合でも総計算時間をデータに影響されないように配慮することは難しくない（一例として、*t*ビットの回転の場合、*t*ビットの左シフトと、*w*-*t*ビットの右シフトによって行える）。いずれの方法でも、RC6の暗号化/復号化に要する時間はデータの影響を受けず、タイミング攻撃の可能性をなくすことができる。

【考察】

タイミング攻撃とは、暗号鍵などのセキュリティパラメータ値によって暗号の演算時間が異なるような実装が行われた場合に、この演算時間を観測することによってこのセキュリティパラメータの値を逆算する攻撃法である。この攻撃法に対処するためには、セキュリティパラメータの値に依存せずに常に同じ時間で暗号化、復号が完了する実装をする必要がある。

自己評価書に書かれた「RC6は、タイミング攻撃を防ぐ様に実装することが容易である。」という一文は、言い換えれば「RC6は、不用意な実装ではタイミング攻撃で破られる危険性がある。」ということであり、そのように読まれるべきである。実際、RC6のような乗除算やデータの回転シフトを多用するアルゴリズムでは、その入力値によって演算時間が異なる場合があるため、プラットフォームによっては（特にＩＣカード用プロセッサなど）実装時にこの攻撃法に対する注意が特に必要である。

もちろんこの自己評価書に書かれているように、それを防ぐような実装は可能ではある。しかしながらこの防御のために、速度やプログラムサイズを犠牲にしなければならないことにも同時に注意する必要がある。またタイミング攻撃はソフトウエア実装だけでなく、ハードウエアでの実装でも起こりうる。結論としては、RC6は実装時にタイミング攻撃に注意すべきであり、それを避けるために注意深い実装が要求されるということである。

２．１４　電流解析に対する安全性

【自己評価における記述】

（自己評価書からの引用）
他のサイド・チャネル攻撃（たとえば差分指数解析）に対しても事前に同様の注意が払われるであろうが、この場合アルゴリズムのレベルで行うべきか、システムのレベルで行うべきか疑問の残るところではある。

【考察】

電流解析攻撃、差分電流解析攻撃はおもにＩＣカードをターゲットとして、暗号鍵などのセキュリティパラメータに依存して電流消費量が変化することに着目して解読を行うというものである。電流消費量はビットデータの値そのもの（あるいはビットデータが変更されたかどうか）によって変化するということがこの解読法の本質的な点である。この攻撃はＩＣカードの物理特性とアルゴリズムの実装方法に大きく依存するため、自己評価書にあるように暗号アルゴリズムそのものの構造によってあらゆる差分電流解析攻撃を防御するのは困難であると考えるべきであろう。

実際ＡＥＳ候補すべてに対し何らかの差分電流攻撃が成立することが（理論的には）知られている。この攻撃法は現在の暗号研究におけるホットトピックのひとつであり、攻撃と防御の両面から研究が進められている。差分電流解析攻撃の防御方法のうち有力なものとして、FSE2000 で発表された Messerges による “Securing the AES Finalists Against Power Analysis Attacks” があり、ここではＡＥＳ最終選考５候補について、差分電流解析攻撃を防御した実装法が示されている。

しかしながら CHES2000 では、この方法は Rijndael と Serpent 以外のアルゴリズム、すなわち算術演算をもちいた暗号には実は適用できないことが示されており、またおなじ CHES2000 の別論文では、この防御方法そのものも、高次の差分電流解析攻撃には必ずしも有効ではないことが示されているなど、この種の研究はまだ途上であり、いかなる暗号アルゴリズムに対しても差分電流解析攻撃について結論的なことが言える段階ではないと考えられる。

３．　まとめ

ＡＥＳ仕様のＲＣ６については、理論的なものを含め、鍵の全数探索よりも高速であるようないかなる解読法も発見されていない。その意味では、現時点ではＲＣ６は十分安全な暗号アルゴリズムといってよい。しかしながら、暗号理論的な観点からは以下の点に注意すべきである。

（１）　ＲＣ６の安全性評価のいくつかは、これに先立つＲＣ５の安全性評価結果をもとにしており、ＲＣ５の安全性を根拠にＲＣ６の安全性を帰結している。しかしながら、ＲＣ５は鍵によって暗号の強度がばらつくことがしられている。そのひとつの例はχ^2攻撃であり、またＲＣ６に対する現在しられている最も強力な攻撃法はχ^2攻撃である。したがってこの種の統計解析による強度評価結果には今後も注目していく必要がある。

（２）　ＲＣ６を特徴づけるコンポーネントである、乗算とデータ依存回転シフト演算が、タイミング攻撃や差分電流解析に対してどのように作用するか、すなわちこれらの解読法に対する防御実装にどの程度のコストがかかるかは、いまだ研究途上であるが、その情報はＲＣ６を特定の環境で利用する上で重要になる可能性がある。

以上

付録　ＲＣ６の仕様　（技術仕様書の抜粋）

RC6は完全にパラメータ制御可能な暗号アルゴリズムの一種である。RC6の各バージョンは、RC6-*w/r/b* と書くことでさらに正確に表現できる。ここで、ワード長は*w*ビット、暗号化には負でない値のラウンド数ｒ が使われ、またb は暗号鍵の長さをあらわすバイト数である。ブロック長が128ビットの場合は、*w*=32および*r*=20が推奨値で、単にRC6といった場合はこのバージョンを指す。文章中でこれ以外の特別な*w* あるいは*r* の値を使う場合、RC6-*w/r* の様にパラメータを指定する。64ビットのブロック長に対しては*w*=16ビットと指定することになる。鍵長は0から256バイトの範囲の値をとることができるが、16バイト、24バイト、そして32バイトの鍵長のRC6のバージョンが最もよく使われている。

RC6-*w/r/b* では、すべての類型においても同様に、4個の*w*ビット長ワードの単位に対する、以下の6個の基本演算がおこなわれる。lg *w* は、2を底とする*w* の対数をあらわす。

$a + b$	整数加算（モジュロ2^w）
$a - b$	整数減算（モジュロ2^w）
$a \oplus b$	wビット長のワードのビット毎の排他的論理和
$a \times b$	整数乗算（モジュロ2^w）
$a<<<b$	wビット長のワード*a*の左回転。回転量は*b* の最下位 lg *w* ビットで与えられる。
$a>>>b$	wビット長のワード*a*の右回転、回転量は*b* の最下位 lg *w*ビットで与えられる。

RC6の記述中に使われる“ ラウンド ”という言葉は、通常のDES式ラウンド数の発想にほぼ近い。つまり、データの一方の半分を他方の半分で更新し、次にそれらを入れ替える操作をいう。

【鍵スケジュール】

RC6-*w/r/b* の鍵スケジュールは、RC5-*w/r/b* の鍵スケジュールと実質的に等価である。事実上唯一の違いは、RC6-*w/r/b* ではより多くのワードが生成され、それらが暗号復号双方で使われる配列S[0,...,2*r*+3]に配置されることである。RC5鍵スケジュールと同様のものを採用した理由は、これには既に6年にわたる広範囲の研究実績が存在しているからである。

図1に、RC6の鍵スケジュールの擬似コードを記す。ユーザは、0≦*b*≦256である*b*バイトの鍵を与える。最終的にはこの鍵を使って2*r*+4ワード（各々*w*ビット）の拡張鍵を生成する。まず、鍵の長さがゼロではない整数で表されるワード数と等しくなるまでゼロのバイトを追加する。これらの鍵の各バイトはリトル・エンディアン方式によって*c*個の*w*ビット長の配列*L*[0],...,*L*[*c*-1]に前もって配置される。このようにして、鍵の第1バイトは*L*[0]の最下位

バイトに置かれ、以下同様に進み、最後に$L[c-1]$には必要ならば上位のバイトにゼロが埋められる（ただしb=0の場合は、c=1および$L[0]$=0となる。）。生成される1ワードwビットの追加ラウンド鍵のワード数は$2r+4$であり、これらは配列$S[0,...,2r+3]$に配置される。（鍵の長さが異なっていても、ゼロを埋めることで同じ配列$L[0],...,L[c-1]$ができる場合は、鍵スケジュールは同じものになる。といっても一般的には鍵が異れば、鍵スケジュールも異なる。）

定数P_{32}=B7E15163とQ_{32}=9E3779B9（16進数）は、RC5の場合と同じく“魔法の定数”である。P_{32}の値は、e-2の二項展開から導かれる。eは自然対数の底である。Q_{32}の値は、Φ-1の二項展開から導かれる。Φは黄金比である。他のワード長のRC6に対しても同様に、P64およびその他に対するRC5の定義を適用することが可能である。これらの値は幾分任意性を持つものであり、他の値を使って、“あつらえ”のつまり独自バージョンのRC6をつくることも可能である。

【暗号化と復号化】

RC6では、wビットのレジスタA,B,C,Dが使われ、最初これらには入力された平文が置かれ、暗号化の最後の段階では出力暗号文が置かれる。平分あるいは暗号文の第1バイトは、Aの最下位のバイトに置かれる。平文あるいは暗号文の最後のバイトは、Dの最上位バイトに置かれる。ここで、$(A,B,C,D)=(B,C,D,A)$という式は、式の右側のレジスタの値を左側のレジスタに同時にまとめて代入することを意味することとする。図2に、RC6による暗号および復号化の擬似コードを示し、RC6による暗号化を図式で表したものを図3に示す。RC6による暗号化のテストベクトルも示す。

図1. RC6-*w*/*r*/*b*鍵スケジュールの擬似コード

RC6-*w*/*r*/*b*の鍵スケジュール

入力： ワード数cの配列 $L[0,\cdots,c-1]$に前もって配置された
bバイトのユーザ提供鍵。
ラウンド数r。

出力： wビット長ラウンド鍵の配列 $S[0,\ldots,2r+3]$。

手順：

$S[0] = P_w$

for $i = 1$ **to** $2r + 3$ **do**
$S[i] = S[i-1] + Q_w$

$A = B = i = j = 0$

$v = 3 \times \max\{c, 2r + 4\}$
for $s = 1$ **to** v **do**
{
$A = S[i] = (S[i] + A + B) \lll 3$
$B = L[j] = (L[j] + A + B) \lll (A + B)$
$i = (i + 1) \bmod (2r + 4)$
$j = (j + 1) \bmod c$
}

図2RC6-*w*/*r*/*b***による暗号化および復号化の擬似コード**

RC6-*w*/*r*/*b*による暗号化

入力： 4個のwビット長の入力レジスタA,B,C,Dに配置された平文。
ラウンド数r。
wビット長ラウンド鍵の配列 $S[0,...,2r+3]$。

出力： A,B,C,Dに配置された暗号文。

手順：
$B = B + S[0]$
$D = D + S[1]$
for $i = 1$ **to** r **do**
{
$t = (B \times (2B + 1)) \lll \lg w$
$u = (D \times (2D + 1)) \lll \lg w$
$A = ((A \oplus t) \lll u) + S[2i]$
$C = ((C \oplus u) \lll t) + S[2i + 1]$
$(A, B, C, D) = (B, C, D, A)$
}
$A = A + S[2r + 2]$
$C = C + S[2r + 3]$

RC6-*w*/*r*/*b*による復号化

入力： 4個のwビット長の入力レジスタA,B,C,Dに配置された暗号文。
ラウンド数r。
wビット長ラウンド鍵の
配列 $S[0,... ,2r+3]$。

出力： A,B,C,Dに配置された平文

手順：
$C = C - S[2r + 3]$
$A = A - S[2r + 2]$
for $i = r$ **downto** 1 **do**
{
$(A, B, C, D) = (D, A, B, C)$
$u = (D \times (2D + 1)) \lll \lg w$
$t = (B \times (2B + 1)) \lll \lg w$
$C = ((C - S[2i + 1]) \ggg t) \oplus u$
$A = ((A - S[2i]) \ggg u) \oplus t$
}
$D = D - S[1]$
$B = B - S[0]$

図3 RC6-*w*/*r*/*b***による暗号化。ここで** *f*(*x*)= *x**(2*x*+l)**とする。**