付録G:RC6の耐高階差分攻撃強度評価

2001年1月29日

Contents

1 はじめに

RC6 は、1998 年に Rivest らによって提案されたブロック暗号であり[1]、AES-5finalists の一つであった。RC6 ではその前身にあたる RC5 で用いられている data-dependent rotation、Round 鍵の整数加算などの演算を安全性確保の主軸とし、これらの演算をより効果的に用いるべく様々な改良がなされている。提案者らは Round 関数に代数的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、安全際の向上、段数の節約、暗号化の効率化を図っていることを主張している。

提案者らは、線形攻撃と差分攻撃に関する安全性については十分な検討を行っているが、高階差分攻撃に対する強度については明らかにしていない。本稿は RC6 の耐高階差分攻撃強度評価を目的とする。

RC6 では暗号化アルゴリズムにおける最小演算単位が 32-bit 幅であり、これより細かい演算単位の暗号モジュールに分解して解析することは不可能である。また、その最小単位のモジュール、例えば F 関数にも複雑度の低い演算を用い、それらを Round 関数内で巧妙に組み合わせることによって暗号としての安全性を高めている。このため、各モジュールに対する調査結果から、暗号アルゴリズムの強度を推し量ることは非常に困難であり、各調査結果を攻撃に結びつけることも困難である。従って、本稿では諸調査対象を RC6 の Round 関数とし、複数段通過後の出力について重点的に調査を行い、現実的な高階差分攻撃耐性強度評価とした。

本稿の構成は以下の通りである。まず 2 章で RC6 の暗号化アルゴリズム、3 章で高階差分の定義について説明を行い、4 章で形式的な代数次数を見積もる。5 章及び 6 章で攻撃に効果的な差分について解析を行い、4-Round RC6 は 2 階差分攻撃を用いて攻撃可能であることを示す。7 章で、その他 RC6 について行った諸調査の結果を示す。

2 RC6

RC6 は、Rivest らによって提案されたブロック暗号である。ワード長 w、段数 r、鍵サイズ k を使用者が決定でき、一般に RC6-w/r/k と表す。RC-6-32/20/k が AES ヴァージョンである。RC6 の構造を Figure 1 に示す。図中の演算記号は以下の通りである。

$a+b$:modulo 2^w の加算
$a \oplus b$:ビット毎の排他的論理和
$a \times b$:modulo 2^w の代数掛算
$a<<<b$:b の下位 $\log_2 w$ 回だけ a を左巡回シフト
$a>>>b$:b の下位 $\log_2 w$ 回だけ a を右巡回シフト

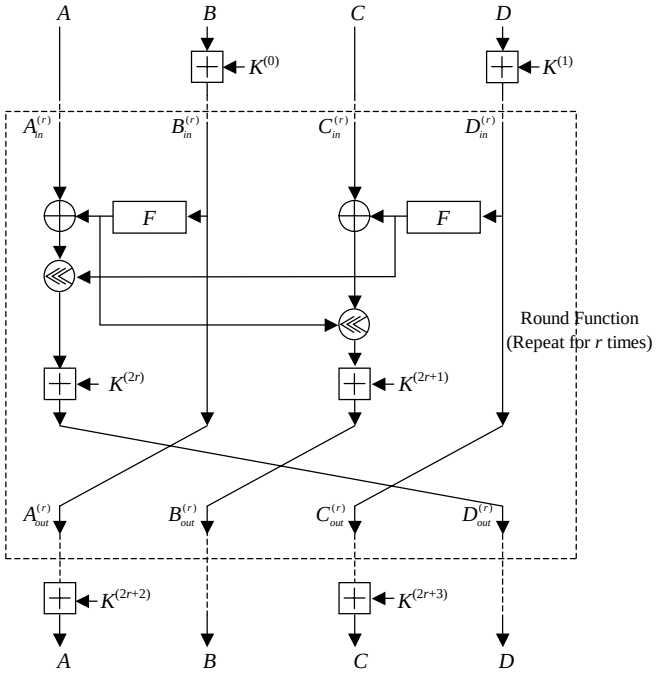

Figure 1: RC6

RC6 は、(w×4)-bit の入力平文を 4 つの w-bit ブロックに当分割して格納し、これらのブロック間で諸演算を行う変形 Feistel 型構造を持つ。本稿ではこの w-bit ブロックを sub-block と呼ぶことにする。

また、F 関数及び f 関数を以下のように定義する。

$$F(x) = (2x^2 + x) <<< \log_2 w \qquad (2.1)$$

$$= f(x) <<< \log_2 w \qquad (2.2)$$

Round 関数の安全性を評価するという立場から、図1における鍵 $K^{(0)}$, $K^{(1)}$, $K^{(2r+1)}$及び $K^{(2r+3)}$は以下では無視する。また、連続した r 個の Round 関数を r-Round 関数と表す。

3 高階差分

$E(X;K)$を、入力 $X \in \mathrm{GF}(2)^n$ と鍵 $K \in \mathrm{GF}(2)^s$ から、$Y \in \mathrm{GF}(2)^m$ を出力する、GF(2)上の暗号化関数とする。

$$Y = E(X;K) \qquad (3.1)$$

$(a_1,a_2,\cdots,a_i)$を $\mathrm{GF}(2)^n$ 上で 1 次独立な i 個のベクトルとする。これらによって張られる、$\mathrm{GF}(2)^i$ の部分空間を $V^{(i)}$で表し、入力差分と呼ぶ。関数 $E(X;K)$の X に関する i 階差分 $\Delta^{(i)}E(X;K)$は以下の式で定義される。

$$\Delta^{(i)}E(X;K) = \bigoplus_{\mathsf{A} \in V^{(i)}} E(X \oplus \mathsf{A};K) \qquad (3.2)$$

関数 $E(X;K)$の X に関する次数が N であるならば、X と K に依存せず、次式が成立する。

$$\Delta^{(N+1)}E(X \oplus \mathsf{A};K) = 0 \qquad (3.3)$$

4 Round 関数の代数次数

RC6 の Round 関数における形式的な代数次数を見積もる。Round 関数は F 関数、Data-dependent Rotation、鍵の整数加算で構成される。それぞれの演算における代数次数増加を順に見積もる。まずワード長 w＝8-bit の場合の Round 関数について考え、これを基にワード長 w-bit 版へ拡張する。

4.1 F 関数の代数次数

F 関数は式(2.1)で与えられる演算であり、全単写関数である。w＝8-bit の縮小型 RC6 における F 関数のブール展開式を算出した。その結果を表 1 に示す。この表から縮小型 RC6 の F 関数の代数次数は形式的に 7 次であり、一般にワード長 w-bit のとき$(w-1)$次である。

F 関数では、$\log_2 w$ ビットの巡回シフトにより、下位ビットの代数次数を高くしている。これにより、次に示す Data-dependent Rotation において、シフト回数を決定するビットの次数が増加し、両演算の組み合わせにより次数増加速度を増加させている。

		次数
y_7	$x_0x_1x_2x_3+x_0x_1x_3+x_0x_1+x_0x_2+x_1x_3+x_4$	4
y_6	$x_0x_1x_2+x_0x_1+x_1+x_3$	3
y_5	$x_0x_1+x_2$	2
y_4	x_0+x_1	1
y_3	x_0	1
y_2	$x_0x_1x_2x_3x_4x_5x_6+x_0x_1x_2x_3x_4x_5+x_0x_1x_3x_4x_5x_6$ $+x_0x_1x_2x_3x_4+x_0x_1x_2x_4x_6+x_0x_1x_2x_5x_6$ $+x_0x_1x_3x_4x_6+x_0x_1x_4x_5x_6+x_0x_2x_3x_4x_5$ $+x_0x_2x_3x_4x_6+x_0x_2x_4x_5x_6+x_1x_2x_3x_4x_5$ $+x_1x_3x_4x_5x_6+x_0x_1x_2x_5+x_0x_1x_3x_4+x_0x_1x_3x_5$ $+x_0x_1x_4x_5+x_0x_2x_3x_6+x_0x_2x_4x_6+x_0x_3x_4x_5$ $+x_0x_3x_5x_6+x_1x_2x_5x_6+x_1x_3x_4x_5+x_0x_2x_4+x_0x_4x_6$ $+x_1x_2x_3+x_1x_2x_6+x_1x_3x_6+x_2x_5x_6+x_0x_5$ $+x_1x_4+x_2x_3+x_3+x_7$	7
y_1	$x_0x_1x_2x_3x_4x_5+x_0x_1x_3x_4x_5+x_0x_1x_2x_4$ $+x_0x_1x_2x_5+x_0x_1x_3x_4+x_0x_1x_4x_5+x_0x_2x_3x_4$ $+x_0x_2x_4x_5+x_1x_3x_4x_5+x_0x_2x_3+x_0x_2x_4+x_0x_3x_5$ $+x_1x_2x_5+x_0x_4+x_1x_2+x_1x_3+x_2x_5+x_6$	6
y_0	$x_0x_1x_2x_3x_4+x_0x_1x_3x_4+x_0x_1x_2+x_0x_1x_4$ $+x_0x_2x_4+x_1x_3x_4+x_0x_3+x_1x_2+x_2+x_5$	5

Table 1 F 関数出力ブール展開式

4.2 Data-dependent Rotation の代数次数

縮小型 RC6 における、巡回シフトの回数を $V=(v_2,v_1,v_0)$とし、$Y=(y_7,\cdots,y_0)$を V によって巡回シフトしたものを $Y'=(y'_7,y'_6,\cdots,y'_0)$とするならば、例えば y'_0 は以下のように計算することができる。

$$y'_0 = y_0(v_2+1)(v_1+1)(v_0+1) \oplus y_1v_2v_1v_0 \\ \oplus\cdots\cdots \oplus y_7(v_2+1)(v_1+1)v_0 \qquad (4.1)$$

これより、縮小型RC6におけるData-dependent Rotationの代数次数は形式的に4次であることがわかる。一般に、ワード長 w-bit の場合、巡回シフト回数は $\log_2 w$ ビットで与えられ、このときの代数次数は形式的に $(\log_2 w+1)$ 次である。

4.3 鍵の整数加算の代数次数

入力を $X=(x_7,\cdots,x_0)$、$K=(k_7,\cdots,k_0)$、出力を $Y=(y_7,\cdots,x_0)$とすれば

$$y_i = x_i \oplus k_i \oplus d_i \\ d_i = x_{i-1}k_{i-1} \oplus x_{i-1}d_{i-1} \oplus k_{i-1}d_{i-1} \qquad (4.2)$$

と表すことができる。ただし d_i は桁の繰り上がり項で $d_0=0$、$(i=0\sim7)$であり、k_i は定数である。整数加算の代数次数の最大値は y_7 において得られ、その値は 7 次である。よって、整数加算の代数次数は形式的に 7 次である。一般に、ワード長 w-bit の場合、整数加算の代数次数は形式的に $(w-1)$ 次である。

以上の結果から、Round 関数の形式的な代数次数は 7×4×7 次以下である。

ワード長 w-bit のとき、形式的な代数次数は$(w-1)\times(\log_2 w+1)\times(w-1)$である。

各段数通過時において、入力変数ビット数に対する次数の飽和を考慮するならば、各 Round 通過時のブール代数次数は Table 2 に示す通りである。

	ブール代数次数			
出力 sub-block 位置	*A*	*B*	*C*	*D*
1st-Round	1	64	1	64
2nd-Round	64	128	64	128
3rd-Round	128	128	128	128

Table 2 各 Round 通過時のブール代数次数

5 効果的な差分

高階差分攻撃において、階数は、攻撃に必要な最小の平文数や計算量に影響を与える。そのため、攻撃に必要な最小の階数を見積もる必要がある。階数は差分の選択の仕方に依存するため、攻撃に用いる差分の選択は重要である。このセクションでは効果的な平文の選択について述べる。

RC6 はワード長可変暗号であるが、ここでは解析対象を AES 版(*w*=32-bit)に限定する。

本章では sub-block ごとに与える差分の中で効果的なものを探索する。まず差分を与える sub-block を 4 つの sub-block の中から選択する。RC6 の Round 関数のもつ対称性から、入力差分を与える sub-block を *A* と *B* の 2 種類に限定する。また、sub-block *B* は初段で *F* 関数を通過するのに対して sub-block *A* は 2 段目まで通過しない。この理由から差分を与える sub-block を sub-block *A* に限定し、以下の議論を行う。

5.1 sub-block *A* に対する 32 階差分攻撃 -SQUARE Attack-

このセクションでは sub-block *A* に 32 階差分を与える、32 階差分攻撃について述べる。入力平文のうち、sub-block *A* 以外、すなわち *B, C, D* を 0 に固定した場合の、3 段目出力における sub-block *C*までの処理を模式的に表したものが Figure 2 である。但し sub-block X($X = A, B, C, D$)の *r*-round 目の入力を $X_{in}^{(r)}$、出力を $X_{out}^{(r)}$と表すものとする。

$A_{in}^{(1)}$ が 2^{32} 通りの値を重複無く 1 度ずつ取るならば、図中の X も 2^{32} 通りの値を重複無く 1 度ずつ取る。このように、*N*-bit の値に対して 2^N 種類の値を重複無く 1 度ずつ取ることを、以後 2^N 通り廻ると呼ぶ。*F* 関数は全単写関数であるから、*Y* は 2^{32} 通り(全通り)廻る。*Y* が全通りの値を取ることは、シフト回数として 2^5 通りの値が、正確に 2^{27} 回ずつ出現することを示す。このため、固定値をシフトした *Z* も、2^5 通りの値が 2^{27} 回ずつ出現することになる。偶数回ずつ出現した値について、それら全ての排他的論理和を取れば 0 であるから *Z* の位置での高階差分値は 0 である。以後、この状態を balance していると呼ぶ。

鍵の整数加算は全単写関数であるため、この balance は鍵加算後も維持され、図中 *H* の位置での 32 階差分値は 0 である。従って 3 段目出力において sub-block *C* の 32 階差分値は常に 0 である。

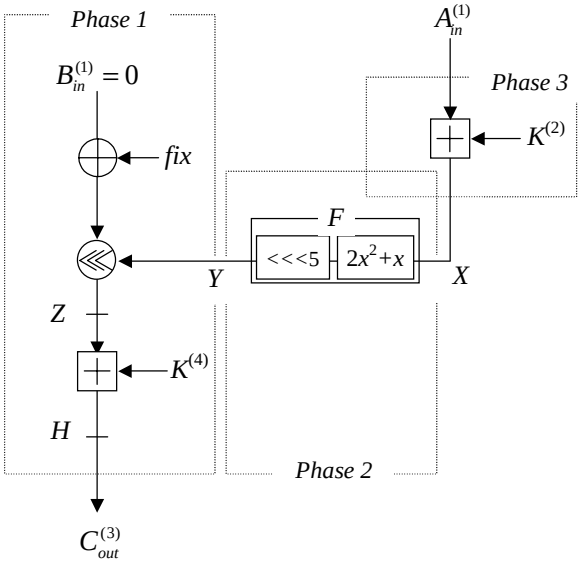

Figure 2: 2-Round における $C_{out}^{(3)}$ の導出

また、この 32 階差分を取ったとき、3 段目出力時、sub-block *A* は 2^{32} 通りの値が一度ずつ出現するため、この 32 階差分値も常に 0 である。計算機実験により、sub-block *A* 及び *C* の 32 階差分値が実際に 0 となることを確認した。

従って、32 階差分を用いて 4 段目の拡大鍵について攻撃方程式を立て、解くことにより、4-round RC6 を攻撃することが可能である。このときの必要平文数は $2^{32}\times 2$ であり、必要計算量はおおよそ $2^{32}\times 2^{32}$ である。

5.2 sub-block *A* に対する 6 階差分攻撃

前節で 4-Round RC6 は 32 階差分攻撃で攻撃可能であることを示した。このセクションでは、4-Round RC6 に対して、より少ない階数での攻撃について考察を行う。Figure 2 を Phase 1 から 3 に分割し、順に検討する。

5.2.1 Phase 1

data-dependent rotation のシフト回数は *Y* の下位 5-bit により決定される。ここで、シフト対象が固定値である為、*Y* において、最下位 5-bit を除いた 27-bit 中 1-bit 以上が廻っていれば *Z* は balance され、従って $K^{(4)}$ 加算後の *H* も balance される。

5.2.2 Phase 2

F 関数は $f(x)=2x^2+x \pmod{2^{32}}$という演算と、5-bit の左巡回シフトから構成される。ここで、$f(x)$という演算上、入力 *X* の最上位から連続した *m*-bit $(0<m\leqq 16)$のみ変数に取るならば、出力 $y=f(x)$の上位 *m*-bit は 2^m 通りの値を一度ずつ取る。

y を 5-bit 左巡回シフトしたものが *F* 関数の出力となる為、Phase 1 と合わせて考えれば、*X* の最上位 6-bit を変数に選ぶことにより、*Y* の最下位 5-bit 及び、最上位 1-bit の計 6-bit を 2^6 通り廻すことが出来る。これは *Z* において 2^5 通りの値がそれぞれ 2 度ずつ出現することを意味し、前述の通り、これは balance される。

5.2.3 Phase 3

$A_{in}^{(1)}$ の最上位から連続したビットのみ変数に選べば、round 鍵の整数加算はその変数ビットに関して全単写関数である。従って $A_{in}^{(1)}$ の最上位 6-bit を変数に取ることにより、*X* の最上位 6-bit を 2^6 通り廻すことができる。

上記 Phase 1 から Phase 3 までの解析により、$A_{in}^{(1)}$ の最上位 6-bit からなる 6 階差分を与えることにより、3 段目出力における sub-block *C* の高階差分値を 0 にすることが出来る。これを 10,000 通りの拡大鍵について、計算機実験により確認した。

従って、この差分を適用し、攻撃方程式(6.3)(6 章に後述)を解くことにより、4 段目に用いられている 2 つの 32-bit 拡大鍵の一方を求めることが可能である。このときの必要平文数は $2^6\times 2=128$ であり、攻撃方程式に含まれる拡大鍵について全数探索をするならば、このときの計算量はおおよそ $2^6\times 2^{32}$ の round 関数計算である。

5.3 計算機による探索　　-2 階差分攻撃-

5.2 節の議論により、4-Round RC6 は 6 階差分攻撃により攻撃が可能であることがわかった。この同じ段数に対して、更に低い階数による攻撃が可能であるかを調査する為に、以下の計算機実験を行った。入力差分を与える sub-block は上記と同じ sub-block A とし、この sub-block の入力 32-bit 中、任意の n-bit（$1 \leqq n \leqq 5$）を変数とした n 階差分全てについて、3 段目出力の n 階差分値を求めた。これを 1000 通りのランダムな拡大鍵について行い、これら全ての鍵に対して n 階差分値の幾つかのビットが常に 0 又は 1 となる入力差分の有無を調査した。

結果として、sub-block A の n 階差分値は以下の特性を持つことを確認した。
sub-block A の最上位（$n-1$）-bit（但し $2 \leqq n \leqq 5$）及び、$32-(n-1)$ビットから任意の 1 ビットを変数に取る n 階差分を与えたとき、3 段目出力において、sub-block A の n 階差分値の最下位 n-bit は常に 0 となる。また、この n 階差分値のうち、鍵に依存せず 0 もしくは 1 となるものは、上記最下位 n-bit のみである。

攻撃に必要な平文数及び、計算量の観点から考え、最も効果的であるものは 2 階差分である。つまり、sub-block A の最上位 1-bit 及び任意箇所の 1-bit による 2 階差分を用いたとき、3 段目出力における sub-block A の 2 階差分値の最下位 2-bit は常に 0 とすることができる。これは以下に示す理由によるものと考えられる。

最上位の変数を x_m、もう一方の変数を x_n とする。この 2 変数の 2 階差分を $\triangle X$ とする。sub-block A に入力差分 $\triangle X$ を与えるとし、sub-block B, C, D の入力を 0 とする。このとき、3 段目出力時、sub-block A は次式により与えられる。

$$A = \{F\,(\triangle X + K^{(2)}) <<< K^{(3)}\} + K^{(5)} \qquad (5.1)$$

$\triangle X + K^{(2)} = X$ とする。この整数加算において x_m が最上位にあるとき、x_m、x_n を共に含んだ桁上がりは存在しない。従って、X の代数次数は 1 次であり、かつ、x_m は最上位のみに保持されたままである。

ここで F 関数について考える。F 関数は $2X^2 + X$ を 5-bit 左巡回シフトしたものである。ここで、$2X^2$ という項は x_m を含まない。この項と X が整数加算されるが、x_m が最上位のみにある限り x_m を含む桁上がりは起きない。したがって、この 2 項の整数加算においても 2 次項は出力されない。これが 5-bit 左巡回シフトしても、代数次数は 1 次のままである。この出力が、Round 鍵 $K^{(3)}$ によってシフトされ、変数位置は特定できなくなるが、$K^{(5)}$ の整数加算では、代数次数 1 次の変数と定数の整数加算であるから、最下位 2-bit は常に 1 次のままである。従って、最下位 2-bit の 2 階差分値は常に 0 である。

この 2 階差分を適用し、攻撃方程式(6.3)(6 章に後述)を解くことにより、4 段目に用いられている 2 つの 32-bit 拡大鍵の一方を求めることが可能である。このときの必要平文数は $2^2 \times 17 = 68$ であり、攻撃方程式に含まれる拡大鍵について全数探索をするならば、このときの計算量はおおよそ $2^2 \times 2^{32} = 2^{34}$ の Round 関数計算である。これは前節 5.2 で述べた 6 階差分による攻撃より必要平文数、計算量共に少なく、これまでの解析において、最も効果的な差分である。

RC6 は 1 段当たり 64 ビットの拡大鍵を使う。さらに 3 段の段関数を総当たりして前述の攻撃を適用するならば、7 段が攻撃可能であり、推定する拡大鍵ビット数は 224 ビット、必要平文組数は $2^2 \left\lfloor \frac{224}{2} \right\rfloor = 448 \approx 2^9$、計算量は 2^{226} の段関数計算である。

6 解読実験

6.1 攻撃方程式

RC6 の r 段目の Round 関数処理において、$B_{out}^{(r)}$ と $D_{out}^{(r)}$ の 2 つの sub-block が暗号化変換されて出力される。従って、この 2 つの sub-block の処理に対してそれぞれ攻撃方程式を立てることが出来るが、ここでは $B_{out}^{(r)}$ の処理について攻撃方程式を導出する。Round 関数の対称性から、次式(6.1)～(6.3)に含まれる変数を、BをDにCをAにそれぞれ置き換えることによって、$D_{out}^{(r)}$ の処理に関する攻撃方程式となる。r-Round 関数の最終段のうち、$B_{out}^{(r)}$ の導出に関する処理を Figure 3 に示す。図中、$C_{in}^{(r)}(X), C_{out}^{(r)}(X), A_{out}^{(r)}(X)$ 及び $B_{out}^{(r)}(X)$ は平文 X に対応する値である。$C_{in}^{(r)}(X)$ は、既知の暗号文と鍵を用いて、以下のように逆算できる。

$$C_{in}^{(r)}(X) = \{(B_{out}^{(r)}(X) - K^{(2r+1)}) >>> F((A_{out}^{(r)}(X))\} \oplus F(C_{out}^{(r)}(X)) \qquad (6.1)$$

ところで、$C_{in}^{(r)}(X)$ の X に関する次数が N であるならば、式(3.3)より以下が成り立つ。

$$\Delta^{(N+1)} C_{in}^{(r)}(X) = 0 \qquad (6.2)$$

式(3.4)、式(3.5)から、以下に示す攻撃方程式を導くことが出来る。

$$\bigoplus_{A \in V^{(N+1)}} [\{(B_{out}^{(r)}(X \oplus A) - K^{(2r+1)}) >>> F((A_{out}^{(r)}(X \oplus A)\} \oplus F(C_{out}^{(r)}(X \oplus A))] = 0 \qquad (6.3)$$

攻撃方程式の等号は、鍵 $K^{(2r+1)}$の値が正しいときは成立する。ワード長 w[bit]の場合、鍵の全数探索によって、この攻撃方程式を解くならば、2^{N+1} 個の選択平文と $2^{N+1} \times 2^w = 2^{N+1+w}$ の計算量が必要である。

6.2 解読実験

5.3 節で述べた選択平文のうち、最上位から連続した 2-bit のみを変数とする 2 階差分を用いて解読実験を行った。式(6.3)に含まれる 32-bit 鍵については全数探索を行った。必要平文数は、上記 2 階差分を張る 2^2 通りの平文×17 組であり、必要計算量は約 $2^2 \times 2^{32}$ である。Pentium III 500MHz を用いて、約 30 分で 4 段目拡大鍵 $K^{(8)}$をただ一つに定めることに成功した。鍵の探索過程の工夫は今後の課題である。

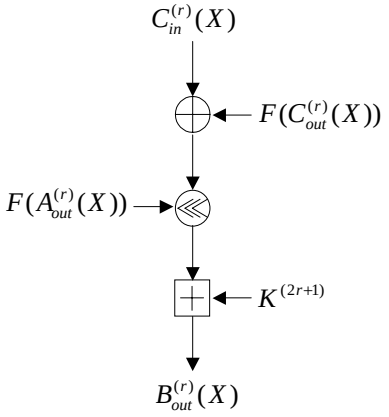

Figure 3 : r-Round 関数の最終段の一部

7 その他諸調査

本章では前章までの解析と独立に、RC6 に関して行った調査等について述べる。

7.1 縮小版 RC6 に対する効果的な差分探索

RC6 では鍵が整数加算されている。このように鍵が使用された場合、Round 関数の出力の代数次数が N 次以上と見積もられても、その N 階差分値を 0 とする拡大鍵が存在することが報告されている[7]。本稿ではこのような鍵を弱鍵と呼ぶ。差分の選び方に対する弱鍵の含まれる割合により効果的な差分探索を行った[9],[10]。

ここでは、ワード長 w＝8[bit]の縮小版 RC6 に対して行った効果的な差分の探索実験について述べる。与える差分を sub-block 単位のもののみに限定し、1 階～8 階差分について計算機探索を行った。RC6 の構造の対称性から、差分を与える sub-block を A と B に限定する。

7.1.1 計算機実験

8 ビットの sub-block に対して、n 階差分を用いた攻撃を適用した場合、${}_8C_n$ 通りの差分取り方が存在する。本研究では連続した n ビットによる n 階差分を考える。

$\Delta_j^{(n)} \in GF(2)^8$ を以下に示すベクトルとする。

$$\Delta_j^{(n)} = (0,0,\cdots,0,a_{N-1},\cdots,a_0) <<< j \qquad (7.1)$$

ここで j（＝0～7）は巡回シフト回数を表す。このベクトルを平文 sub-block M（＝A, B）に与え、差分を $\Delta_{Mj}^{(n)}$ と表す。

r-Round 関数の場合、$2^8 \times 2 \times r$ 通りの鍵を取り得る。ここでは、ランダムに 1 万通りの鍵を選び、計算機実験により弱鍵が含まれる割合を調べた。入力する平文 X を

$$X = P_f \oplus \Delta_{Mj}^{(n)}$$
$$\text{（ただし } P_f \in \mathrm{GF}(2)^8 \text{ は all-zero）} \qquad (7.2)$$

とし、弱鍵が含まれる割合を、3 段目の出力について sub-block ごとに高階差分値を計算し、その値が 0 となる確率で見積もった。ただし、開始段を 1 段目とし、第 r 段目における sub-block M の入力を $M_{in}^{(r)}(X)$、出力を $M_{out}^{(r)}(X)$ と表記する。

7.1.2 結果

実験結果を Table 2 に示す。紙面の都合上、1, 2, 4, 8 階差分値に付いて、確率が 0.50 以上となった差分についてのみ示す。実験結果から、w＝8[bit]の縮小版 round 関数において sub-block A に 8 階差分を適用した場合、3 段目出力において sub-block A 及び C の高階差分値は確率1で 0 となる。この 8 階差分は、w=32-bit 版 RC6 における 32 階差分に相当し、5.1 節で述べた理由によるものである。また、sub-block A に対してある種の 4 階差分を入力することにより、3 段目出力時の sub-block C の 4 階差分値を 0 とすることが出来るが、この理由も 5.2 節で述べた balance の議論に帰結することが出来る。すなわち、シフト回数を決定する最上位 3-bit 及び他の数ビットが廻っている為に出力が balance された為である。

また、これらの差分を用いたときの 4 段目以降の出力計算機実験により見積もった。sub-block A に対し 8 階差分を適用した場合、5 段目出力において、sub-block A の 4 階差分値が確率 0.6%で、8 階差分値が確

確率	$A_{out}^{(3)}$	$B_{out}^{(3)}$	$C_{out}^{(3)}$	$D_{out}^{(3)}$
$P=1.00$	$A_{A0}^{(8)}, \sim, A_{A7}^{(8)}$ $A_{B0}^{(8)}, \sim, A_{B7}^{(8)}$	-	$A_{A4}^{(4)}, \sim, A_{A7}^{(4)}$ $A_{A0}^{(8)}, \sim, A_{A7}^{(8)}$	-
$0.80 \leqq P < 1.00$	-	-	-	-
$0.70 \leqq P < 0.80$	$A_{A4}^{(4)} : 0.745$ $A_{B3}^{(2)} : 0.714$	-		-
$0.60 \leqq P < 0.70$	$A_{A5}^{(4)} : 0.692$	-		-
$0.50 \leqq P < 0.60$	$A_{A6}^{(2)} : 0.582$ $A_{A6}^{(4)} : 0.560$	-		-

Table 2: 3 段出力時において、各 sub-block の高階差分値が 0 となる確率 P に対応する差分

率 0.6%で 0 となる。ランダムな差分の取り方では、4 階差分値もしくは 8 階差分値が 0 となる確率は共に約 0.4%であるから、これらは十分効果的な選択法であるといえる。

7.2 Round 関数のブール展開式の各次数の項数

RC6 の Round 関数のブール展開式に含まれる項数を、次数別に見積もった。計算量の都合により、sub-block ごとに差分を与えた場合について、1 次項から 6 次項までの項数のみ算出した。これを Table 4 に示す。RC6 の Round 関数において、処理が行われて出力されるのは sub-block *D* 及び sub-block *B* である。ここでは sub-block *D* についてのみ調査を行うが、Round 関数の対称性から sub-block *B* についても同様の結果が得られる。

sub-block *D* に関する処理を模式的に表したものが Figure 4 である。この処理は sub-block *A*、*B*、*D* を入力し、sub-block *D* を出力する関数と見ることができる。前述の様に、sub-block ごとに差分を与える場合のみ調査するため、差分を与える sub-block 以外の 2 つの sub-block への入力を 0 とする。このとき、Figure 4 から判るように、sub-block *A* に差分を与えた場合、出力 sub-block *D* は入力 sub-block *A* そのものであり、また、sub-block *D* に差分を与えた場合、出力 sub-block *D* は常に 0 である。従って、sub-block *B* にのみ差分を与える調査を行うが、これは *F* 関数に対する調査に他ならない。本稿の 1 章で述べたように、RC6 の *F* 関数は、単独に用いて攪拌性を保証するという位置付けではない。このため、本調査データについて他の詳細強度評価対象暗号と比較することは公平さを欠くと思われる。

次数	平均	最大値	最小値	期待値
1 次	15.000	2	1	16
2 次	7.938	16	0	248
3 次	11.188	28	0	2480
4 次	80.719	221	0	17980
5 次	498.719	1414	0	100688
6 次	3287.063	11194	0	453096

Table 4 Round 関数のブール展開式の各次数の項数

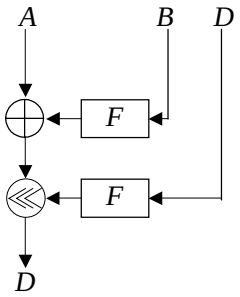

Figure 4: Round 関数の一部

参考文献

[1] R Rivest, M. Robshaw, R. Sidney and Y. Yin: “The RC6 Block Cipher vl.1 ”, (1998-08), http://www.rsa.com/rsalabs/aes

[2] X. Lai: “Higher Order Derivatives and Differential Cryptanalysis” Communications and Cryptography, pp. 227-233, Kluwer Academic Publishers 1994.

[3] T. Jacobsen, L. R. Knudsen: “The Interpolation Attack on Block Cipher”, FSE ’97 International Workshop, pp. 29-40, LNCS.1267

[4] H. Tanaka, K. Hisamatsu and T. Kaneko “Strength of MISTY1 without FL functions for Higher Order Differential Attack” AAECC13, (1999-11) (Now printing).

[5] S. Moriai, T. Shimoyama and T. Kaneko, “Higher Order Attack of a CAST Cipher”, FSE ’98 pp.17-31 International Workshop, LNCS.1372

[6] T. Shimoyama, S. Moriai and T. Kaneko, “Improving the Higher Order Differential Attack and Cryptanalysis of the *KN* Cipher”, ISEC97-29, pp. 1-8 (1997-09)

[7] H. Tanaka, M. Uemura and T. Kaneko, “On Weak Keys in SPEED Cipher by Higher Order Differential Attack”, ISITA ’98, Oct. pp. 243-246 (1998-10)

[8] S. Contini and Y. Yin; “On Differential Properties of Data -Dependent Rotations and Their Use in MARS and RC6 (Extended Abstract)” http//www.csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm

[9] H. Tanaka, H. Tanaka and T. Kaneko; “Strength of Round Function in RC6 against Higher Order Differential Attack” ISEC99-23, pp. 89-96, (1999-07)

[10] H. Tanaka, H. Tanaka and T. Kaneko; “A Study on Strength of modified RC6 Against Higher Order Differential Attack”, SITA ’99, (1999-10)

[11] H. Tanaka, H. Tanaka and T. Kaneko; “Strength of Round function in RC6 Block Cipher against Higher Order Differential Attack”, JWISC, (2000-1)