RC2 の安全性詳細評価報告

２００１年度

株式会社 日立製作所

RC2の安全性詳細評価報告

要旨: 本報告では、RC2の安全性について弊社による評価の報告を行う。結果としてRC2について、評価者が限られた時間内で安全性評価を行う限りでは、Knudsenの結果よりも大きな安全性の問題点は発見されなかった。
今回の評価では、(1) 従来知られる結果のまとめ、(2) 新しい解読手法の適用可能性検討を行った。特に (2) については個々の解読手法に対するコメントを記し、より重要な攻撃手法については詳細な検討結果を記す。

Abstract: In this report, we address our security evaluation of RC2. As a result, we found no more critical security flaw than what Knudsen described.
Our activities are dedicated to (1) known attacks on RC2 and (2) applicability of new cryptanalytic techniques. Especially for the latter, we leave comments on each new cryptanalysis. For more relevant cryptanalysis, we made detailed technical comments on its applicability.

1 はじめに

本報告は、弊社が CRYPTREC 事務局より委託された業務として行った詳細評価、及び RC2[1]の安全性に関する収集した情報のまとめを行う。

RC2 は 1989 年に Rivest によって開発されたブロック暗号アルゴリズムである。RC2 の安全性に関する文献のうち、評価者が知るものは文献 [KSW97] と文献 [KRRR98] である。このうち [KRRR98] では差分解読法の安全性について質の高い評価を行っている。特に差分解読法は現状、RC2 に対して最も有効であると考えられる攻撃法である。[KRRR98] の結果から、15 段 RC2 について、有意な差分特性が存在するが、16 段完全仕様のものについては鍵導出に成功していない。

本報告書では評価者が行った RC2 の安全性に関する詳細評価について、詳細に報告を行う。報告は安全性評価の結果のみならず、評価者が検討した評価方針やその過程についてもなるべく実験結果などを多く掲載し、第三者が本報告の信憑性を検証できるようにした。

この結果、評価者による詳細評価では RC2 の安全性について、[KRRR98] の結果を上まわる解読手法を見付けることはできなかった。

2 RC2 評価方針

この章では、主に評価方針について述べる。その前に、評価者が最終的な評価方針に至るまでに考慮した RC2 の特徴についてまず説明する。

RC2 は鍵スケジュール部とデータ撹拌部分を持つブロック暗号である。データ撹拌部分は、source-heavy な Feistel 構造をしており、出力は算術可算に巡回シフトを組み合わせた演算である。段関数には論理積演算と上述の算術可算、巡回シフトのみを用いたものである。また、これら RC2 のデータ撹拌部分はワード (16 ビット) 単位の処理でほぼ構成されているという特徴がある。

さらに、段関数は合計 16 段繰りかえされるが、平文側から 5 段、11 段終了時に `MASH` と呼ばれる、補助的な鍵可算段が組み込まれている。

これらの構造上の特徴は、これまで知られるブロック暗号には稀な構造となっており、安心した使用には、幅広い観点からの暗号解読の結果が必要である。

本評価の大きな目標として、等価鍵や DES のビット反転特性など明らかな暗号の欠陥の特性がないことを一点目に、そして第二の目標として、過去の結果の正当性の検証、最後に新しい攻撃手法の適応可能性の検討を置いた。

以下の章では、これらについて別々に評価報告を行ってゆく。

3 準備

記号の定義を行う。定数の表記には主に 10 進数、16 進数、2 進数表記を用いる。10 進数でない場合には、これらが文脈から自明でないとき、数字の添字として、これらを識別する。例えば $23_{16} = 35 = 100011_2$。16 進数表記の場合の 10 以上の数字には順に `a,b,c,d,e,f` を用いる。

二項演算子+は算術加算を表す。特に剰余の大きさ (ビット長) を明示する場合添字でそれを示す。例えば変数 a、b を 16 ビットのレジスタで可算する場合 $a +_{16} b$ と書く。二項演算子 $\oplus, \wedge$ はそれぞれビット毎の排他的論理和、論理積を表す。16 ビット変数 a に対して、左巡回シフトを 16 ビットレジスタで t ビット行う場合、$a <<<_{16} t$ と表記する。2 つの長さが定義された変数 a、b に対して $a||b$ はビット文字列としての連結を示す。ここで注意するのは b の部分はその定義された長さとなるまで上位に 0 がパディングされることである。例えば、ふたつのバイト変数

[1]RC2 は RSA Security Inc. 社の登録商標である。

a、b にそれぞれ 0,1 が代入される場合、$a||b = 0000\ 0000\ 0000\ 0001_{\,2}$ となる。

4 RC2 アルゴリズム

RC2 は文献 [RFC2268] で仕様が定義される 64 ビットブロック暗号である。鍵インターフェース長は可変で 1〜128 バイトとし、この長さ以下の任意ビット長の**実効鍵長**がある。鍵の全数探策は入力鍵ではなく、実効鍵長の中間鍵に対して行うことができるため、実効鍵長を計算量的安全性の指標として扱う。

アルゴリズムは暗号化処理と復号化処理からなり、各々の処理には鍵スケジュール部とデータ撹拌部分からなる。鍵スケジュール部は秘密鍵より 16 ビットの拡大鍵 $K[i], 0 \leq i \leq 63$ を生成する。暗号化 (復号化) のデータ撹拌部分は $K[i]$ を用いて、平文 (暗号文) 64 ビットから暗号文 (平文)64 ビットを生成する。

鍵スケジュール部では、PITABLE と呼ばれる 8 ビット入出力の置換表と 8 ビット算術可算が用いられる。データ撹拌部分は 16 ビット算術可算、16 ビットレジスタによる 1/2/3/5 ビット巡回シフト、論理積、データ依存の拡大鍵可算がある。

4.1 鍵スケジュール部

簡単にアルゴリズムを記述する。バイト列として与えられる鍵 $L[i], 0 \leq i < T$、T バイトは以下の処理により 128 バイトに展開される。

$$
\begin{aligned}
&for\ (i = T, \ldots, 127)\ do: \\
&\qquad tmp1_i \quad = \quad L[i-1] +_8 L[i-T], \\
&\qquad L[i] \quad = \quad \mathtt{PITABLE}[tmp1_i], \\
&endfor.
\end{aligned}
$$

生成された 128 バイトの L はインデックスの大きい方から実効鍵長 $T1$ ビットを残してすべて 0 にリセットされる。より詳細には、$T1 = T1_8 \times 8 + T1_1, 0 \leq T1_1 < 8$ となる $T1_8$ と $T1_1$ を決定したとき、$T[128 - T1_8], \ldots, T[127]$ の $T1$ バイトと $T[127 - T1_8]$ の下位 $T1_1$ ビットが残る。

$T[127 - T1_8]$ はここで PITABLE で一度置換されてから、以下の鍵拡大処理を行う。

$$
\begin{aligned}
&for\ (i = 127 - T1_8, \ldots, 0)\ do: \\
&\qquad tmp2_i \quad = \quad L[i+1] \oplus L[i + T1_8], \\
&\qquad L[i] \quad = \quad \mathtt{PITABLE}[tmp2_i], \\
&endfor.
\end{aligned}
$$

最後に 16 ビット拡大鍵を 64 個以下の埋め込みに従って生成する。

$$
\begin{aligned}
&for\ (i = 0, \ldots, 63)\ do: \\
&\qquad K[i] \quad = \quad L[2i+1]\ ||\ L[2i].
\end{aligned}
$$

ここで埋め込みは little-endian であることに注意する。

4.2 データ撹拌部

本稿では復号化処理には触れないため、暗号化処理のみを記述する。復号化処理が必要な場合、文献 [RFC2268] を参照のこと。

バイト列の平文 $P[i], 0 \leq i < 8$ は 4 個の 16 ビットレジスタ $R[i], 0 \leq i < 4$ に以下のような little-endian で埋め込みを行う。

$$\begin{aligned} & for(i = 0, \ldots, 3)do : \\ & \qquad R[i] \quad = \quad P[2i+1]||P[2i]. \end{aligned}$$

これら 4 つの 16 ビットレジスタに対して、5 段 MIX ラウンド、MASH、6 段 MIX ラウンド、MASH、5 段 MIX ラウンドを順に処理して結果的に R に格納されているデータ 64 ビットを暗号文として出力する。

MIX　1 段 `MIX` は、source heavy な変型 Feistel 構造の 4 段分と考えることもできるが、仕様書 [RFC2268] や Knudsen らの解析結果 [KRRR98] で扱うとおり 1 段 `MIX` を以下の処理を順に行うものとする：

$$\begin{aligned} R[0] \quad &:= \quad \texttt{MIXUP}(R[0], R[1], R[2], R[3], K[4i], 1), \\ R[1] \quad &:= \quad \texttt{MIXUP}(R[1], R[2], R[3], R[0], K[4i+1], 2), \\ R[2] \quad &:= \quad \texttt{MIXUP}(R[2], R[3], R[0], R[1], K[4i+2], 3), \\ R[3] \quad &:= \quad \texttt{MIXUP}(R[3], R[0], R[1], R[2], K[4i+3], 5). \end{aligned}$$

ここで段数カウンタ i は $0/leqi < 16$ とし、上記 4 つの処理が終ると 1 インクリメントする。また、`MIXUP` は以下の処理である：

$$\begin{aligned} MIXUP(source, operand1, operand2, select, key, shift) := (source + tmp3) <<<_{16} shift, \\ tmp3 = (operand1 \wedge select) +_{16} (operand2 \wedge \overline{select}) +_{16} key. \end{aligned}$$

ここで $tmp3$ の定義では、演算子とオペランドの制限から以下のような表記でも、この RC2 の定義では等価である。

$$\begin{aligned} tmp3 \quad &= \quad (operand1 \wedge select) \mid ((operand2 \wedge \overline{select})) +_{16} key, \\ tmp3 \quad &= \quad (operand1 \wedge select) \oplus ((operand2 \wedge \overline{select})) +_{16} key. \end{aligned}$$

ここで一段 `MIX` を図示したものを図 1 に示す。

MASH　`MASH` は、データ依存の鍵加算である。`MASH` の 1 段は以下の順に従って処理される：

$$\begin{aligned} R[3] \quad &:= \quad R[3] +_{16} K[R[2] \wedge \texttt{3f}_{16}], \\ R[2] \quad &:= \quad R[2] +_{16} K[R[1] \wedge \texttt{3f}_{16}], \\ R[1] \quad &:= \quad R[1] +_{16} K[R[0] \wedge \texttt{3f}_{16}], \\ R[0] \quad &:= \quad R[0] +_{16} K[R[3] \wedge \texttt{3f}_{16}]. \end{aligned}$$

これらの処理は、秘密鍵から生成される、鍵依存 6 ビット入力 16 ビット出力の置換表を使った変換とも考えることができる。ここで、ソース側の上位 10 ビットが用いられないことに注意する。

ここで一段 `MASH` を図示したものを図 2 に示す。

5　既存の解読結果

これまで知られた RC2 の解析結果については 2 例が知られている。Kelsery らによる関連鍵攻撃 [KSW97] と Knudsen らによる差分解読、線形解読 [KRRR98] である。

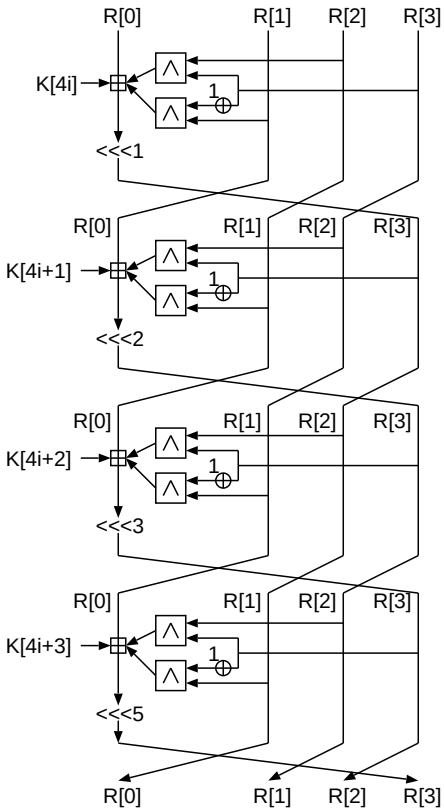

図 1: MIX round of RC2

5.1 関連鍵攻撃

Kelsey らは RC2 に対する差分攻撃を強化する目的で、関連鍵攻撃を提案した。Kelsey らの関連鍵攻撃が成立するには、以下のようないくつかの条件が満たされる必要がある：

1. 鍵長が 64 バイト (=512 ビット) である、
2. 実効鍵長が 1024 ビットである (すなわち、マスク処理や、後半の鍵拡大処理を無視する)、
3. 攻撃者は未知の鍵 K と、ある非線形な変換で処理された関連鍵 K^* の両方の鍵による選択平文を入手できる。

このとき、RC2 の差分攻撃が強化され、2^{34} 個の選択平文により攻撃が可能と記述されている。

これらの RC2 に対する解析の記述は Springer-Verlag の Lecture Notes in Computer Science の実質 1 ページ分であり、各々のデータに対して検証するためにはより多くの情報と時間が必要である。

また、仮にこの攻撃が有効であった場合にも、上に示した前提条件は現実応用例には以下の理由から稀にしか実装されない。まず、実効鍵長が 1024 ビットとひじょうに大きくとっているにもかかわらず、鍵長が 512 ビットに制限されている。また、現状の暗号製品としては過去の輸出規制などの観点から実効鍵長が無制限のものの普及量は多くないと考えられる。さらに、関連鍵が非線形な相関であることから、暗号化オラクルへの関連鍵の挿入は著しく困難となる。

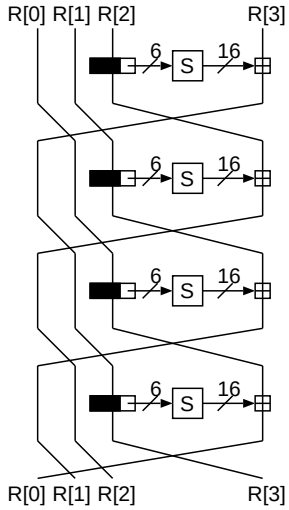

図 2: MASH round of RC2

5.2 差分解読法、線形解読法

RC2 は 1989 年に設計されたアルゴリズムである。現在、暗号の安全性の評価として重要な項目に差分解読法 [BS93]、線形解読法 [M93, M93] がある。これらの攻撃手法はどちらも汎用的な攻撃法であり、かつ RC2 の設計後に提案された攻撃手法であるため、差分解読法、線形解読法に対する RC2 の強度評価は重要である。

文献 [KRRR98] では、RC2 に対する差分解読法と線形解読法について、特性パスの例を示しながら、これらの解読法に対する強度評価を行っている。特に差分解読法については効率的に近似するパスを示し、MASH 付き 15 段 (標準 16 段) のものについて、最大 2^{-58} の差分特性パスほか、いくつかの有意な 15 段差分特性パスを示している。また、これらの自明な発展として、完全仕様の 16 段の有意な差分特性も構成することができる。この 16 段差分特性は乱数との識別に使うことができる。

しかし、結論として full round の RC2 が理論解読できたという結論には至っていない。鍵のゲスを考えた場合、16 段全体で少なくとも $2^{-58.7}$ の差分確率の差分パスが必要であり、現状見つかっている最大差分パス (15 段で 2^{-58}) との間のギャップが埋まっていない。

Knudsen らの差分攻撃はいくつかの実験結果により裏づけられている。MASH 付き 8 段 RC2(8 段分の差分特性確率は 2^{-29}) について、暗号文のハミング重みによる 1 ビットの鍵情報 ($K[0]$) の導出の実験結果を掲載している。これによると攻撃 (0,1 の識別) 成功確率がほぼ 100% となるための必要な選択平文の数は 2^{30} 個程度である。ここでは、Knudsen が使った 8 段の差分パスの例ついて図示する。

また、定数加算が重なることによるマルチパスの影響についても議論し、理論解析値と、実験結果が一致することも 3,4,6,7 段 RC2 を用いて実証している (6、7 段の場合には MASH 付き)。

また、RC2 の線形解読法についても (差分解読法のそれに比べれば簡単に) 評価を行っている。しかし、xor で定義される線形解読法は、算術加算を近似する場合、上位ビットにマスクを立てないよう線形特性を構成する必要がある、ということを主張し、現状ではこれが構成できない、特に巡回シフト演算がこれに貢献していることを記している。また、差分と同様、算術可算によるマルチパスの影響についても議論し、実際に可算を複数回処理したあとに、線形 (特性?) 確率が増幅している例についても指摘している。

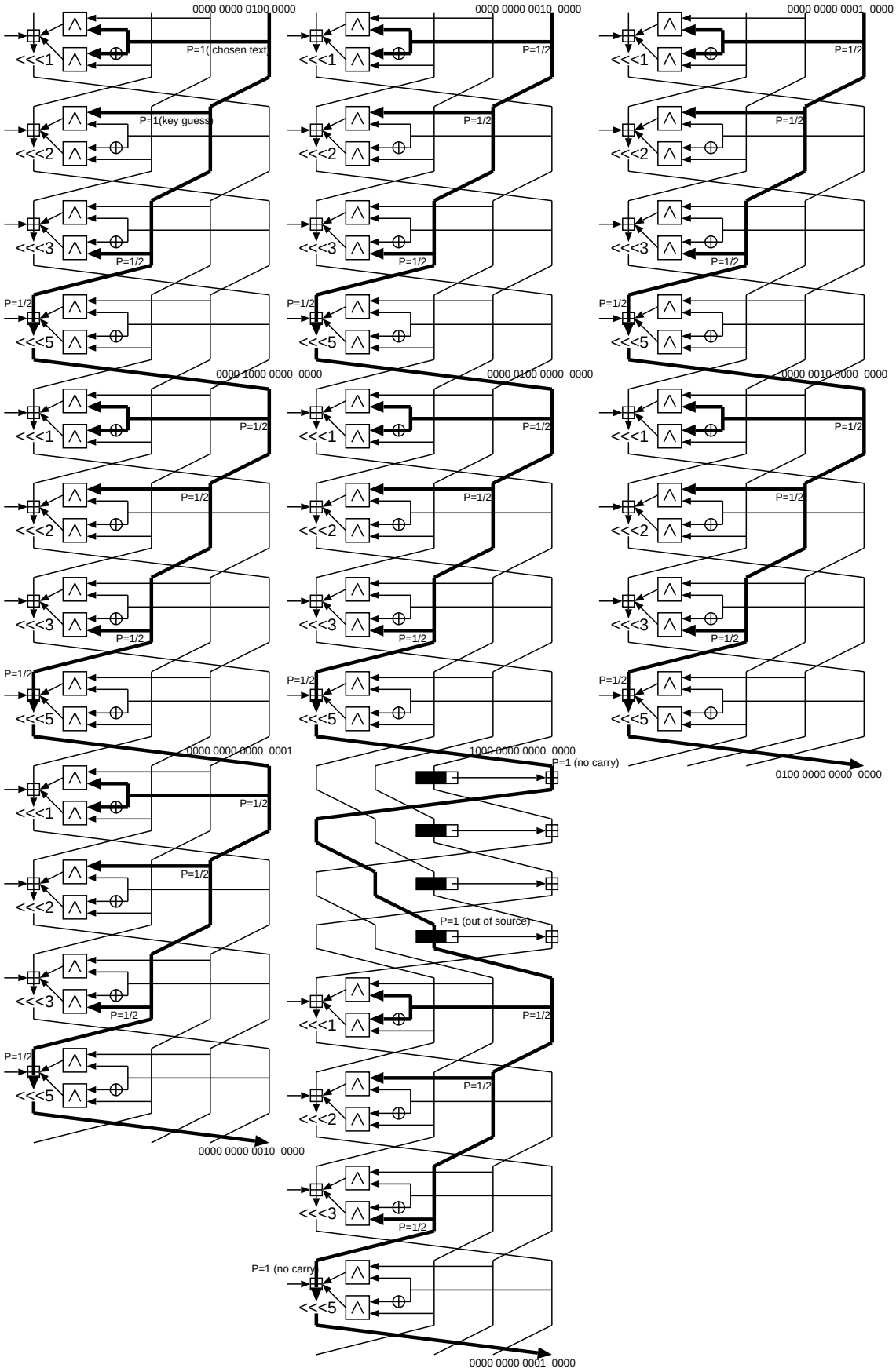

図 3: An example of Knudsen's differential characteristics of RC2

6　近年の解析手法の適用性検討

この章では、これまで知られるRC2の解析結果を基に、これら攻撃手法の拡張や強化について議論する。特に、新しく発見された手法の適用性を重点に置きながら議論を進める。

6.1　鍵スケジュール部の評価

鍵スケジュール部は鍵インターフェース長の柔軟さと、実効鍵長の調整が可能であるという特徴がある。ブロック暗号が期待された安全性を保つためにはこれらすべての組み合わせで用いるのは避けるべきである。特に等価鍵や関連鍵攻撃、中間値一致攻撃などの攻撃を想定して以下の使用は避けるべきである。

1. どのような実装であれ、鍵入力に冗長性(カウンタや定数のパディング、鍵のくり返し)をパディングすること、
2. 不必要に鍵インターフェース長く取り、実効鍵長が極端に短い使用、
3. 関連鍵が想定できるようなプロトコルで、実効鍵長と鍵インターフェース長を長く設定すること。

これらはRC2に起因する問題ではなく、鍵インターフェースの柔軟さから誘惑される、ユーザの侵しがちなミスの指摘である。

以下に簡単に上記使用方法が危険である理由を示す。

鍵の全数探索の減少　鍵インタフェース部は鍵の全数探策の対象とはならない。鍵の全数探策の計算量は、(1)鍵エントロピーから生成されるすべての可能性の数、(2)実効鍵長の全数探索、のどちらか少ない方である。よって、鍵情報にカウンタや定数パディング、鍵のくり返しを挿入することは(1)を劇的に少なくする。また、アルゴリズムの構造上、これらの冗長性も含めて拡大鍵の生成を行うため、鍵の拡大に用いる計算量がこれらエントロピーのない部分へも費されるため、データがじゅうぶんに撹拌されず危険である。その極端な攻撃例として以下のものを最後に示す。

定数パディングにより全数探索が上記(1)(2)よりもさらに下まわる例:
鍵インターフェース長を$T \geq 64$バイト、実効鍵バイト長を$T1_8$とする。秘密鍵のバイト列$L[i], 0 \leq i < T$のうち、$L[j], 129 - T - T1_8 \leq j < 128 - T$の値の少なくとも1つが攻撃者によって、ゲスされる情報である場合、実効鍵長Tよりも少ない計算量で全数探索が可能である。

証明: 攻撃者にとって効率的にゲスできる秘密鍵バイトを$L[j], 129 - T - T1_8 \leq j < 128 - T$とする。このとき、拡張アルゴリズムにより$L[j+T] = \texttt{PITABLE}(L[j] + L[j+T-1])$となる。$129 - T - T1_8 \leq j$より$j + T \geq 129 - T1_8$となり、$L[j+T-1]$、$L[j+T]$の2バイトは両方ともに実効鍵の全数探索の対象である。しかし、$L[j+T]$は$L[j+T-1]$から容易に計算できるため、実質、実効鍵の全数探索の計算量は2^{8T1_8}よりも小さくなる。

明らかな等価鍵の存在　上で述べた、避けるべき使い方の二番目についてより詳細な説明を加える。鍵インターフェース長をTバイトとする。このとき、拡張される$128 - T$バイトの生成には秘密鍵$L[i], 0 \leq i < T$のうち、$L[j], 0 \leq j < 128 - T, j = T - 1$のみが用いられる。よって、$T8_1 \leq 128 - T$の場合には、$2T - 129$バイトの未使用秘密鍵バイトが存在し、自明に1個あたり$2^{16T-258} - 1$個の等価鍵が存在する。また、$T8_1 > 128 - T$にも同様に等価鍵が存在する場合がある。

これらの等価鍵は、鍵インターフェースへ入力される秘密鍵に関するものであり、暗号全体として直接安全性を下げるものではない。しかし、暗号の使い方として、鍵を変化させた場合には異なる暗号処理を行うことが期待される場合がある。このような実装への攻撃手法の例として、関連鍵攻撃のように一部のバイトのみを固定することで、異なる鍵を挿入しつつも、同じ鍵が挿入されたものとしてブロック暗号処理が機能してしまう、という機能のすり替えを使った攻撃が考えられるかもしれない。

現実的にはおこり得る実装はひじょうに限られたものになるが、上位のスキームの攻撃の足がかりとならぬよう、無駄に等価鍵を増やさない使用は考えられるべきである。

関連鍵攻撃について　鍵インターフェース長と実効鍵長が長い場合には、鍵スケジュールの処理自体がひじょうに少なくなり、秘密鍵がそのまま拡大鍵として使われる部分が多くなる。

拡大鍵はデータ撹拌部分への入力となるため、鍵からの安易な制御は避けられるべきである。しかし、上流プロトコルでの鍵交換でメッセージ認証を行ってない場合などには、鍵の改竄や関連鍵攻撃 [KSW97] を実現する切っ掛けになる。

特に秘密鍵の認証が不十分であるような場合には、これら鍵インターフェース長と実効鍵長の長いもの (たとえば両方が 128 バイトなど) の使用は避けるべきである。

6.2　その他 RC2 の解析

ここでは、RC2 に対する他の攻撃手法、ならびに Knudsen の結果の強化する手法の可能性について議論してゆく。議論は二つの観点から行う。一点目は差分確率そのものを強化するアプローチである。これには (1)Knudsen の発見した差分特性とは別のより強力な差分パスの存在の検討、(2)Knudsen の発見した差分特性確率をマルチパスを懸念した差分確率での評価、がある。二点目には、Knudsen の発見した差分特性を使いながらより効率的に鍵のゲスをする手法について議論する。

6.2.1　差分 (特性) 確率の強化

Knudsen の特性確率と最大差分特性確率　Knudsen の指摘した差分特性は発見的手法によるものであり、最良のものである保証はない。よって、より大きな確率の差分特性が存在するかどうかという問題は重要である。ここではまず、シングルパスのみを考慮してゆきこの問題を扱う。Knudsen の差分特性確率が、1 段当たり $1/2^4$ かそれ以上の確率で、くり返し表現を実現している。1 段の差分特性確率には以下の段関数の性質からいくつかの制約がかかる。

1. 段関数は論理積演算、算術加算のみからなっている、
2. 算術加算の場合、差分値の立つビットが最上位ビットのみでない限り、差分確率は高々1/2 である、
3. 論理積に差分値が入る場合、どのような有意な差分表現であっても、差分確率は高々1/2 である、
4. 各 1/4 段毎にすべてのビットが算術加算、論理積のどちらかの演算に必ず用いられるため、Feistel 構造のような 0 差分の 1/4 段は存在しない。

以上の理由から、r 段の `MIX` 関数のくり返し表現を考えた場合、差分特性確率 2^{-4r} を基本に、このうち r 個の算術加算による差分確率のうち (差分値が最上位のみに立つことによる) 決定的差分の確率を引いたものが最良の差分特性となる。文献 [KRRR98] で Knudsen は、1 ビットの差分特性についてはある程度の探索をしたように見えるため、彼が示した差分特性はよい近似を網羅的に調べたことになっている可能性が高く、Knudsen が示す 2^{-58} の差分特性は 15 段 (もしくはその他の段数) の最良差分特性である可能性が強い。

よって、より強力な別の差分特性が例えあったとしても、それほど劇的に差分攻撃を強化するものではない。

Knudsen の特性確率と差分確率　次にこの差分特性確率について、マルチパスの影響による特性確率の増加について検討する。これについては、Knudsen も文献 [KRRR98] において、検討・実証を行っており、[KRRR98] で知ることができる結果以上の期待はできない。

まず、Knudsen は文献 [KRRR98] で、マルチパスによる差分確率の増幅について指摘している。これは、扱う差分表現がビット毎の排他論理和で定義されるものでありながら、それとは異なる可換な群演算 (ここでは法 2^{16} の剰余加算) を繰りかえし近似していることに起因する。このとき、計算可能な差分特性確率に比べて、実際の差分確率が大きくなる可能性がある。

Knudsen らはこれらをある程度の範囲で理論的に強化した。またこれらを実証するために、3,4,6,7 段 RC2 を用いて right pair の数え上げを行い、理論的強化された確率と実際の right pair との比較を行っている。このとき r 段では差分確率は $(1/15)^{r-1}(1/16)$ まで強化されている。

Knudsen の差分確率の検討は 1 ワードの差分値については有意なものを効率的に数えあげており、また実験結果がほぼ一致していることから、真の差分確率に近い評価ができていると考えられる。よって、Knudsen の結果を大きく上回るような差分確率の期待は低いと考える。

6.3 Knudsen の差分確率を使った別の最尤法

Knudsen らは上記の差分確率を用いた鍵の導出に、最初の段の 1 ビット鍵を対象にした真偽判定により、鍵情報 1 ビットの導出を考えている。最終段、すなわち暗号文では、暗号文ペアの差分値のハミング重みを計算し、ハミング重みがある閾値 (64 ビット中 11, 12, 13 などを Knudsen では想定) を越えない場合には 1 のカウントを与え、真偽の判定は、2 つの鍵候補に相当するカウント値の差が *excess* 閾値を越えたときにそれを判定するものである。*excess* 閾値は 2, 4, 8 が考えられているが、実験ではこれら閾値を越えるには 8 段 RC2 に対してそれぞれ約 2^{29}、2^{30}、2^{31} 個の選択平文が必要であるとの実験結果を得ている。

上記の差分確率を使った攻撃の場合、ハミング重みによる判定は以下の二つの理由から有効である。まず、加算の性質から入力差分のハミング重みが 1 である場合、出力差分はハミング重みの小さいものから順に確率が大きくなっていることである。従って、最後の段でパスが崩れるような平文ペアでも Knudsen が導入した *excess* 判定法で有効に性質の崩れを吸収できる。そして、2 番目の理由として、仮に上記の差分パスが数段手前で崩れた場合にもハミング重みの増加は急激でないためある程度の数えあげが期待されることである。つまり最後の数段分の伝播を広く許容できるため、理論的な差分 (特性) 確率よりも広い性質として偽鍵との判定に用いることができると考えられる。

しかし、Knudsen らの手法はゲスする鍵の情報量が少ないために、通常よりもより長い差分パスを使った攻撃となっていると考えられる。一般に、ゲスする鍵の情報量が多くなると、近似するパスの長さは短くなり、近似確率が上昇、結果として解読に必要な選択平文の数が減少する。その反面、使用するメモリの大きさが大きくなることと、判定する偽鍵数の増加による若干の選択平文数の増加という問題が発生する。

今回の RC2 に関しては、使用するメモリの領域が著しく少なく、この改良として暗号文側からの鍵の導出を考えたものも考えられ、ある程度の選択平文数の減少は可能性があると見るべきである。また、完全仕様の RC2 の識別が可能でありながら、鍵導出攻撃が (かろうじて) 見つかっていない現状を考えると、より大きなメモリを用いた差分攻撃は RC2 の安全性評価では重要な課題である。

ここでは簡単な鍵導出アルゴリズムを実装した結果について述べる。攻撃対象は 4 段の `MASH` なし RC2 である。この場合、Knudsen の攻撃では、差分確率 $(1/15)^3(1/16) \approx 1.21 \times 2^{-16}$ であり、必要な選択平文の数は約 2^{16} 個程度と見積もられている。我々の攻撃では、最終段の差分が起こりそうなワードの鍵 $K[15]$ の 16 ビットすべてをゲスして、ある中間値を検査する。この中間値とは、4 段目の `MIX` 関数に含まれる 4 つの `MIXUP` 小関数のうち、3 番目のものを処理する以下の式 :

$$R^{new}[2] \quad := \quad \texttt{MIXUP}(R^{old}[2], R[3], R[0], R[1], K[14], 3),$$

において、$R^{new}[2] - K[14]$ である。差分解読法を適用しているので、より具体的には、$R^{new}[2] - K[14]$ の差分値を検査することになる。暗号の性質上、$K[14]$ をゲスしても有意な最尤法が期待できない。これは RC5[THST00]

の解読にもあるような、偽鍵が正しい鍵と似た統計的振るまいをすることに起因する。ゲスする鍵と、観察する中間値の位置を、図 4 に示す。ただし、この中間値は鍵加算を除けば、さらに 4 段前に生成された中間値であり、期待されるワード差分は 0 である。この 0 差分が生成された平文組数をスコアとして鍵の優先順位を付ける。

今回は詳細な実験資料が掲載できず、また実験の検証も十分でないことをまず最初に明言させて頂く。そのうえでいくつかの実験結果を示す。まず、実験はすべて秘密鍵 8 バイトかつ実効鍵長 64 ビットで行った。また、一部を除き、すべての実験で鍵は $\texttt{0000 0000 0000 0000}_{16}$ に固定した。これは後に示す平文との組み合わせで、第 1MIX 段第 2MIXUP 処理で差分値が決定的にハミング重み 1 に保存でき、差分確率をかせぐことができる。確認のために、このときの我々の攻撃のターゲットとなる $K[15]$ の値は $K[15] = \texttt{a4a5}_{16}$ である。また、与える平文はカウンタ i により定義され、$P_i = \texttt{00 00 00 00 00 00}\ i_u i_l$ である (ここで i_u はカウンタ i の上位バイト、i_l はカウンタ i の下位バイトを表す)。平文組の数により i の走る範囲が変化する。平文組数が $n(< 2^{16})$ のとき、$0 \leq i < n$ である。平文組は P_i と $P_i \oplus \texttt{00 00 00 00 00 00 40 00}_{16}$ であり、i により一意に決定される。ここで上記差分値はバイト列としての並びであり、これが 4 つの 16 ビットワードに埋め込まれると、埋め込み方法が little-endian なので差分値は、$\texttt{0000 0000 0000 0040}_{16}$ となり、Knudsen の差分特性の一つとなる。

$n = 2^{10}$ としたときの Knudsen の判別アルゴリズムが用いる暗号文差分値のハミング重みの分布は表 1 のとおり：ここで参考のために、秘密鍵は $\texttt{ffff ffff ffff ffff}_{16}$ の場合の同じ平文、平文数、平文差分による実験を併記する。ここで注意するのは、後者の場合 1 段目 MIX ラウンドの第 2MIXUP 処理で差分が他のワードへ移ってしまい、right pair ではない場合の差分となる。Knudsen の攻撃ではこれら二者を識別することで鍵 1 ビットを導出する。具体的な判定方法は、ある閾値以下の差分値のハミング重みのみをカウントし、早く *exess* 組だけ差を付けた時点で判定を行い攻撃が終了するものである。今回は 2^{-14} の差分特性確率を持つパスであるにもかかわらず 2^{10} 個の選択平文でも判別できるほどに分布に差があるのがわかる。

我々は特性が顕著に現れる 1 ワードの値に着目することとし、なるべく特性差分確率が高くなるような中間値による判定を行った。2^{10} 個の平文に対して、ゲス鍵依存の中間値に対しハミング重みを計算した結果を表 2 に示す。表 2 に示したとおり、正しい鍵は 0 差分の個数が 2^{16} 個の鍵のうち最大 (5 個) であった。また同じ 5 個のゼロ差分エントリを記録した鍵はもう一つ (a2a5) あった。次の表 3 には、2^{16} 個の鍵候補うち、ゼロ差分をカウントした鍵の個数を示したものである。太字は正しい鍵が含まれているカテゴリである。これらの結果から、最終段の鍵 $K[15]$ については、上記の方法のような鍵候補の順序付けにより、真の鍵 a4a5 が上位に位置付けされており、また選択平文の数も Knudsen の手法と比較できる量である。この最尤法が Knudsen の手法より優れているかどうかは現時点ではわからない。しかし、このような最終段の鍵を導出する手法やそれを強化する方法は他にもさまざま考えられる。

Knudsen が見付けた 16 段の差分特性と、鍵の導出には現状ギャップがあるとされているが、今後の研究、特に最尤法の強化により理論解読ではあるが完全仕様の RC2 が攻撃されることは十分考えうる。

6.4 新しい差分解読法に基づいた攻撃手法

ここでは、RC2 に対するその他の差分解読法に基づいた攻撃手法の攻撃適用性について検討する。ここでは、差分解読法と関係の深い (1) 不可能差分攻撃 [BBS99a, BBS99b]、(2)truncate 差分攻撃 [KB96, KB96]、(3) 算術差分攻撃、(4) ブーメラン/増幅ブーメラン/長方形攻撃 [W99, KKS00, BDK01] について議論する。

不可能差分攻撃　不可能差分攻撃 [BBS99a, BBS99b] は暗号アルゴリズムの解析により、差分確率が 0 の特性を使った攻撃である。典型的には、平文と中間値 (最終段の入力など) との間の不可能差分を使って、ゲスする最終段の鍵のうち実際の暗号文から不可能差分を生成するものをふるい落とすことで、残った鍵候補を正しい鍵とする攻撃手法である。

不可能差分は主に大域的な撹拌が疎であるような場合に有効であり、不可能差分の構成には、差分値の決定的な

偏りを平文側、暗号文側からそれぞれ考え、これらを矛盾させることで不可能差分を生成する方法が一般的である。

RC2 に対する不可能差分を考えるために、まず平文側からの決定的な差分特性の偏りについて考える。ここでは二通りの決定的な偏りの探索を考える。

一つめは、ビット毎に独立に 0(差分なし)/1(差分あり)/?(差分値は確率的に変動) の 3 つの状態を有力な入力差分に対して観察した。ここでは計算機による簡単な探索を行った。ここでは各々の演算子について差分伝播の法則を以下のように適用した：

論理積：
入力差分 (0,0) →出力差分 [0],
その他→出力差分 [?]
算術加算：

1. まず出力差分状態を「0」にセットする、
2. 二つの入力差分のうち「1」の状態である最下位ビットに相当する出力差分ビットを「1」にセットし、それより上位は「?」にセットする、
3. さらに二つの入力差分のうち「?」最下位ビットに相当する出力差分ビット、およびそれより上位の出力差分ビットをすべて「?」とする。

これらのルールの適用の結果、決定的な差分の偏りは多くとも 5/4 段までしか偏りを制御することができなかった。これは、特に 2 段目の最初の 1/4 ラウンドで、加算により最上位に立った「?」状態が、加算後の 1 ビット巡回シフトにより最下位として、他のワードへ最下位に「?」差分を立ててしまい、それぞれ後につづく算術加算ですべてのビットが「?」にセットされてしまうため、偏りが検出できなくなる。

しかし実際には算術加算は、あるビットから上位に向かって連続的に差分を立てる性質がある。そこで二つ目の不可能差分の考え方として、隣り合うビットとの相関を考えることを考慮する。これにより、加算演算については、より詳細な決定的差分の偏りを観察できることが期待できる。しかし、この場合、もう 1 つの演算子である論理積演算子が隣のビットとの相関を崩しながら、加算演算の被演算子の差分の状態を生成する。よって、この隣り合うビットとの相関を記録しながらの決定的な差分の偏りもすぐに消えてしまう。

以上二点から、不可能差分を生成するための決定的な差分の偏りを解析したが、結果として完全仕様の 16 段など高い段数への攻撃に用いることができるものは存在しない。

truncate 差分攻撃　近年提案されるブロック暗号アルゴリズムは、差分特性確率が小さくなるように設計されており、通常ではそのまま差分攻撃を適用するはできない。しかし、byte oriented な暗号など、バイト毎の差分特性をうまくまとめて扱い、一部の (典型的にはミクロな観点の) マルチパスを効率的に扱うことで差分特性確率よりも大きな差分特性の束を考えることができる場合がある。Truncate 差分攻撃 [KB96, KB96] は、このような差分確率を高める手段の一つである。

ここで考えている RC2 はその構造から次の二つの truncate 差分の構築が可能である。一つ目は 16 ビットワード単位の truncate 差分である。この場合、定数加算などのワード内の差分の発散を、確率を失うことなく扱うことができる。しかしながら、RC2 には両方に差分が入る算術加算と、ビット論理積演算が組み込まれている。これらを無条件で扱おうとすると、さきに示した Knudsen の差分特性よりもひじょうに悪い確率の近似となってしまう。また、RC2 への差分攻撃の観点からはなるべく差分ハミング重みが少ない性質を解析する必要があるにもかかわらず、このような truncate 差分の取り方では、差分ハミング重みの少なさを保存することができず、攻撃として非効率的である。

次に、差分ハミング重みを意識した truncate 差分の考え方の例として、より細かく truncate したものを考えてみる。各ワードを bit slice した 4 ビット毎の truncate を行ったモデルである。この場合には、ビット毎の演算、論理積をほぼ確率のロスをすることなく性質を保存できる一方、加算についてはある程度の確率を意識しなければならない。しかし、この truncate 差分の大きな問題点は、暗号処理に用いられる 16 ビットレジスタによる巡回シフト演算である。加算に加えて、巡回シフトでも確率が効くため、Knudsen の差分パスほど効率のよい近似はできないと考えられる。

算術差分攻撃　この攻撃は算術演算に基づく差分値の定義を用いて差分特性を考えるものである。特に Feistel 構造で用いる排他論理和演算が加算演算に置き換わった構造をした暗号への近似に有効である。

RC2 もまた拡張 Feistel 構造の $target$ への演算が算術加算演算であり、算術差分を考えることは重要である。ここでは、Knudsen の近似にならい、各段で少なくとも 2^{-4} で近似できるような算術差分特性の構築を試みる。ここで用語を整理する。`MIXUP` 小関数には 1 つの $target$ と 3 つ source の、合計 4 つのレジスタがある。さらに 3 つの source レジスタは、両方の論理積演算の入力となる $selector$ と、二つの (論理積の片方にしか影響しない)$operand1$, $operand2$ からなる。

ここで算術差分が 0 でないと仮定し、$target$, $selector$, $operand$ でどのような確率を考えなければならないかまとめる。$target$ に算術差分が入ったとき、演算子は算術加算と巡回シフトである。算術加算は決定的に算術差分を伝播するので、近似の効率はよい。しかし、t ビット巡回シフト演算は、ラフに見積もった場合 2^t 演算と考えることができるが t の値に応じて近似確率 $2^t - 1$ を考えなければならない。

算術差分が $selector$ に立っている場合、これは両方のビット論理積に影響する。特性確率のうち (自明でなく) 最大のものを考えると、論理積への影響がなるべく少ないような算術差分の場合である。しかし、論理積への影響ビット数は、算術差分で決定的に扱えず確率的に考える必要がある。論理積はビットあたり 1/2 の確率がかかることを考えると、算術差分が $2^t, 0 leqt < 16$ であって、ビット差分のハミング重みが 1 となる場合が (論理積の確率も含めて) 最良と考えられる。算術差分が 2^t で、かつ通常の差分表現による差分ハミング重みが 1 となる確率は 1/2 であるが、論理積でこの差分が消失するのはさらに 1/2 の確率となり、合計で多くとも 1/4 の確率がかかってしまう。算術差分が $operand$ にかかっている場合も $selector$ と同じ理由で、最大でも 1/4 の確率を上乗せする必要がある。

RC2 の `MIXUP` 小関数は、入力-出力は単射であり、差分値が消失することはなく、また $target$ を除いてはワードが active な場合次の `MIXUP` も active となる。$target$ に関しても、その接続状態から次の二つのうち少なくとも一方は真である：(1) 次の `MIXUP` の $selector$ が active となる、(2) ソースがすでに active である、すなわち $selector$, $operand$ がすでに active である。このような差分伝播の条件から一段あたり 4 回の `MIXUP` を近似する必要があり、この場合の確率は Knudsen の近似よりもいいものを期待できない。

ブーメラン攻撃など　近年提案された、ブーメラン攻撃 [W99] や増幅ブーメラン攻撃 [KKS00]、長方形攻撃 [BDK01] などは差分解読法を強化した攻撃法である。

ブーメラン攻撃は適応的選択平文暗号文攻撃であり、攻撃者がひじょうに限定される。これに対し増幅ブーメラン攻撃は選択平文攻撃へと拡張したものであるが、ブーメランに比べて効率が多少劣る。さらに長方形攻撃は増幅ブーメラン攻撃と結局同じであるが、選択平文数の見積もりの手法が異なっている。

これら、どの攻撃にも共通しているが、差分解読法よりも効率的に動くための条件がある。暗号アルゴリズムが平文側、暗号文側で二分割したとき、平文側処理の差分確率と暗号文側処理の差分確率の積が、暗号全体の差分確率を著しく上回っているときにのみ適用可能である。典型的には、暗号アルゴリズムの構造上、ある段数以上では急激に差分確率が低下する、などの暗号で効率的である。

RC2 に関してはこの性質はなく、とても効率的な繰り返し表現が見付かっている。よって Knudsen の攻撃以上の効率で、これらが攻撃可能とはならない。

6.5 その他の新しい解読法

高階差分攻撃、Square 攻撃 高階差分攻撃 [JK97] は代数的な次数の低さを利用した攻撃である。Square 攻撃 [DKR97] はそのバリエーションのひとつである。RC2 では論理積演算と算術加算が同時に使われる。論理積演算は標数 2 で扱いやすいのに対して、算術演算は剰余群上で簡単に扱うことができる。またその逆は近似が複雑となり一般にはあまり考えない。よって RC2 はどちらの演算種類で考えたとしてもすべての演算を簡単に扱うことは困難である。確率的な近似を採り入れればこの複雑さをある程度無視することはできるが、すでに差分特性確率で各 `MIXUP` を 1/2 で近似できる攻撃が知られているので、高階差分攻撃は Knudsen の差分攻撃ほど大きな脅威にはならない。

関数の出力偏差を利用した攻撃 DES の三番目の攻撃として知られる Davie's attack[BB97] は Feistel 構造で用いる通称 F 関数の出力に偏差があることを用いた攻撃である。RC2 では source heavy な段関数を使った Feistel 構造と考えることができるが、この段関数は multiple permutation、すなわち入力が均等である場合、出力も均等であることから、Davis attack は適用できない。ただし、選択平文などで入力に偏差がある場合はその限りではないが、RC2 は段数も多く、頻繁に target-source が入れ替わるので、特殊な選択平文による Davis attack は完全仕様の RC2 には脅威とはならない。

線形解読法・Partitioning 解析・χ^2 検定 線形解読法 [M93, M93] は偶数パリティ値の保存を利用した攻撃手法である。算術加算演算は線形解読法でさまざまな近似が可能であり、重要な攻撃手法である。しかし、有意な確率で近似できるためのマスク値への条件も多数あり、特に上位ビットにマスクが立つ場合には、その下位にもマスクがない限り (Knudsen の差分特性に比べて) 効率的な近似はできない。RC2 では 16 ビットの巡回シフト演算が線形解読法に対し、有効に働いており、以下の理由から効率的な近似は望めない。

その理由を簡単に記述する。まず巡回シフト数を無視した構成の場合、1 段の `MIX` は偏差 $2^3 \times 2^{-2}2^{-2}2^{-1}2^{-1}$ 程度の効率的な近似が可能である。ただし、この確率は (1) 算術加算の近似、(2) 論理積の近似の両方からなっており、近似を多段に拡張した場合、(1)(2) が両方ともに大きく影響することを考慮する必要がある。RC2 には固定の巡回シフトがあるために、マスク値が算術加算の上位に立ってしまう。この場合には (1) がひじょうに小さくなってしまい、攻撃には不適切である。これを避けるために、隣接する下位ビットにもマスク値を立てることや、移動したあとのビットにマスク値を予め立てておいて後でキャンセルアウトする方法などで高いビット位置のマスクを避けることが考えられる。しかし、これらキャンセルするために立てたマスクは、最初の平文から近似しなければならず、その途中の加算、論理積演算で上記 (1)(2) の確率が効いてしまう。よって、巡回シフトの数が重要であるが、現状の RC2 の仕様で策定された巡回シフト量は線形解読法を排除する安全な仕様である。

ただし、線形近似の対象がそれほど長くない場合にはこの議論の限りでない。例えば、暗号のほとんどの部分を差分特性で近似し、平文側、または暗号文側数段を線形近似する場合には、有意な線形近似は複数存在する。

Partitioning[HM97] 解析や χ^2[KM00] 検定についても同様に、長い段数の性質については、現状の解析結果では、Knudsen の結果を越えるものは期待できない。ただし partitioning 解析は、概念としては広いものの実際にどのような partition を考えたらよいか、またその探索方法などは考えられておらず、解読の分類として存在するだけで、実際には暗号への脅威という意味では一般にそれほど大きくない。ただし、χ^2 検定は、差分解読法を強化する最尤法への適用としてはじゅうぶん役に立つであろう強力な解読テクニックである。

mod n 解析 算術加算は、ビット単位の処理との相互の性質の保存が一般には難しい。この mod n 解析 [KSW99] は算術加算と巡回シフト演算の混在する暗号アルゴリズムについて、効率的に統計的性質を扱うテクニックである。RC2 では、算術加算と巡回シフトが混在しており、部分的には mod n 解析により効率的な近似が可能である。しかし、算術加算の被演算子である論理積出力は、mod n 解析で扱う剰余の解釈では効率が悪い。現状の解析結果で

は適用できない、と考えるが、論理積をうまく近似する手法が見付かれば mod n 解析に関する安全性評価は重要となる。

スライド攻撃　スライド攻撃 [BW99, BW99b] は段関数と拡大鍵の周期性や (上下の) 対称性を利用して非線形変換の機能を skip する手法である。スライド攻撃を適用するためには、段関数と拡大鍵に周期性や対称性を見付ける必要があるが、RC2 には MASH 段、算術加算、巡回シフトなど方向性のあるものが適宜利用されており、スライド攻撃は適用できない。

補間攻撃、線形和攻撃　補間攻撃 [JK97] は暗号で用いられる変換や演算を簡単に代数体などで表現できる場合に、暗号全体として複雑な関数とならないことを用いた攻撃である。線形和攻撃 [A99] はこれを強化するアイデアである。

RC2 については、高階差分攻撃で述べたように、論理積演算と算術加算がお互いに近似しにくい関数となっており、標数が 2 の体や剰余環、どちらでも暗号で使われるすべての部分関数を効率的に近似することはできない。

7　まとめ

本報告では、ブロック暗号 RC2 の安全性評価について、既存の結果のまとめ、検証と、これらの強化、そして新しい攻撃手法の適用性の検討を行った。

評価者の主張として、大きな論点では、

1. Knudsen の攻撃は現状でも最強の攻撃手法であるが、現状乱数との判別が可能なだけで鍵の導出に至っていない。しかし、鍵の導出が強化される可能性はじゅうぶん残っており今後も注意が必要である、
2. 鍵スケジュール部は仕様が極度に柔軟であり、ユーザーの誤った使用で極端に弱くなることがある。安心して使うことができるパラメータ、例えば鍵インターフェース長 64 ビット、実効鍵長 64 ビットなどに設定した上での普及が望ましい、

がある。

参考文献

[A99] 青木和麻呂, "線形和攻撃," 1999 年暗号と情報セキュリティシンポジウム予稿集, 電子情報通信学会, 1999.

[THST00] 竹内清史, 早川珠理, 下山武司, 辻井重男, "共通鍵ブロック暗号 RC5, RC6 に対する correlation attack," 2000 年暗号と情報セキュリティシンポジウム SCIS2000 講演予稿集, 2000.

[BB97] A. Biham, A. Biryukov, "An Improvement of Davies' Attack on DES," *Journal of Cryptology, Vol. 10, No. 3*, Springer-Verlag, 1997.

[BBS99a] A. Biham, A. Biryukov, A. Shamir, "Miss in the middle attacks on IDEA, Khufu, Khafre," *Fast Software Encryption, 6th International Workshop, FSE'99, Proceedings*, LNCS Vol. 1636, Springer-Verlag, 2000.

[BBS99b] A. Biham, A. Biryukov, A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," *Advances in Cryptology, -EUROCRYPT'99*, LNCS Vol. 1592, Springer-Verlag, 1999.

[BS93] E. Biham and A. Shamir, *Differential Crptanalysis of the Data Encryption Standard*, Springer-Verlag, New York, 1993.

[BDK01] E. Biham, O. Dunkelman, N. Keller, "The Rectangle Attack – Rectangling the Serpent," *Advances in Cryptology–EUROCRYPT 2001*, LNCS Vol. 2045, Springer-Verlag, 2001.

[BW99] A. Biryukov, D. Wagner, "Slide Attacks," *Fast Software Encryption, 6th International Workshop, FSE'99, Proceedings*, LNCS Vol. 1636, Springer-Verlag, 2000.

[BW99b] A. Biryukov, D. Wagner, "Advanced Slide attacks," *Advances in Cryptology–EUROCRYPT 2000*, LNCS Vol. 1807, Springer-Verlag, 2000.

[DKR97] J. Daemen, L. Knudsen, V. Rijmen, "The Block Cipher SQUARE," *Fast Software Encryption, 4th International Workshop, FSE'97, Proceedings*, LNCS Vol. 1267, Springer-Verlag, 1997.

[HM97] Harpes, C., Massey, J. L., "Partitioning Cryptanalysis," *Fast Software Encryption, 4th International Workshop, FSE'97*, LNCS Vol. 1267, Springer-Verlag, 1997.

[JK97] Jakobsen, T., Knudsen, L. R., "The Interpolation Attack on Block Ciphers," *Fast Software Encryption, 4th International Workshop, FSE'97*, LNCS Vol. 1267, Springer-Verlag, 1997.

[KSW99] J. Kelsey, B. Schneier, D. Wagner, "Mod n Cryptanalysis, with Applications Against RC5P and M6" *Fast Software Encryption, 6th International Workshop, FSE'99*, LNCS Vol.1636, Springer-Verlag, 2000.

[KSW97] J. Kelsey, B. Schneier, D. Wagner, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," ICICS'97, Springer-Verlag, 1997.

[KKS00] J. Kelsey, T. Kohno, B. Schneier, "Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent," *Fast Software Encryption, 6th International Workshop, FSE'2000*, LNCS Vol.1978, Springer-Verlag, 2001.

[KB96] L. R. Knudsen, T. A. Berson, "Truncated Differentials of SAFER," *Third International Workshop of Fast Software Encryption*, LNCS Vol.1039, Springer-Verlag, 1996.

[KRRR98] L. R. Knudsen, V. Rijmen, R. L. Rivest, M. J. B. Robshaw, "On the Design and Security of RC2," *Fast Software Encryption, 5th International Workshop, FSE'98*, LNCS Vol. 1372, Springer-Verlag, 1999.

[KM00] L. R. Knudsen, W. Meier, "Correlations in RC6 with a Reduced Number of Rounds," *Fast Software Encryption, 6th International Workshop, FSE'2000*, LNCS Vol.1978, Springer-Verlag, 2001.

[M93] M. Matsui, "Linear Cryptanalysis method for DES cipher," *Advances in Cryptology – Eurocrypt'93*, Lecture Notes in Computer Science Vol.765, Springer-Verlag, 1994.

[M93] M. Matsui, "The first experimental cryptanlaysis of the Data Encryption Standard," *Advances in Cryptology – Ccrypto'93*, Lecture Notes in Computer Science Vol.839, Springer-Verlag, 1994.

[KB96] M. Matsui, T. Tokita, "Cryptanalysis of a Reduced Version of the Block Cipher *E2*," *Fast Software Encryption, 6th International Workshop, FSE'99*, LNCS Vol.1636, Springer-Verlag, 2000.

[RFC2268] R. L. Rivest, A Description of the RC2$^{\mathrm{TM}}$ Encryption Algorithm, File `draft-rivest-rc2desc-00.txt` available from `ftp://ftp.ietf.org/internet-drafts/`.

[W99] D. Wagner, "The Boomerang Attack," *Fast Software Encryption, 6th International Workshop, FSE'99*, LNCS Vol.1636, Springer-Verlag, 2000.

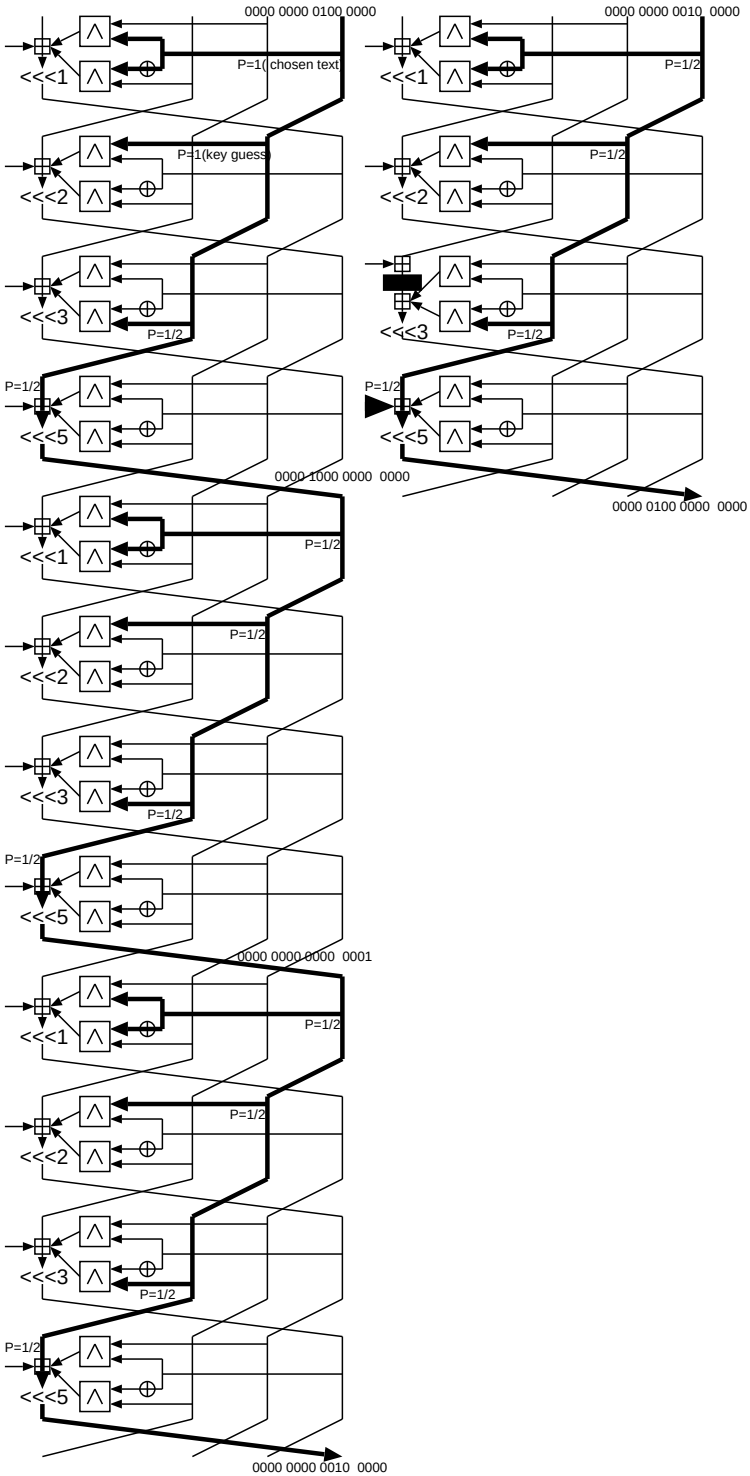

図 4: An attack strategy of 4-round RC2 (reduced version)

表 1: Example hamming-weight distribution of 4-round RC2 (reduced version)

Hamming weight	Right key	Sample wrong pair	Hamming weight	Right key	Sample wrong pair
0	0	0	1	0	0
2	0	0	3	0	0
4	0	0	5	0	0
6	0	0	7	1	0
8	0	0	9	0	0
10	1	0	11	0	0
12	0	0	13	3	0
14	2	0	15	2	0
16	3	0	17	7	0
18	4	1	19	12	0
20	9	1	21	12	6
22	11	7	23	21	6
24	23	13	25	45	25
26	42	27	27	71	57
28	50	59	29	81	79
30	78	70	31	72	85
32	79	93	33	95	98
34	71	91	35	66	93
36	45	67	37	42	59
38	28	29	39	19	15
40	12	17	41	11	17
42	6	5	43	0	2
44	0	1	45	0	0
46	0	1	47	0	0
48	0	0	49	0	0
50	0	0	51	0	0
52	0	0	53	0	0
54	0	0	55	0	0
56	0	0	57	0	0
58	0	0	59	0	0
60	0	0	61	0	0
62	0	0	63	0	0
64	0	0			

表 2: Example hamming-weight distribution of the intermediate value guessed by $K[15]$ in 4-round RC2 (reduced version)

鍵候補	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	0	1	9	19	47	95	143	172	180	164	100	67	16	9	2	0	0
0001	0	2	9	21	64	99	160	163	176	149	96	65	13	6	1	0	0
0002	0	3	6	23	48	93	138	179	188	141	110	54	29	12	0	0	0
0003	0	3	7	28	51	100	142	186	171	148	118	46	21	3	0	0	0
0004	0	4	9	20	43	101	132	165	182	158	113	66	21	10	0	0	0
...																	
a2a5	5	6	38	70	118	131	145	173	151	104	56	20	4	3	0	0	0
...																	
a4a5	5	33	71	105	129	131	148	133	106	90	38	25	8	2	0	0	0
...																	
ffff	0	1	11	17	48	92	166	168	176	173	99	55	15	1	2	0	0

表 3: Number of guessed keys that counts a certain number of the zero difference over all plaintext pairs

選択平文数	(ゼロ差分となる平文組総数) ゲス鍵の数						
2^9	(0)64414	(1)1077	**(2)44**	(3)1	–	–	–
2^{10}	(0)52291	(1)12265	(2)872	(3)99	(4)7	**(5)2**	–
2^{11}	(0...7)59913	(8...15)5499	(16)65	(17)41	(18)9	**(19)5**	(20)4