共通鍵ブロック暗号CLEFIAの
安全性評価報告書

平成23年1月31日

東京理科大学理工学部電気電子情報工学科

金子 敏信

概要

CLEFIA は、2007 年に SONY より発表された [5]128 ビットブロック、鍵長 128/192/256 ビットをサポートする共通鍵ブロック暗号である。データ処理部は、4 系列 Type-2 一般化 Feistel 構造を持ちそのラウンド数は、128 ビット鍵に対し 18 ラウンド、192,256 ビット鍵に対し、それぞれ 22,26 ラウンドである。

本研究では、CLEFIA の安全性に関し、自己評価書 [1] [12] 及び関連論文をベースとし、その記述を計算機実験を含め確認すると共に、新たな評価を付け加えた。安全性評価は、差分攻撃、線形攻撃、不能差分攻撃、高階差分攻撃（飽和攻撃)、補間攻撃及び代数攻撃の立場からデータ攪拌部に対し評価を行い、関連鍵攻撃の立場から鍵処理部に対し評価した。

以下、各手法の評価結果及び明らかになった点をまとめる。

差分攻撃 CLEFIA で使用されている 2 種類の S-box（S_0,S_1) に関し、差分確率を求め、自己評価書の最大差分確率の値を確認した。CLEFIA で使用されている 2 種類のラウンド関数の truncate 差分確率を求めた。これによると、S_0,S_1 の最大差分確率と拡散行列の分岐数で求めた差分確率よりも、真の truncate 差分確率の方が小さくなる場合がある。この確率を使用し、データ攪拌部の最大差分特性確率の上界を Viterbi 探索により求めた。自己評価書では、DSM(Diffusion Switching Mechanism) の効果を分岐数で評価し、active S-box 数の最小値で最大差分特性確率の上界を評価している。本報告書では、ラウンド関数内の分岐数条件は、関数の truncate 差分確率で置き換え、2 種類の S-box の差分確率の違いを反映し、DSM の効果は、自己評価書に述べられている分岐条件を Viterbi 探索時のトレリス線図の遷移条件として反映している[1]。結果として、128,192,256 ビット秘密鍵に対し、自己評価書での上界は、それぞれ $2^{-205.48}$,$2^{-256.85}$,$2^{-303.55}$ であるのに対し、本報告書の評価では、この上界が $2^{-227.42}$,$2^{-282.78}$,$2^{-338.46}$ となり、より精緻化されている。安全性指標である 2^{-128},2^{-192},2^{-256} を、初めて下回るラウンド数は、それぞれ 12,16,20 ラウンドであり、仕様のラウンド数より小さいので、通常の差分攻撃に対し CLEFIA は、耐性を有すると判断する。

線形攻撃 最大線形特性確率の上界を truncate 解析で評価する。2 種類の S-box（S_0,S_1) の最大線形確率は、自己評価書の値で有る事、ラウンド関数の truncate 差分確率は、S_0,S_1 の最大線形確率と分岐数で定まるものと一致する事を確認した。この確率を使用し、前項と同様の手法でデータ攪拌部の最大線形特性確率の上界を Viterbi 探索により求めた。ラウンド関数内の分岐数条件は、関数の truncate 線形確率で置き換え、2 種類の S-box の線形確率の違いを反映し、DSM の効果は、自己評価書に述べられている分岐条件を Viterbi 探索時のトレリス線図の遷移条件として反映している。結果として、128,192,256 ビット秘密鍵に対し、自己評価書での上界は、それぞれ $2^{-201.48}$,$2^{-240.90}$,$2^{-289.08}$ であるのに対し、本報告書の評価

[1] 文献 [13] で新たに示された DSM の効果は、盛り込んでいない。しかし、使用しているラウンド関数の truncate 差分確率に関しては、本報告書の方がより正確な、取り扱いであり、本報告書の手法に、文献 [13] の DSM の効果を組み込む事で、よりタイトな上界が期待できる

では、この上界が $2^{-222.54}$,$2^{-277.38}$,$2^{-331.38}$ となり、より精緻化されている。安全性指標である 2^{-128},2^{-192},2^{-256} を、初めて下回るラウンド数は、それぞれ 12,16,21 ラウンドであり、仕様のラウンド数より小さいので、通常の線形攻撃に対し CLEFIA は、耐性を有すると判断する。

不能差分攻撃 現在までに知られている、CLEFIA に対する最良の攻撃は、辻原らの不能差分攻撃である。9 段の不能差分パスを利用し、128 ビット鍵に対して 12 段まで、鍵の総当たりより少ない計算量で攻撃が可能と報告されている。この攻撃の必要平文数は 2^{111}、計算量は 2^{111} である。同じ不能差分パスを用い、192,256 ビット鍵の場合、それぞれ、13 段、14 段まで、必要平文数 $2^{111.8}$,$2^{112.3}$, 計算量 2^{155},2^{220} で攻撃が可能とされている。今回の評価においては、彼らと同様の不能差分パス探索プログラムを作成し、この結果を追試すると共に、他の可能性を調査した。結果として、彼らの不能差分パスが、現時点の最良パスである事及び彼らの評価が妥当である事を確認した。これより、CLEFIA は、不能差分攻撃に対し耐性を有すると判断する。

高階差分攻撃（飽和攻撃） データ攪拌部に対し 8 階、16 階、24 階、32 階の高階差分特性を、計算機探索した。これらは、S-box 単位の飽和攻撃の調査になっている。32 階差分に関し、自己評価書と同じく 6 段の高階差分特性（飽和特性）が見いだされた。自己評価書では、この特性が、6 段目の出力 4 ワード (=128 ビット) 中の 1 ワード (=32 ビット) に出現するとされているが、実験結果では 2 ワード (=64 ビット) に出現している。これは、角尾らの評価結果に於いても示されている。従来の技法で、視察により飽和特性解析をすると飽和特性は、1 ワードにしか出現しない。角尾らは、この新しい飽和特性を”特殊なバランス状態”と呼び、SP 構造のラウンド関数を持つ一般化 Feistel 構造において出現し、m ビットの独立なバランス出力 n 本を XOR 加算したデータは、$m \leq 2n$ の時、特殊な性質を持つと予想している [8]。ここでは、この特殊なバランス状態が成り立つ事を、より広い範囲の関数に対し、数学的に証明した。CLEFIA のデータ攪拌部の 6 段飽和特性は、2 段拡張可能であり、8 段 96 階差分特性（飽和特性）として、攻撃に利用可能である。自己評価書では、この 8 段飽和特性を利用し、10 段 CLEFIA が鍵の総当たりより少ない計算量で攻撃可能としている。本評価では、攻撃アルゴリズムを改良し、128 ビット鍵の場合、選択平文数 $2^{97.6}$、計算量 2^{98} で攻撃可能であり、192,256 ビット鍵では、それぞれ、11,12 段まで、選択平文数 $2^{98.3}$,$2^{98.8}$ 及び計算量 2^{159},2^{223} で攻撃可能であることを確認した。しかし、フルラウンドの CLEFIA は、高階差分攻撃（飽和攻撃）に対し、耐性を有すると判断する。

補間攻撃及び代数攻撃 S-box (S_0,S_1) のブール多項式代数表現及び項数を求め、S_0 に関し 6 次、S_1 に関し 7 次であり、自己評価書と一致している事を確認した。従って、F_i 関数 1 段については、6 次ないし 7 次である。データランダム化部の構造を見ると同一の F_i 関数の直列接続で次数の上昇が起き、異なる F_i 関数の直列接続では無い事に着目し、F_i 関数の直列接続 3 段までに関し最高次数項を求めた。自己評価書では、6（又は 7) のべき乗で次数上昇を見積もっているが、実際に確認すると、2 段の F_i 関数で 28 次であり、3 段で 31 次である。しかし、フルラウンドでは、18 段以上あり、自己評価書の主張と同じ理由でブール多項式に基づく補間攻撃や代数攻撃は、困難と考える。また、拡大体上の代数攻撃を意識し、S-box に対し、$GF(2^8)$ 上の多項式表現を、すべての法多項式の上で求め、項数を評価した。多項式表現の最小個数は、自己評価書の値と一致している。自己評価書と同じ理由で、この種の攻撃は、困難と考える。

関連鍵攻撃 鍵関連攻撃の実装の為には、鍵処理部の差分特性の把握が必要である。128 ビット鍵の場合、鍵処理部は、データ攪拌部と同じ、4 系列 Type2 一般化 Feistel 構造 12 段であり、データ攪拌部の差分攻撃耐性評価がそのまま使用でき、最大差分特性確率の上界は $2^{-144.39}$ である。192 及び 256 ビット鍵で 使用される鍵処理部は、8 系列 Type2 一般化 Feistel 構造 10 段であり、ここでは、その最大差分特性確率の上界を、新たに truncate 解析で評価した。その値は、$2^{-151.72}$ である。128,192,256 ビット鍵、何れも、十分小さな最大差分特性確率であり、関連鍵攻撃に対する耐性に問題は無いと考える。また、鍵処理部の高階差分特性を評価し、192、256 ビット鍵の場合、フルラウンドの 10 段に渡る 32 階の高階差分特性（飽和特性）が存在する事、9 段に対し、8 階の高階差分特性（256 種類の鍵の組）が存在する事を発見した。この特性が直接に攻撃に結びつくとは、考えにくいが、CLEFIA をそのまま、ハッシュ関数の構成部品として使う場合、注意が必要であろう。なお、128 ビット鍵の場合、このような特性は存在しない。

目 次

第1章　はじめに

CLEFIA は、SONY により、FSE2007 において発表された共通鍵ブロック暗号である。2009 年度の CRYPTREC による「電子政府推奨暗号リスト改訂の為の暗号技術の公募」に応募した共通鍵暗号の一つである。ここでは、CLEFIA の安全性評価として、汎用の攻撃法である差分攻撃、線形攻撃、高階差分攻撃、補間攻撃、代数攻撃、関連鍵攻撃の立場から考察を行う。CLEFIA の自己評価書に於いて、これらの攻撃耐性が議論されており、評価項目によっては、自己評価書の記述の追試を行い、他の項目では、新しい手法で評価すると共に、自己評価書の主張の妥当性を確認した。

以下に本報告書の構成について述べる。第２章では、対象暗号である CLEFIA の構造をまとめ、第３章では、差分攻撃耐性を評価し、第４章では線形攻撃耐性に対し確認し、第５章では、不能差分攻撃、第６章では高階差分攻撃（飽和攻撃）耐性を評価し、第７章では、補間攻撃及び代数攻撃を議論する。第３～７章では、CLEFIA のデータランダム化部の強度を議論するが、第８章では、関連鍵攻撃を意識し、鍵処理部の特性について議論する。最後に本報告書に関る学会発表論文を資料として添付する.

第2章　CLEFIAの構造

共通鍵ブロック暗号 CLEFIA は、4-way 一般化 Feistel 構造のデータ攪拌部と、4(又は 8)-way 一般化 Feistel 構造の鍵処理部を持つ。以下の章での説明の補助として、CLEFIA 仕様書より該当部分を示す。

2.1　データ攪拌部

図 2.1 に CLEFIA のデータ攪拌部を示す。CLEFIA は 4 系列の一般化 Feistel 構造で、1 ラウンドで F_0 及び F_1 の 2 種類の F 関数が並列に配置されている。CLEFIA のデータ攪拌部では 128bit の平文 $X_0^{(0)} \parallel X_1^{(0)} \parallel X_2^{(0)} \parallel X_3^{(0)}$ と 32bit のラウンド鍵 $2r$ 個（$\mathrm{RK}_0, \cdots, \mathrm{RK}_{2r-1}$）及びホワイトニング鍵 4 個（$\mathrm{WK}_0, \cdots, \mathrm{WK}_3$）から 128bit の暗号文を生成する。なお、ラウンド数 r は鍵長 128，192，256bit それぞれで 18，22，26 である。

ラウンド関数 F は、図 2.2 に示す F_0，F_1 の 2 種類が使われる。図 2.2 の F_0，F_1 関数において、S_0，S_1 は 8 ビット入出力の S-box であり、2 つの行列 M_0，M_1 は次のように定義される。

$$M_0 = \begin{pmatrix} 0\mathrm{x}01 & 0\mathrm{x}02 & 0\mathrm{x}04 & 0\mathrm{x}06 \\ 0\mathrm{x}02 & 0\mathrm{x}01 & 0\mathrm{x}06 & 0\mathrm{x}04 \\ 0\mathrm{x}04 & 0\mathrm{x}06 & 0\mathrm{x}01 & 0\mathrm{x}02 \\ 0\mathrm{x}06 & 0\mathrm{x}04 & 0\mathrm{x}02 & 0\mathrm{x}01 \end{pmatrix}, \tag{2.1}$$

$$M_1 = \begin{pmatrix} 0\mathrm{x}01 & 0\mathrm{x}08 & 0\mathrm{x}02 & 0\mathrm{x}0a \\ 0\mathrm{x}08 & 0\mathrm{x}01 & 0\mathrm{x}0a & 0\mathrm{x}02 \\ 0\mathrm{x}02 & 0\mathrm{x}0a & 0\mathrm{x}01 & 0\mathrm{x}08 \\ 0\mathrm{x}0a & 0\mathrm{x}02 & 0\mathrm{x}08 & 0\mathrm{x}01 \end{pmatrix}. \tag{2.2}$$

M_0 と M_1 の最小分岐数は 5 であり、DSM（Diffusion　Swiching Mechanism) による活性 S-box 数の速やかな増加を期待し、行列 $(M_0 \parallel M_1)$ の最小分岐数も 5 である。なお、これらの行列とベクトル間で実行される乗算は、原始多項式 $z^8 + z^4 + z^3 + z^2 + 1$ で定義される GF(2^8) 上の演算である。

2.2　鍵スケジュール部

鍵スケジュール部における中間鍵 L の生成は CLEFIA の鍵長が 128bit のときの鍵処理部は鍵ホワイトニングのないラウンド数 r' が 12 のデータ攪拌部と同様である。ただし、ラウンド鍵は定数である。また、鍵長 192,256bit のときは 8 系列の一般化 Feistel 構造により中間鍵 L を生成する。なお、鍵長 192，256bit のときのラウンド数 r' は 10 である。図 2.3 に鍵長 192，256bit の鍵処理部（以下、「$GFN_{8,10}$」という。）を示す。ホワイトニング鍵 WK_i $(0 \leq i < 4)$ 及びラウンド

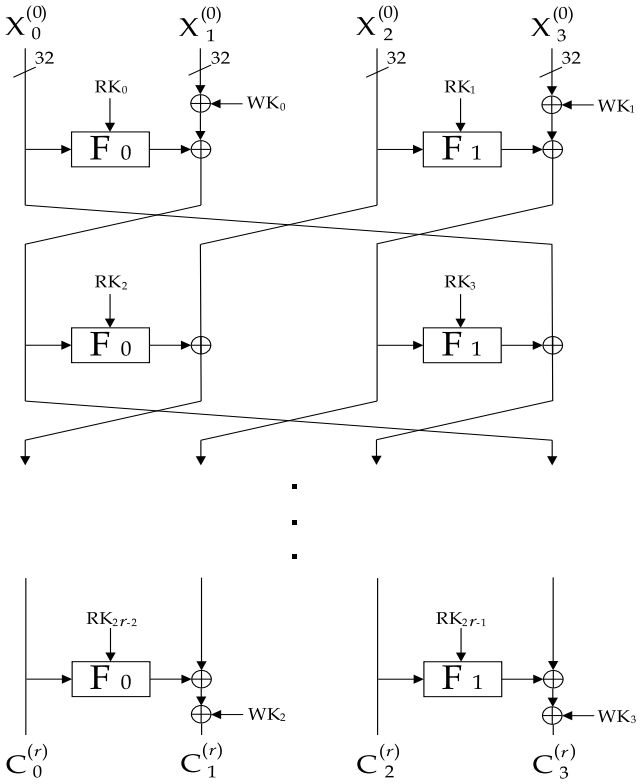
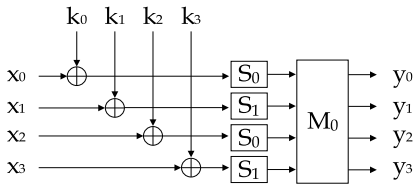
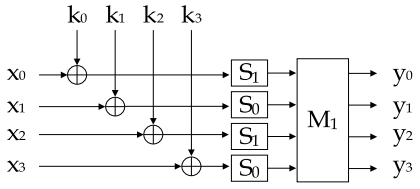


図 2.1: CLEFIA のデータ攪拌部



(a) F_0関数



(b) F_1関数

図 2.2: F_0, F_1 関数

鍵 RK_j $(0 \leq j < 2r)$ は秘密鍵 K 及び中間鍵 L を使用し、生成する。なお、その細部についての説明は省略する。

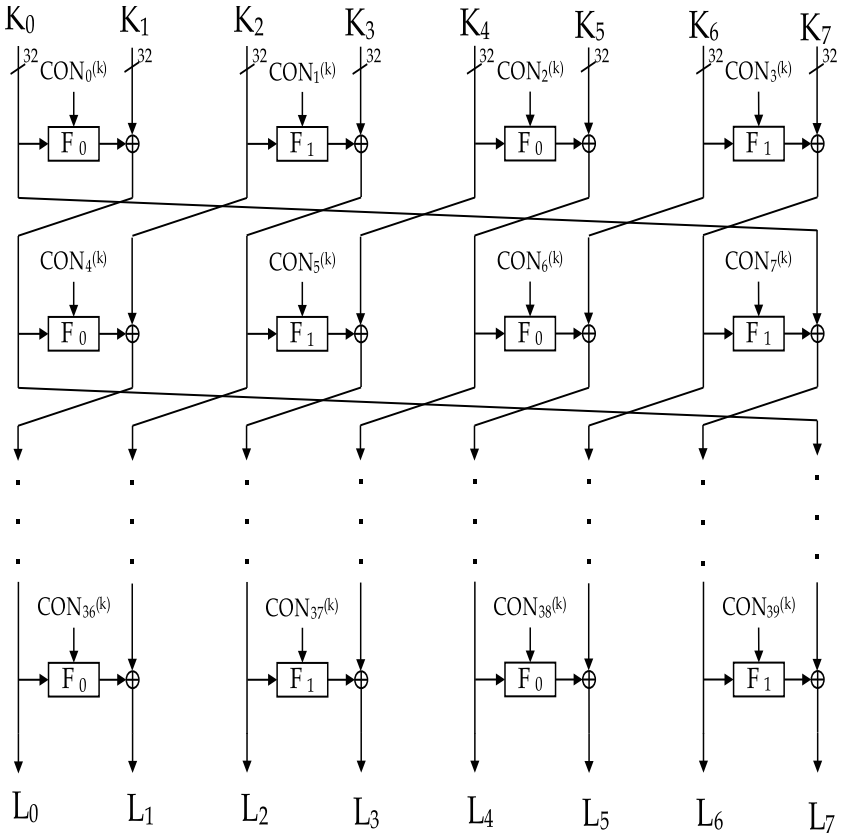


図 2.3: $GFN_{8,10}$

第3章　CLEFIAの差分攻撃耐性

自己評価書 [1] では差分攻撃耐性指標である最大差分特性確率の上界が報告されており、この上界は CLEFIA の特徴である拡散行列切り替え法 (DSM) に着目して導出されている。しかしながら、 CLEFIA のもう一つの特徴である異なる 2 種類の S-box が使われているという点には着目されていない。ここではこれら 2 つの特徴に着目して、自己評価書よりも精密な最大差分特性確率の上界を導出する。結果として 128,192,256bit 秘密鍵の CLEFIA の場合、自己評価書での上界はそれぞれ $2^{-205.48}$,$2^{-256.85}$,$2^{-303.55}$ であるが、我々の評価法ではこの上界が $2^{-227.42}$,$2^{-282.78}$,$2^{-338.46}$ となることを示す。[14]

3.1　差分攻撃

差分攻撃とは、差分伝搬を観測することによりある入力差分がどのような出力差分に高確率で伝搬するかどうかを探索する攻撃方法である。ここでは差分攻撃とその耐性を考える際に必要な事柄をまとめる。

3.1.1　差分確率

関数 $f(x)$ に対して、入力差分 Δx と出力差分 Δy が与えられたとき、差分確率 DP_f は次式のように定義される。ここで n は x のビット長を示す。

$$DP_f(\Delta x, \Delta y) = \frac{\sharp\{x \in \{0,1\}^n \mid f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \tag{3.1}$$

差分攻撃に対する関数 $f(x)$ の強度は、次式で定義される最大差分確率 DP_{max} で評価され、確率が小さいほど強度が強い。

$$DP_{max} = \max_{\Delta x \neq 0, \Delta y} DP_f(\Delta x, \Delta y) \tag{3.2}$$

3.1.2　最大差分特性確率

暗号化関数全体に対しても、式 3.2 を用いて最大差分確率を計算して強度指標とすることが正確な評価となるが、それは計算量の問題で困難である。その場合最大差分特性確率を強度指標とする。$f(x)$ が R ラウンド繰り返される暗号系では、i 段目の入力差分を Δx_i、出力差分を Δx_{i+1} としたとき、最大差分特性確率 DCP_{max} は以下の式で定義される。

$$DCP_{max} = \max_{\substack{\Delta x_0 \neq 0 \\ \Delta x_1, \cdots, \Delta x_R}} \prod_{i=0}^{R-1} DP_{f_i}(\Delta x_i, \Delta x_{i+1}) \tag{3.3}$$

ここで途中段の差分の伝搬状況 $\Delta_{x_0} \to \Delta_{x_1} \to \cdots \to \Delta_{x_R}$ を差分パスという。

3.1.3 truncate 差分解析

実際に最大差分特性確率を求めることも計算量的に困難である場合、truncate 差分解析を用いて、最大差分特性確率の上界である最大 truncate 差分確率 DCP_{Tmax} を求める。この手法は、複数 bit の差分の有無を 1bit で表す。すなわち差分有の場合”1”、無の場合を”0”と表記し、”1”の差分を active 差分と呼ぶ。Sbox の bit 幅である 8bit の truncate 解析を行う場合,4 バイト差分はバイト単位に差分の有無を考えるので次式のように 4 ビットで表記される。

$$\Delta = (\Delta_0, \Delta_1, \Delta_2, \Delta_3) \tag{3.4}$$

差分の伝播は分岐においては (図 3.1) で示され、排他的論理和においては (図 3.2) のいずれかで示される。図中の太線は active 差分が伝播しているパスを示している。ここで

$$\Delta' = (\Delta'_0, \Delta'_1, \Delta'_2, \Delta'_3) \tag{3.5}$$

$$\Delta'' = (\Delta''_0, \Delta''_1, \Delta''_2, \Delta''_3) \tag{3.6}$$

とすれば truncate 差分ベクトルの成分毎に次式の演算規則が適用される。

$$\Delta'_i = \Delta_i \bar{\oplus} \Delta''_i \tag{3.7}$$

ただし、$\bar{\oplus}$ は truncate 差分としての排他的論理和であり、表 3.1 に従う。

表 3.1: truncate 差分の排他的論理和

$\bar{\oplus}$	0	1
0	0	1
1	1	0or1

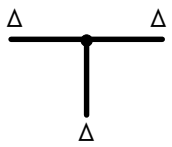


図 3.1: 分岐における差分の伝播

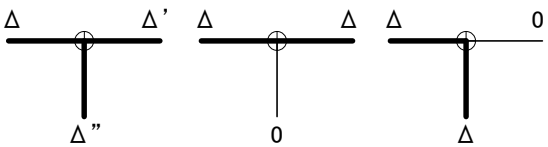


図 3.2: 排他的論理和における差分の伝播

3.1.4 active Sbox

Sbox の入力差分が active であれば、その Sbox を active Sbox と呼ぶ。以降、ラウンド t の処理で生じる active Sbox の数を $as^{(t)}$ と書き、active Sbox 数の合計を AS(=$\Sigma as^{(t)}$) と表す。さらに AS のうち最小のものを AS_{min} と表す。

3.2 CLEFIA の差分解析

ここでは、CLEFIA の差分攻撃耐性を評価するために最大差分特性確率の上界を 8bit truncate 差分解析により導出するアルゴリズムを説明し、解析結果を示す。

3.2.1 Sbox

2 種類の Sbox、S_0, S_1 の最大差分確率を計算すると

$$S_0 : DP_{max} = 2^{-4.67} \tag{3.8}$$

$$S_1 : DP_{max} = 2^{-6.00} \tag{3.9}$$

となった。これは自己評価書と同一の結果である。S_0, S_1 それぞれについて DP_{max} を与える入出力差分の例を表 3.2, 3.3 に示す。

表 3.2: $S_0 : DP_{\max}$ を与える入出力差分

Δ_x	0x97	0xE2	0xE3	0x32	0xC6	0xF3	0x8
Δ_y	0xA	0xD	0x1E	0x20	0x30	0x7D	0x7E

表 3.3: $S_1 : DP_{\max}$ を与える入出力差分

Δ_x	0xA6	0xBE	0xEA	0xA2	0xFB	0xA9	0x2
Δ_y	0x14	0x1	0x13	0x2	0x3	0xFF	0xE9

3.2.2 F_i 関数の差分特性

F_i 関数の最大 truncate 差分確率 DP_{FTmax} を評価するために、分岐数条件と 2 種類の Sbox の最大差分確率の最大差分確率 ($S_0 : DP = 2^{-4.67}, S_1 : DP = 2^{-6.00}$) を与える出力差分を用いて実際に計算機探索を行った。その結果、存在しえない値があったため S_0 に関しては次点の確率 $DP = 2^{-5.00}$ を考慮して評価した。この結果を表 3.4, 3.5 に示す。なお、DP_{FTmax} は $(2^{-4.67})^{AS_0'} \times (2^{-5.00})^{AS_0''} \times (2^{-6.00})^{AS_1}$ と表すことができる。ここで、(AS_0', AS_0'', AS_1) はそれぞれ差分確率 $(2^{-4.67}, 2^{-5.00}, 2^{-6.00})$ を与える active Sbox 数である。例として、F_0 の (入力バイト差分)=0xB,(出力バイト差分)=0x3 の場合、DP_{FTmax} は $(AS_0' = 2, AS_1 = 1)$ より $2^{-15.34}$ となるが、実際の DP_{FTmax} は $(AS_0' = 1, AS_0'' = 1, AS_1 = 1)$ より $2^{-15.67}$ となる。但し、表中の数値は $(\log_2 DP_{FTmax})$ を示す。表中の横線はそのような入出力差分がないパターンを示しており、最小分岐数の条件から存在しえないパスである。

表 3.4: F_0 の差分確率の上界 [$\log_2$]

		出力バイト差分															
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
入力バイト差分	0x0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x3	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x5	-	-	-	-	-	-	-	-12	-	-	-	-12	-	-12	-12	-12
	0x6	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x7	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0x8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x9	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xA	-	-	-	-	-	-	-	-9.67	-	-	-	-9.34	-	-9.67	-9.34	-9.34
	0xB	-	-	-	-15.67	-	-15.67	-15.34	-15.34	-	-15.34	-15.67	-15.34	-15.34	-15.34	-15.34	-15.34
	0xC	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xD	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0xE	-	-	-	-15.34	-	-15.67	-15.34	-15.34	-	-15.34	-15.67	-15.34	-15.67	-15.34	-15.34	-15.34
	0xF	-	-21.34	-21.67	-21.34	-21.34	-21.34	-21.34	-21.34	-21.67	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34

表 3.5: F_1 の差分確率の上界 [$\log_2$]

		出力バイト差分															
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
入力バイト差分	0x0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x3	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x5	-	-	-	-	-	-	-	-9.67	-	-	-	-9.67	-	-9.67	-9.67	-9.34
	0x6	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x7	-	-	-	-15.67	-	-15.67	-15.34	-15.34	-	-15.67	-15.67	-15.34	-15.67	-15.34	-15.34	-15.34
	0x8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x9	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xA	-	-	-	-	-	-	-	-12	-	-	-	-12	-	-12	-12	-12
	0xB	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0xC	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xD	-	-	-	-15.67	-	-15.67	-15.67	-15.34	-	-15.34	-15.67	-15.34	-15.67	-15.34	-15.34	-15.34
	0xE	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0xF	-	-21.67	-21.67	-21.34	-21.67	-21.34	-21.34	-21.34	-21.67	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34

3.2.3 CLEFIA の truncate 差分解析

解析方法

データ処理部における非線形関数は Sbox のみであることから、最大差分特性確率の上界を求めるためには、すべての差分パスに対する DCP_{Tmax} を求めればよい。そこでまず、F_i 関数の入出力差分対に対し差分確率の最大値を求める。

その値を使い CLEFIA 全体の差分パスに対する特性確率の最大値を探索アルゴリズムである Viterbi アルゴリズムで導出する。

探索アルゴリズム

差分パス及び最大差分特性確率の上界を探索する手法として Viterbi アルゴリズムを用いる。各段の入力を状態と考え、トレリス線図を用いて状態遷移を解析することにより、DCP_{Tmax} とその差分パスを導出できる。解析で用いる状態及び状態遷移を図 3.3 を用いて説明する。各ラウンドにおける F 関数の 8 ビット truncate 入力差分を $\Delta\mathscr{X}_i$ とし、$\Delta\mathscr{X}_i$ における active バイト数を D_i と表す。$\Delta\mathscr{X}_i$ は 4 ビットベクトルであり D_i は $(0 \leq D_i \leq 4)$ の値を持つ。

状態遷移は初めの 2 段の状態変数である次式

$$st(0) = (\Delta\mathscr{X}_0, \Delta\mathscr{X}_1, \Delta\mathscr{X}_2, \Delta\mathscr{X}_3) \tag{3.10}$$

からスタートする。ここではこれをラウンド 0 の状態と呼ぶ。次のラウンド $t = 1$ では状態変数として次式をとる。

$$st(1) = (\Delta\mathscr{X}_2, \Delta\mathscr{X}_3, \Delta\mathscr{X}_4, \Delta\mathscr{X}_5, D_0, D_1) \tag{3.11}$$

$st(0)$ から $st(1)$ への可能な遷移は節 3.1.3 で述べた truncate 差分伝播規則に従って決定される。また、D_0、D_1 は次節に述べる DSM の効果を状態遷移に反映させるためのものである。さらに次のラウンド $t = 2$ では次式を状態変数にとる。

$$st(2) = (\Delta\mathscr{X}_4, \Delta\mathscr{X}_5, \Delta\mathscr{X}_6, \Delta\mathscr{X}_7, D_0, D_1, D_2, D_3) \tag{3.12}$$

同様に D_2、D_3 は DSM の効果を反映したものである。$t \geq 2$ においては次式の状態変数をとる。

$$\begin{aligned} st(t) = (&\Delta\mathscr{X}_{2t}, \Delta\mathscr{X}_{2t+1}, \Delta\mathscr{X}_{2t+2}, \Delta\mathscr{X}_{2t+3}, \\ &D_{2t-4}, D_{2t-3}, D_{2t-2}, D_{2t-1}) \end{aligned} \tag{3.13}$$

拡散行列 M(DSM) の取り扱い

ここでは CLEFIA の拡散行列の MDS 特性を viterbi 探索にいかに反映させるかを述べる。

M_0, M_1 の最小分岐数は、M_0, M_1 の入出力における非零バイト差分の数の総和のうち、最小のものを指す。ただし、入力バイト差分がオールゼロの場合を除く。本解析で用いる $(M_0), (M_1), (M_0|M_1)$ の最小分岐数はいずれの場合も 5 である。ここで $(M_0|M_1)$ は行列 (M_0) と (M_1) の連結である 4 行 8 列の行列である。このような拡散行列が使われている場合、次のような性質[1]が成り立つ。

[1] 自己評価書 [1], p.24, 性質 2.2、2,3

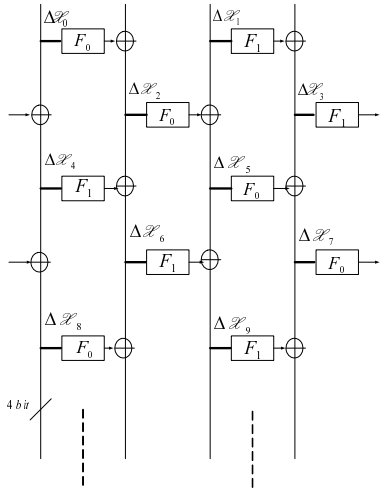


図 3.3: 等価変形したデータ処理部の構造

- 性質 1

  $t = 2a - 1 (a \geq 1)$ のとき
  $D_{2t+2} + D_{2t+1} + D_{2t-2} \geq 5 \ \ (D_{2t+1} \neq 0)$
  $D_{2t+3} + D_{2t} + D_{2t-1} \geq 5 \ \ (D_{2t} \neq 0)$

  $t = 2a (a \geq 1)$ のとき
  $D_{2t+2} + D_{2t} + D_{2t-2} \geq 5 \ \ (D_{2t} \neq 0)$
  $D_{2t+3} + D_{2t+1} + D_{2t-1} \geq 5 \ \ (D_{2t+1} \neq 0)$

- 性質 2

  $t = 2a - 1 (a \geq 2)$ のとき
  $D_{2t+2} + D_{2t+1} + D_{2t-3} + D_{2t-6} \geq 5 \ \ (D_{2t+1} + D_{2t-3} \neq 0)$
  $D_{2t+3} + D_{2t} + D_{2t-4} + D_{2t-5} \geq 5 \ \ (D_{2t} + D_{2t-4} \neq 0)$

  $t = 2a (a \geq 2)$ のとき
  $D_{2t+2} + D_{2t} + D_{2t-4} + D_{2t-6} \geq 5 \ \ (D_{2t} + D_{2t-4} \neq 0)$
  $D_{2t+3} + D_{2t+1} + D_{2t-3} + D_{2t-5} \geq 5 \ \ (D_{2t+1} + D_{2t-3} \neq 0)$

性質 1 は F 関数単体の最小分岐数特性に関するものであり、すでに節 3.2.2 で述べた F 関数差分特性のデータとして組み込まれている。$st(t)$ から $st(t+1)$ の状態遷移では $st(t)$ が truncate 差分 $\Delta\mathscr{X}_{2t}, \Delta\mathscr{X}_{2t+1}, \Delta\mathscr{X}_{2t+2}, \Delta\mathscr{X}_{2t+3}$ を状態として持っているので性質 1 の代りに節 3.2.2 の DP_{FTmax} を用いる。探索ではこれと図 3.1, 3.2 に示した差分の伝播条件、性質 2 を満足するパスの中から最良パスを探索する。

例えば、$st(5) = (\Delta\mathscr{X}_{10}, \Delta\mathscr{X}_{11}, \Delta\mathscr{X}_{12}, \Delta\mathscr{X}_{13}, D_6, D_7, D_8, D_9)$
=(0xB,0x7,0x1,0xA,0x0,0x0,0x4,0x2,0x1) のとき $\Delta\mathscr{X}_{14}$ は $(\Delta\mathscr{X}_{14} = F(\Delta\mathscr{X}_{12}) \bar{\oplus} \Delta\mathscr{X}_{10})$ で表される。但し $F(\Delta\mathscr{X}_{12})$ は F 関数の出力バイト差分となる。ここで節 3.1.3 で記した伝播条件から (0x4, 0x5, 0x6, 0x7, 0xC, 0xD, 0xE, 0xF) の 8 通りの差分が存在することがわかる。しかしながら、性質 2 より式 (3.14) を満たさない差分は伝播しないため実際に伝搬する差分は (0x5,0x6,0x7,0xC,0xD,0xE,0xF) の 7 通りとなる。

$$D_{14} \geq (D_{12} + D_8 + D_6) = 2 \tag{3.14}$$

表 3.6: 本解析と自己評価書での解析による DCP_{Tmax} [$\log_2$] の比較

r	AS_{min}	自己評価書	本稿
1	0	0	0
2	1	-4.67	-4.67
3	2	-9.34	-10.67
4	6	-28.02	-32.01
5	8	-37.36	-41.35
6	12	-56.04	-62.69
7	14	-65.38	-73.69
8	18	-84.06	-92.37
9	20	-93.40	-102.04
10	22	-102.74	-113.71
11	24	-112.08	-123.05
12	28	-130.76	-144.39
13	30	-141.10	-155.06
14	34	-158.78	-176.40
15	36	-168.12	-185.07
16	38	-177.46	-196.41
17	40	-186.80	-206.74
18	44	-205.48	-227.42
19	46	-214.82	-237.42
20	50	-233.50	-256.77
21	52	-242.84	-269.43
22	55	-256.85	-282.78
23	56	-261.52	-290.10
24	59	-275.53	-306.45
25	62	-289.54	-320.78
26	65	-303.55	-338.46

最大差分特性確率の上界

節 3.2.3 の探索アルゴリズムを適用し計算機で解析した結果、128bit CLEFIA では $DCP_{Tmax}^{18round} = 2^{-227.42}$、192bit CLEFIA では $DCP_{Tmax}^{22round} = 2^{-282.78}$、256bit CLEFIA では $DCP_{Tmax}^{26round} = 2^{-338.46}$ となった。表 3.6 に AS_{min} を示し、その確率を自己評価書と本稿とを比較する。なお、確率は $\log_2(DCP_{Tmax})$ で示されている。自己評価書では S_0 の DCP_{Tmax} のみを考慮した確率であり、本稿では S_0、S_1 の確率の違いを評価したものである。また、DCP_{Tmax} を与えるパスにそって active Sbox 数をカウントしたものが AS_{min} の欄に示してある。結果として、自己評価書と同じ個数の最小 Sbox 数となっている。表 3.7 は本解析による 18 ラウンドに渡る差分パスの一例である。

3.3 まとめ

ここでは CLEFIA の 2 種類の Sbox と DSM を考慮し、差分攻撃耐性評価を行った。Viterbi アルゴリズムを用いて truncate 差分パスを探索し、最大 truncate 差分特性確率の上界を導出した。その結果、1 種類の Sbox のみを考慮した自己評価書の評価と比べ、2 種類の Sbox を考慮した本評価では上界値が低下した。これは本評価が自己評価書よりも精密であることを示している。これにより、CLEFIA の差分攻撃に対するセキュリティーマージンが増加した。

表 3.7: 本解析による 18 ラウンドに渡る差分パスの一例

t	F 関数の入力差分
1	$(\Delta\mathscr{X}_0 \vert \Delta\mathscr{X}_1) = 0\text{x}01$
2	$(\Delta\mathscr{X}_2 \vert \Delta\mathscr{X}_3) = 0\text{x}00$
3	$(\Delta\mathscr{X}_4 \vert \Delta\mathscr{X}_5) = 0\text{x}01$
4	$(\Delta\mathscr{X}_6 \vert \Delta\mathscr{X}_7) = 0\text{x}0\text{f}$
5	$(\Delta\mathscr{X}_8 \vert \Delta\mathscr{X}_9) = 0\text{x}b1$
6	$(\Delta\mathscr{X}_{10} \vert \Delta\mathscr{X}_{11}) = 0\text{x}67$
7	$(\Delta\mathscr{X}_{12} \vert \Delta\mathscr{X}_{13}) = 0\text{x}0\text{b}$
8	$(\Delta\mathscr{X}_{14} \vert \Delta\mathscr{X}_{15}) = 0\text{x}60$
9	$(\Delta\mathscr{X}_{16} \vert \Delta\mathscr{X}_{17}) = 0\text{x}00$
10	$(\Delta\mathscr{X}_{18} \vert \Delta\mathscr{X}_{19}) = 0\text{x}60$
11	$(\Delta\mathscr{X}_{20} \vert \Delta\mathscr{X}_{21}) = 0\text{x}0\text{b}$
12	$(\Delta\mathscr{X}_{22} \vert \Delta\mathscr{X}_{23}) = 0\text{x}6\text{a}$
13	$(\Delta\mathscr{X}_{24} \vert \Delta\mathscr{X}_{25}) = 0\text{x}b5$
14	$(\Delta\mathscr{X}_{26} \vert \Delta\mathscr{X}_{27}) = 0\text{x}07$
15	$(\Delta\mathscr{X}_{28} \vert \Delta\mathscr{X}_{29}) = 0\text{x}05$
16	$(\Delta\mathscr{X}_{30} \vert \Delta\mathscr{X}_{31}) = 0\text{x}00$
17	$(\Delta\mathscr{X}_{32} \vert \Delta\mathscr{X}_{33}) = 0\text{x}05$
18	$(\Delta\mathscr{X}_{34} \vert \Delta\mathscr{X}_{35}) = 0\text{x}07$

第4章　線形攻撃

ここでは CLEFIA の線形攻撃耐性指標として、S-box の最大線形確率とデータ処理部における active S-box 数の最小値を解析し、自己評価書における値と比較する。自己評価書では、線形攻撃耐性指標である最大線形特性確率の上界が報告されている。この上界は CLEFIA の特徴である拡散行列切り替え法 (DSM) に着目して導出されているが、CLEFIA のもう一つの特徴である異なる 2 種類の Sbox が使われているという点には着目されていない。本稿ではこれら二つの特徴に着目し、自己評価書よりも精密な最大線形特性確率の上界を導出する。

4.1　線形確率と線形特性確率

線形攻撃は松井によって提案されたブロック暗号に対する汎用的な攻撃であり、関数入出力の線形相関の偏りを利用する。ここでは、線形攻撃の耐性指標である線形確率と線形特性確率について述べる。

n bit 入出力の関数 $f(x)$ に対して、入力マスク Γ_x と出力マスク Γ_y が与えられたとき、$f(x)$ の線形確率 $LP(\Gamma_x, \Gamma_y)$ は次式で定義される。

$$LP(\Gamma_x, \Gamma_y) = (2 \cdot \frac{\#\{x \in \{0,1\}^n \mid x \bullet \Gamma_x = y \bullet \Gamma_y\}}{2^n} - 1)^2 \tag{4.1}$$

ここで、$\#$ は線形近似式 $(x \bullet \Gamma_x = y \bullet \Gamma_y)$ の成立回数を表し、$\bullet$ は GF(2) 上の内積演算を表す。さらに最大線形確率 LP_{max} は次式で与えられ、LP_{max} が小さいほどその関数は線形攻撃に対する強度が高い。

$$LP_{max} = \max_{\Gamma x, \Gamma y \neq 0} LP(\Gamma x, \Gamma y) \tag{4.2}$$

暗号化関数に対しても、式 (4.2) を用いて線形攻撃耐性を評価することが望ましいが、それは計算量の問題で困難である場合が多い。その場合、一般的には次に示す最大線形特性確率を強度指標とする。

関数 $f(x)$ が R ラウンド繰り返される暗号系では i 段目の入力マスクを Γ_{x_i}、出力マスクを $\Gamma_{x_{i+1}}$ としたとき、最大線形特性確率 LCP_{max} は、各ラウンドの線形確率 $LP(\Gamma_{x_i}, \Gamma_{x_{i+1}})$ の積として次式で与えられる。

$$LCP_{max} = \max_{\substack{\Gamma_{x_0}, \Gamma_{x_1}, \cdots, \Gamma_{x_R} \\ \Gamma_{x_i} \neq 0}} \prod_{i=0}^{R-1} LP(\Gamma_{x_i}, \Gamma_{x_{i+1}}) \tag{4.3}$$

ここでマスクの伝搬状況 $\Gamma_{x_0} \to \Gamma_{x_1} \to \Gamma_{x_2} \to \cdots \to \Gamma_{x_R}$ を線形パスという。

4.2 truncate 線形解析

truncate 線形解析とは、複数 bit のマスクの有無を 1bit で表現し、その 1bit の情報の伝播を解析するものである。マスクが非ゼロの場合は"1" と表記し active と呼ぶ、一方、マスクがゼロの場合は"0" と表記し non-acive 或いは passive と呼ぶ。Sbox の bit 幅である 8bit の truncate 解析を行う場合、1 ワード (4 バイト) のマスクは次式のように 4 ビットのマスク Γ で表される。

$$\Gamma = (\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3), \quad \Gamma_i \in \mathrm{GF}(2). \tag{4.4}$$

マスクの伝播は分岐においては図 4.1 のいずれかで表され、排他的論理和においては図 4.2 で表される。図中の太線は active マスクが伝播しているパスを示している。データの排他的論理和 $\oplus$ においてはマスク Γ は、コピーされ、同一のマスク Γ が伝わって行く。図 4.1 で示されるデータの分岐においては、マスク Γ、Γ'、Γ'' をベクトルとしてその要素に対し、表 4.1 で示される truncate マスク Γ_i の XOR 演算規則が適用され、次式となる。

$$\Gamma = \Gamma' \bar{\oplus} \Gamma'' \tag{4.5}$$

また Sbox においては、Sbox の入出力マスクが active であれば、その Sbox を active Sbox と呼ぶ。

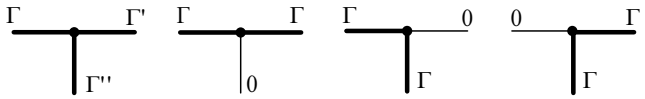


図 4.1: 分岐におけるマスクの伝播

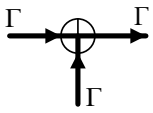


図 4.2: 排他的論理和におけるマスクの伝播

表 4.1: truncate マスクの XOR 演算規則

$\bar{\oplus}$	0	1
0	0	1
1	1	0or1

4.3 CLEFIA の線形解析

4.3.1 Sbox の線形特性

CLEFIA の S-box S_0 と S_1 の最大線形確率について式 (4.2) により解析した結果、 S_0 の場合 $LP_{max} = 2^{-4.38529}$ となり、S_1 の場合 $LP_{max} = 2^{-6.00}$ となった。自己評価書ではそれぞれ

$LP_{max} = 2^{-4.38}$, $LP_{max}^{S_1} = 2^{-6.00}$ である。S_0 について、LP_{max} を与える (Γ_x, Γ_y) は 52 組あり、それを表 4.2 に示す。S_1 について LP_{max} を与える (Γ_x, Γ_y) は 1275 組あり、その一部を表 4.3 に示す。

表 4.2: S_0 について最大線形確率を与えるマスクの組 (Γ_x, Γ_y)

(0x1,0xdd)	(0x2,0x7c)	(0x3,0x6a)	(0x3,0xe6)	(0x7,0x7f)
(0xb,0xe7)	(0x1c,0xac)	(0x1d,0x80)	(0x2a,0x7d)	(0x2b,0xec)
(0x35,0x6)	(0x35,0x6c)	(0x3e,0xa0)	(0x40,0xd1)	(0x49,0x6f)
(0x4b,0xfe)	(0x53,0x90)	(0x54,0x49)	(0x5c,0x7a)	(0x5f,0xc1)
(0x69,0xe2)	(0x6d,0xa2)	(0x75,0x9)	(0x79,0xe0)	(0x7f,0x2b)
(0x81,0x59)	(0x85,0x11)	(0x8c,0xd2)	(0x95,0x20)	(0x9d,0xa8)
(0xa1,0x90)	(0xa8,0xc7)	(0xa8,0xf1)	(0xb4,0xc0)	(0xb9,0x23)
(0xc0,0x39)	(0xc3,0x16)	(0xc4,0xbe)	(0xc9,0x50)	(0xd8,0x31)
(0xdd,0x2c)	(0xe5,0x68)	(0xe6,0x6e)	(0xed,0x30)	(0xee,0x5)
(0xee,0x19)	(0xef,0x55)	(0xf5,0x38)	(0xf7,0x66)	(0xf7,0xb0)
(0xfc,0x5f)	(0xff,0x22)			

表 4.3: S_1 について最大線形確率を与えるマスクの組 (Γ_x, Γ_y) の一部

(0x1,0x1)	(0x1,0x10)	(0x1,0x11)	(0x1,0x88)	(0x1,0x98)
(0x2,0x3)	(0x2,0x10)	(0x2,0x13)	(0x2,0x88)	(0x2,0x9b)
(0x3,0x2)	(0x3,0x10)	(0x3,0x8a)	(0x3,0x98)	(0x3,0x9a)
(0x4,0xc)	(0x4,0x42)	(0x4,0x4e)	(0x4,0xa0)	(0x4,0xee)
(0x5,0xd)	(0x5,0x47)	(0x5,0x4a)	(0x5,0x64)	(0x5,0x69)
(0x6,0x9)	(0x6,0x62)	(0x6,0xb5)	(0x6,0xbc)	(0x6,0xde)
(0x7,0x29)	(0x7,0x30)	(0x7,0xc9)	(0x7,0xe0)	(0x7,0xf9)
(0x8,0x5c)	(0x8,0x85)	(0x8,0xa1)	(0x8,0xd9)	(0x8,0xfd)
(0x9,0x44)	(0x9,0x77)	(0x9,0x84)	(0x9,0xc0)	(0x9,0xf3)
(0xa,0x18)	(0xa,0x2c)	(0xa,0x34)	(0xa,0xdd)	(0xa,0xe9)
(0xb,0x20)	(0xb,0x28)	(0xb,0x52)	(0xb,0x72)	(0xb,0x7a)
(0xc,0x2)	(0xc,0x10)	(0xc,0x11)	(0xc,0x12)	(0xc,0x13)
(0xd,0x1)	(0xd,0x10)	(0xd,0x8b)		

4.3.2 F_i 関数の線形特性

次に Sbox の線形確率の解析結果を用いて、F_i 関数の線形確率の上界を解析した。その結果を表 4.4, 4.5 に示す。表中の数値は、入出力バイトマスクを与えたときの線形確率の最大値の対数 $log_2 LP_{Fmax}$ を示す。例をあげれば入力バイトマスク 0x03, 出力バイトマスク 0x07 に対する F_i の線形確率の最大値は $2^{-10.38}$ である。また、横線は、そのような入出力バイトマスクは、最小分岐

数の条件から存在しえないパスである事を示す。この表から、F_i 関数において存在する線形パスは、全て S-box の最大線形確率を使って接続可能で有る事がわかる。

表 4.4: F_0 の線形確率の上界 [$\log_2$]

		出力バイトマスク															
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
入力バイトマスク	0x0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.38
	0x3	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0x4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x5	-	-	-	-	-	-	-	-12	-	-	-	-12	-	-12	-12	-12
	0x6	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0x7	-	-	-	-16.38	-	-16.38	-16.38	-16.38	-	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38
	0x8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.38
	0x9	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0xA	-	-	-	-	-	-	-	-8.76	-	-	-	-8.76	-	-8.76	-8.76	-8.76
	0xB	-	-	-	-14.76	-	-14.76	-14.76	-14.76	-	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76
	0xC	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0xD	-	-	-	-16.38	-	-16.38	-16.38	-16.38	-	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38
	0xE	-	-	-	-14.76	-	-14.76	-14.76	-14.76	-	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76
	0xF	-	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76

表 4.5: F_1 の線形確率の上界 [$\log_2$]

		出力バイトマスク															
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
入力バイトマスク	0x0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.38
	0x2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x3	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0x4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.38
	0x5	-	-	-	-	-	-	-	-8.76	-	-	-	-8.76	-	-8.76	-8.76	-8.76
	0x6	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0x7	-	-	-	-14.76	-	-14.76	-14.76	-14.76	-	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76
	0x8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x9	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0xA	-	-	-	-	-	-	-	-12	-	-	-	-12	-	-12	-12	-12
	0xB	-	-	-	-16.38	-	-16.38	-16.38	-16.38	-	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38
	0xC	-	-	-	-	-	-	-	-10.38	-	-	-	-10.38	-	-10.38	-10.38	-10.38
	0xD	-	-	-	-14.76	-	-14.76	-14.76	-14.76	-	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76	-14.76
	0xE	-	-	-	-16.38	-	-16.38	-16.38	-16.38	-	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38	-16.38
	0xF	-	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76	-20.76

4.3.3 CLEFIA の truncate 線形解析

解析方法

データ処理部における非線形関数は Sbox のみであることから、最大線形特性確率を求めるには、すべての線形パスに対する LCP_{Tmax} を求めればよい。前節の F_i 関数の truncate 線形確率を使い CLEFIA 全体の線形パスに対する線形特性確率の最大値を探索アルゴリズムである Viterbi アルゴリズムで導出する。

探索アルゴリズム

線形パス及び最大線形特性確率の上界を探索する手法として Viterbi アルゴリズムを用いる。各段の入力を状態と考え、トレリス線図を用いて状態遷移を解析することにより、DCP_{Tmax} とその線形パスを導出できる。解析で用いる状態及び状態遷移を図 4.3 を用いて説明する。各ラウンドにおける F 関数の 8 ビット truncate 入出力マスクを Γx_i,Γy_i とし、Γx_i における active バイト数を L_i と表す。Γx_i,Γy_i は 4 ビットベクトルであり L_i は $0 \leq L_i \leq 4$ の値をもつ。
状態遷移の始めは 2 段の状態変数である次式

$$st(0) = (\Gamma y_0, \Gamma y_1, \Gamma y_2, \Gamma y_3) \tag{4.6}$$

からスタートする。ここでは、これをラウンド 0 の状態と呼ぶことにする。次のラウンド t=1 では状態変数として次式

$$st(1) = (L_0, L_1, L_2, L_3, \Gamma y_2, \Gamma y_3, \Gamma y_4, \Gamma y_5) \tag{4.7}$$

となる。$st(0)$ から $st(1)$ への可能な遷移は 4.2 節で述べた truncate マスク伝播規則に則って決定される。また、L_0, L_1, L_2, L_3 は次節に述べる DSM の効果を状態遷移に反映させるためのものである。$t \geq 1$ では次式の状態変数をとる。

$$st(t) = (L_{2t-2}, L_{2t-1}, L_{2t}, L_{2t+1}, \Gamma y_{2t}, \Gamma y_{2t+1}, \Gamma y_{2t+2}, \Gamma y_{2t+3}) \tag{4.8}$$

拡散行列 M_0, M_1(DSM) の取り扱い

ここでは CLEFIA の 2 種類の拡散行列の DSM 特性を Viterbi 探索にいかに反映させるかを述べる。M_0, M_1 の最小分岐数は、M_0, M_1 の入出力における非零バイトマスクの総数のうち、最小のものを指す。ただし、入力バイトマスクがオール零の場合を除く。本稿で用いる $({}^tM_0^{-1})$,$({}^tM_1^{-1})$, $({}^tM_0^{-1}|{}^tM_1^{-1})$ は、$({}^tM_0^{-1})$=(M_0), $({}^tM_1^{-1})$=(M_1),$({}^tM_0^{-1}|{}^tM_1^{-1})$= $(M_0|M_1)$ の関係があり、最小分岐数はいずれの場合も 5 である。ここで $({}^tM_0^{-1}|{}^tM_1^{-1})$ 及び $(M_0|M_1)$ はそれぞれ、$({}^tM_0^{-1})$ と $({}^tM_1^{-1})$,(M_0) と (M_1) の連結であり 4 行 8 列の行列である。このような拡散行列が使われている

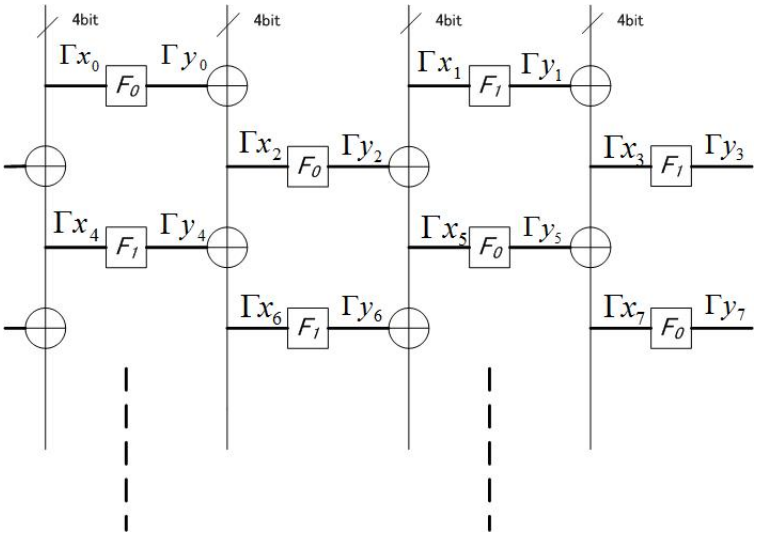


図 4.3: データ処理部の構造

場合、次のような性質が成り立つ。

$$t = 2a - 1 (a \geq 1) \text{ のとき}$$

$$L_{2t-2} + L_{2t} + L_{2t+2} \begin{cases} = 0 \\ \geq 5 \end{cases} \tag{4.9}$$

$$L_{2t-1} + L_{2t+1} + L_{2t+3} \begin{cases} = 0 \\ \geq 5 \end{cases} \tag{4.10}$$

$$t = 2a (a \geq 1) \text{ のとき}$$

$$L_{2t-2} + L_{2t+1} + L_{2t+2} \begin{cases} = 0 \\ \geq 5 \end{cases} \tag{4.11}$$

$$L_{2t-1} + L_{2t} + L_{2t+3} \begin{cases} = 0 \\ \geq 5 \end{cases} \tag{4.12}$$

探索ではこの性質と図 4.1,4.2 に示した差分の伝播条件を満足するパスの中から最良パスを探索する。

最大線形特性確率の上界

4.3.3 の探索アルゴリズムを適用し計算機で解析した結果、128bit CLEFIA では $LCP_{Tmax}^{18round} = 2^{-222.54}$、192bitCLEFIA では $LCP_{Tmax}^{22round} = 2^{-277.38}$、256bitCLEFIA では $LCP_{Tmax}^{26round} = 2^{-331.38}$ となった。表 4.6 に AS 数及び最大線形特性確率を示し、自己評価書と比較する。なお、確率は $log_2(LCP_{Tmax})$ の値で表している。自己評価書では S_0 の LCP_{Tmax} のみを考慮した確率であり、本解析では S_0, S_1 の確率の違いを評価したものである。また、LCP_{Tmax} を与えるパスにそって 2 種類の active Sbox S_0、S_1 を数えたもの及びその合計がそれぞれ AS_0数,AS_1数,AS 数 の欄に示してある。表 4.7,4.8, 4.9,4.10 に本解析結果を与える線形パスの一例を示す。

表 4.6: 段数と active Sbox 数及び LCP_{Tmax} の関係 (記号 $*$ は自己評価書に基づく評価量を表し、記号 † は本解析に基づく評価量を表す。

段数	AS 数 *	AS 数 †	AS_0 数 †	AS_1 数 †	$LCP_{Tmax}{}^*$	$LCP_{Tmax}{}^\dagger$
1	0	0	0	0	0	0
2	1	1	1	0	-4.38	-4.38
3	5	5	5	0	-21.90	-21.90
4	6	6	4	2	-26.28	-29.52
5	10	10	8	2	-43.80	-47.04
6	15	15	13	2	-65.70	-68.94
7	16	16	10	6	-70.08	-79.80
8	18	19	12	7	-78.84	-94.56
9	20	21	14	7	-87.60	-103.32
10	23	23	16	7	-100.74	-112.08
11	26	26	20	6	-113.88	-123.60
12	30	30	25	5	-131.40	-139.50
13	32	33	27	6	-140.16	-154.26
14	34	37	32	5	-148.92	-170.16
15	36	37	24	13	-157.68	-183.12
16	39	40	26	14	-170.82	-197.88
17	42	42	28	14	-183.96	-206.64
18	46	46	33	13	-201.48	-222.54
19	48	49	37	12	-210.24	-234.06
20	50	53	43	10	-219.00	-248.34
21	52	56	46	10	-227.76	-261.48
22	55	60	51	9	-240.90	-277.38
23	58	58	38	20	-254.04	-286.44
24	62	62	42	20	-271.56	-303.96
25	64	65	45	20	-280.32	-317.10
26	66	69	51	18	-289.08	-331.38

4.4 まとめ

本稿では、共通鍵ブロック暗号 CLEFIA の 2 種類の Sbox と DSM を考慮し、線形攻撃耐性評価を行った。評価は 8bit truncate 線形解析で行い、Viterbi アルゴリズムを利用して truncate 線形パスを探索し、最大 truncate 線形特性確率の上界を求めた。その結果、1 種類の Sbox のみを考慮した自己評価書の評価と比較すると、2 種類の Sbox を考慮した本評価では上界値が低下した。よって、本評価は自己評価書よりも精密であることを示すと同時に、CLEFIA は線形攻撃に対して十分な耐性を持つと結論づける。

表 4.7: 12 ラウンドにおける本稿の結果を与えるマスクの一例

t	F 関数の出力マスク
1	$(\Gamma y_0 \| \Gamma y_1)$=0x07
2	$(\Gamma y_2 \| \Gamma y_3)$=0x00
3	$(\Gamma y_4 \| \Gamma y_5)$=0x07
4	$(\Gamma y_6 \| \Gamma y_7)$=0xb0
5	$(\Gamma y_8 \| \Gamma y_9)$=0x57
6	$(\Gamma y_{10} \| \Gamma y_{11})$=0xfb
7	$(\Gamma y_{12} \| \Gamma y_{13})$=0x77
8	$(\Gamma y_{14} \| \Gamma y_{15})$=0x5f
9	$(\Gamma y_{16} \| \Gamma y_{17})$=0x0f
10	$(\Gamma y_{18} \| \Gamma y_{19})$=0x0f
11	$(\Gamma y_{20} \| \Gamma y_{21})$=0x00
12	$(\Gamma y_{22} \| \Gamma y_{23})$=0x0f

表 4.8: 18 ラウンドにおける本稿の結果を与えるマスクの一例

t	F 関数の出力マスク
1	$(\Gamma y_0 \| \Gamma y_1)$=0xb0
2	$(\Gamma y_2 \| \Gamma y_3)$=0x00
3	$(\Gamma y_4 \| \Gamma y_5)$=0xb0
4	$(\Gamma y_6 \| \Gamma y_7)$=0x07
5	$(\Gamma y_8 \| \Gamma y_9)$=0xba
6	$(\Gamma y_{10} \| \Gamma y_{11})$=0x7f
7	$(\Gamma y_{12} \| \Gamma y_{13})$=0xba
8	$(\Gamma y_{14} \| \Gamma y_{15})$=0x0b
9	$(\Gamma y_{16} \| \Gamma y_{17})$=0xb0
10	$(\Gamma y_{18} \| \Gamma y_{19})$=0x00
11	$(\Gamma y_{20} \| \Gamma y_{21})$=0xb0
12	$(\Gamma y_{22} \| \Gamma y_{23})$=0x07
13	$(\Gamma y_{24} \| \Gamma y_{25})$=0xbb
14	$(\Gamma y_{26} \| \Gamma y_{27})$=0x3f
15	$(\Gamma y_{28} \| \Gamma y_{29})$=0x0f
16	$(\Gamma y_{30} \| \Gamma y_{31})$=0x0f
17	$(\Gamma y_{32} \| \Gamma y_{33})$=0x00
18	$(\Gamma y_{34} \| \Gamma y_{35})$=0x0f

表 4.9: 22 ラウンドにおける本稿の結果を与えるマスクの一例

t	F 関数の出力マスク
1	$(\Gamma y_0 \vert \Gamma y_1)$=0xb0
2	$(\Gamma y_2 \vert \Gamma y_3)$=0x00
3	$(\Gamma y_4 \vert \Gamma y_5)$=0xb0
4	$(\Gamma y_6 \vert \Gamma y_7)$=0x07
5	$(\Gamma y_8 \vert \Gamma y_9)$=0xba
6	$(\Gamma y_{10} \vert \Gamma y_{11})$=0x7f
7	$(\Gamma y_{12} \vert \Gamma y_{13})$=0xbb
8	$(\Gamma y_{14} \vert \Gamma y_{15})$=0xfb
9	$(\Gamma y_{16} \vert \Gamma y_{17})$=0xab
10	$(\Gamma y_{18} \vert \Gamma y_{19})$=0xb0
11	$(\Gamma y_{20} \vert \Gamma y_{21})$=0x0b
12	$(\Gamma y_{22} \vert \Gamma y_{23})$=0x00
13	$(\Gamma y_{24} \vert \Gamma y_{25})$=0x0b
14	$(\Gamma y_{26} \vert \Gamma y_{27})$=0x70
15	$(\Gamma y_{28} \vert \Gamma y_{29})$=0xab
16	$(\Gamma y_{30} \vert \Gamma y_{31})$=0xf7
17	$(\Gamma y_{32} \vert \Gamma y_{33})$=0xbb
18	$(\Gamma y_{34} \vert \Gamma y_{35})$=0xaf
19	$(\Gamma y_{36} \vert \Gamma y_{37})$=0x0f
20	$(\Gamma y_{38} \vert \Gamma y_{39})$=0x0f
21	$(\Gamma y_{40} \vert \Gamma y_{41})$=0x00
22	$(\Gamma y_{42} \vert \Gamma y_{43})$=0x0f

表 4.10: 26 ラウンドにおける本稿の結果を与えるマスクの一例

t	F 関数の出力マスク
1	$(\Gamma y_0 \vert \Gamma y_1)$=0xf0
2	$(\Gamma y_2 \vert \Gamma y_3)$=0x00
3	$(\Gamma y_4 \vert \Gamma y_5)$=0xf0
4	$(\Gamma y_6 \vert \Gamma y_7)$=0x0f
5	$(\Gamma y_8 \vert \Gamma y_9)$=0xfa
6	$(\Gamma y_{10} \vert \Gamma y_{11})$=0x77
7	$(\Gamma y_{12} \vert \Gamma y_{13})$=0x7f
8	$(\Gamma y_{14} \vert \Gamma y_{15})$=0x57
9	$(\Gamma y_{16} \vert \Gamma y_{17})$=0x07
10	$(\Gamma y_{18} \vert \Gamma y_{19})$=0x07
11	$(\Gamma y_{20} \vert \Gamma y_{21})$=0x00
12	$(\Gamma y_{22} \vert \Gamma y_{23})$=0x07
13	$(\Gamma y_{24} \vert \Gamma y_{25})$=0x0e
14	$(\Gamma y_{26} \vert \Gamma y_{27})$=0x37
15	$(\Gamma y_{28} \vert \Gamma y_{29})$=0xbb
16	$(\Gamma y_{30} \vert \Gamma y_{31})$=0x0a
17	$(\Gamma y_{32} \vert \Gamma y_{33})$=0xb0
18	$(\Gamma y_{34} \vert \Gamma y_{35})$=0x00
19	$(\Gamma y_{36} \vert \Gamma y_{37})$=0xb0
20	$(\Gamma y_{38} \vert \Gamma y_{39})$=0x07
21	$(\Gamma y_{40} \vert \Gamma y_{41})$=0xbb
22	$(\Gamma y_{42} \vert \Gamma y_{43})$=0x3f
23	$(\Gamma y_{44} \vert \Gamma y_{45})$=0x0f
24	$(\Gamma y_{46} \vert \Gamma y_{47})$=0x0f
25	$(\Gamma y_{48} \vert \Gamma y_{49})$=0x00
26	$(\Gamma y_{50} \vert \Gamma y_{51})$=0x0f

第5章　CLEFIAの不能差分攻撃耐性

これまでに報告されているCLEFIAに対する最良の攻撃は、辻原らの不能差分攻撃である[6]。9段の不能差分パスを利用し、128ビット鍵に対して12段まで、鍵の総当たりより少ない計算量で攻撃が可能と報告されている。また、同じ不能差分パスを用い、192,256ビット鍵の場合、それぞれ、13段、14段まで、必要平文数$2^{111.8}$,$2^{112.3}$、計算量2^{155},2^{220}で攻撃が可能とされている。

ここでは、不能差分パス探索プログラムを作成し、CLEFIAの不能差分パスの探索を行った。また、辻原らと同様の不能差分パス探索プログラムを作成し、この結果を追試した。結果として、辻原らの不能差分パスが、現時点の最良パスである事及び彼らの評価が妥当である事を確認した。

5.1　不能差分特性探索法

2007年に角尾らは変形Feistel構造のブロック暗号に対する不能差分特性探索法を提案した[7]。角尾らの手法は、Feistel構造の暗号でF関数が全単射であれば必ず5ラウンドの不能差分が存在することを基とし、変形Feistel構造へ拡張したものである。また、角尾らはこの手法をHIGHTへ適用し、HIGHT提案者の自己評価結果である14ラウンドの不能差分よりも長い15ラウンドの不能差分があることを示した。

ここでは、角尾らが提案した不能差分特性探索法をCLEFIAに適用した探索法について説明する。差分要素、F関数の入出力差分要素の関係、XORによる差分要素の変化及び不能差分を決定する差分要素の特性(以下、「不能差分特性」という。)について述べ、不能差分特性の探索アルゴリズムについて示す。

5.1.1　差分要素の定義並びにXORによる差分要素の変化及びF関数の入出力差分要素の関係

F関数の全単射性のみを利用した場合

不能差分特性の探索で扱う差分要素を表5.1に示す。

表5.1: 差分要素

Z*ero*	：差分0
F*ix*	：非0の任意差分
D*elta*	：非0の非固定差分(**Z***ero*を除く)
R*andom*	：非固定差分(**Z***ero*を含む)

以下、差分要素をZ，F，D，Rで表記する。

XOR による差分要素の変化及び F 関数の入出力差分要素の関係は表 5.2 に従う。例えば、XOR 演算 $\Delta x \oplus \Delta y = \Delta z$ において、Δx の差分要素が Z で Δy の差分要素が D であるとき、Δz の差分要素は D となる。また、CLEFIA の F 関数は S-box は全単射、かつ、MDS 行列は正則であるから、F 関数は全単射性[1]をもつ。また、F 関数が非線形な全単射関数であるとき、その入出力差分 Δx，$\Delta y = \mathrm{F}(\Delta x)$ における差分要素の関係は表 5.2 で与えられる。例えば、F 関数への入力差分要素 Δx が F のとき、出力差分要素 Δy は D となることを表している。

表 5.2: XOR による差分要素の変化及び F 関数の入出力分要素の関係

		XOR Δy				F 関数
		Z	F	D	R	
Δx	Z	Z	F	D	R	Z
	F	F	Z	R	R	D
	D	D	R	R	R	D
	R	R	R	R	R	R

F 関数の MDS 行列の分岐数を利用した場合

不能差分特性の探索で扱う差分要素を表 5.3 に示す。

表 5.3: 差分要素

Z*ero*	: 差分 0
F*ix*	: 非 0 の任意差分
D*elta*	: 非 0 の非固定差分 (**D***elta* $\neq$ **F***ix*)
D*elta***2**	: 非 0 の非固定差分 (**D***elta***2** = **F***ix* $\oplus$ **D***elta*)
D*elta***3**	: 非 0 の非固定差分
R*andom*	: 非固定差分 (**Z***ero* を含む)

以下、差分要素 **D***elta***2**，**D***elta***3** をそれぞれ D2，D3 で表記する。
XOR による差分要素の変化及び F 関数の入出力差分要素の関係は表 5.4 に従う.

表 5.4: XOR による差分要素の変化及び F 関数の入出力分要素の関係

		XOR Δy						F 関数
		Z	F	D	D2	D3	R	
Δx	Z	Z	F	D	D2	D3	R	Z
	F	F	Z	D2	D	R	R	D
	D	D	D2	R	R	R	R	D3
	D2	D2	D	R	R	R	R	D3
	D3	D3	R	R	R	R	R	D3
	R	R	R	R	R	R	R	R

[1]入力差分が非 0 のとき、出力差分が非 0 となる。

辻原らの不能特性探索手法 [6]

不能差分特性の探索で扱う差分要素を表 5.5 に示す.

表 5.5: 差分要素

$\mathbf{Z}ero$	: 差分 0
$\mathbf{F}ix$	: 非 0 の任意差分
$\mathbf{D}elta$	: 非 0 の非固定差分 ($\mathbf{D}elta \neq \mathbf{F}ix$)
$\mathbf{D}elta\mathbf{2}$	: 非 0 の非固定差分 ($\mathbf{D}elta\mathbf{2} = \mathbf{F}ix \oplus \mathbf{D}elta$)
$\mathbf{D}elta\mathbf{3}$	: 非 0 の非固定差分 ($\mathbf{D}elta\mathbf{3} = \mathbf{D}elta \oplus \mathbf{D}elta$)
$\mathbf{D}elta\mathbf{4}$	: 非 0 の非固定差分 ($\mathbf{D}elta\mathbf{4} = \mathbf{D}elta \oplus \mathbf{D}elta\mathbf{2}$)
$\mathbf{D}elta\mathbf{5}$	: 非 0 の非固定差分
$\mathbf{R}andom$	: 非固定差分 ($\mathbf{Z}ero$ を含む)

以下、差分要素 $\mathbf{D}elta\mathbf{4}$, $\mathbf{D}elta\mathbf{5}$ をそれぞれ D4, D5 で表記する。
XOR による差分要素の変化及び F 関数の入出力差分要素の関係は表 5.6 に従う.

表 5.6: XOR による差分要素の変化及び F 関数の入出力分要素の関係

		XOR Δy								F 関数
		Z	F	D	D2	D3	D4	D5	R	
Δx	Z	Z	F	D	D2	D3	D4	D5	R	Z
	F	F	Z	D2	D	R	R	R	R	D
	D	D	D2	D3	D4	R	R	R	R	D5
	D2	D2	D	D4	R	R	R	R	R	D5
	D3	D3	R	R	R	R	R	R	R	D5
	D4	D4	R	R	R	R	R	R	R	D5
	D5	D5	R	R	R	R	R	R	R	D5
	R	R	R	R	R	R	R	R	R	R

5.1.2 不能差分特性

不能差分特性は全単射型不能差分特性 (以下、「全単射型」という。) 及び中間不一致型不能差分特性 (以下、「不一致型」という。) の 2 つのタイプが存在する。以下に、不一致型及び全単射型について説明する。

全単射型

F 関数の全単射性を利用した不能差分特性である.

暗号化処理において、k 系列 Feistel 構造の入力差分 α=(α_0, α_1, $\cdots$, α_{k-1}) が与えられたとき、α が持つ特性を $\boldsymbol{a}$=($\boldsymbol{a}_0$, $\boldsymbol{a}_1$, $\cdots$, $\boldsymbol{a}_{k-1}$) と表記する。また、入力差分 α が与えられたとき、r ラウンド後の出力差分を $\alpha^{(r)}$=($\alpha_0^{(r)}$, $\alpha_1^{(r)}$, $\cdots$, $\alpha_{k-1}^{(r)}$) とし、$\alpha^{(r)}$ が持つ特性を $\boldsymbol{a}^{(r)}$=($\boldsymbol{a}_0^{(r)}$, $\boldsymbol{a}_1^{(r)}$, $\cdots$, $\boldsymbol{a}_{k-1}^{(r)}$) と表記する。ここで、$\boldsymbol{a}_i$ と $\boldsymbol{a}_i^{(r)}$($0 \leq i \leq k-1$) を差分要素という。なお、復号処理の場合は α, α_i, $\alpha^{(r)}$, $\alpha_i^{(r)}$, $\boldsymbol{a}$, $\boldsymbol{a}_i$, $\boldsymbol{a}^{(r)}$, $\boldsymbol{a}_i^{(r)}$ の代わりに β, β_i, $\beta^{(r)}$, $\beta_i^{(r)}$, $\boldsymbol{b}$, $\boldsymbol{b}_i$, $\boldsymbol{b}^{(r)}$, $\boldsymbol{b}_i^{(r)}$ と表記する。

暗号化方向に計算された r_e ラウンド後の出力差分要素を $\boldsymbol{a}^{(r_e)}$=($\boldsymbol{a}_0^{(r_e)}$, $\boldsymbol{a}_1^{(r_e)}$, $\boldsymbol{a}_2^{(r_e)}$, $\boldsymbol{a}_3^{(r_e)}$)、復号方向に計算された r_d ラウンド後の出力差分要素を $\boldsymbol{b}^{(r_d)}$=($\boldsymbol{b}_0^{(r_d)}$, $\boldsymbol{b}_1^{(r_d)}$, $\boldsymbol{b}_2^{(r_d)}$, $\boldsymbol{b}_3^{(r_d)}$) としたとき、次の関係式が成り立つ。

$$\begin{cases} \boldsymbol{b}_0^{(r_d)} = \boldsymbol{a}_1^{(r_e)} \oplus \mathrm{F}_0\left(\boldsymbol{a}_0^{(r_e)}\right) & (5.1) \\ \boldsymbol{b}_1^{(r_d)} = \boldsymbol{a}_2^{(r_e)} & (5.2) \\ \boldsymbol{b}_2^{(r_d)} = \boldsymbol{a}_3^{(r_e)} \oplus \mathrm{F}_1\left(\boldsymbol{a}_2^{(r_e)}\right) & (5.3) \\ \boldsymbol{b}_3^{(r_d)} = \boldsymbol{a}_0^{(r_e)} & (5.4) \end{cases}$$

(5.2) 式及び (5.4) 式より、(5.1) 式及び (5.3) 式は次式で表される。

$$\boldsymbol{a}_1^{(r_e)} = \boldsymbol{b}_0^{(r_d)} \oplus \mathrm{F}_0\left(\boldsymbol{b}_3^{(r_d)}\right), \tag{5.5}$$

$$\boldsymbol{a}_3^{(r_e)} = \boldsymbol{b}_2^{(r_d)} \oplus \mathrm{F}_1\left(\boldsymbol{b}_1^{(r_d)}\right). \tag{5.6}$$

今、(5.1) 式，(5.5) 式並び (5.3) 式，(5.6) 式において、$\boldsymbol{a}_1^{(r_e)} = \boldsymbol{b}_0^{(r_d)} \in \{Z, F\}$，$\boldsymbol{a}_3^{(r_e)} = \boldsymbol{b}_2^{(r_d)} \in \{Z, F\}$ のとき、XOR による差分要素の変化及び F 関数の入出力差分要素の関係より、

$$\mathrm{F}_0\left(\boldsymbol{a}_0^{(r_e)}\right) = \mathrm{F}_0\left(\boldsymbol{b}_3^{(r_d)}\right) = \mathrm{Z}, \tag{5.7}$$

$$\mathrm{F}_1\left(\boldsymbol{a}_2^{(r_e)}\right) = \mathrm{F}_1\left(\boldsymbol{b}_1^{(r_d)}\right) = \mathrm{Z}, \tag{5.8}$$

となる。このとき、$\boldsymbol{a}_0^{(r_e)}$，$\boldsymbol{b}_3^{(r_d)}$，$\boldsymbol{a}_2^{(r_e)}$ または $\boldsymbol{b}_1^{(r_d)}$ のいずれかが Z 以外のとき、(5.7) 式または (5.8) 式に矛盾する。このような差分要素の矛盾を $(r_e + r_d + 1)$ ラウンドの全単射型という。

不一致型

一方が 0 差分、他方が非 0 差分で矛盾となるような不能差分特性である。

暗号化方向に計算された r_e ラウンド後の出力差分要素 $\boldsymbol{a}^{(r_e)}$=($\boldsymbol{a}_0^{(r_e)}$, $\boldsymbol{a}_1^{(r_e)}$, $\boldsymbol{a}_2^{(r_e)}$, $\boldsymbol{a}_3^{(r_e)}$) と復号方向に計算された r_d ラウンド後の出力差分要素 $\boldsymbol{b}^{(r_d)}$=($\boldsymbol{b}_0^{(r_d)}$, $\boldsymbol{b}_1^{(r_d)}$, $\boldsymbol{b}_2^{(r_d)}$, $\boldsymbol{b}_3^{(r_d)}$) において，($\boldsymbol{a}_i^{(r_e)}$, $\boldsymbol{b}_i^{(r_d)}$)($0 \leq i \leq 3$) の差分要素組に 1 つでも矛盾が生じている場合，$(r_e + r_d)$ ラウンドの不一致型という.

5.1.3 探索アルゴリズム

不能差分特性の探索アルゴリズムは次の 3 つのステップで実行される。

Step1：暗号化方向のすべての入力差分要素 $\boldsymbol{a}$=($\boldsymbol{a}_0$, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$)$\in$ { (Z, Z, Z, F), (Z, Z, F, Z) , $\cdots$, (F, F, F, F)} における差分要素の伝搬を XOR による差分要素の変化及び F 関数の入出力分要素の関係に従い、探索する。

Step2：復号方向のすべての入力差分要素 $\boldsymbol{b}$=($\boldsymbol{b}_0$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$,$\boldsymbol{b}_3$)$\in$ { (Z, Z, Z, F), (Z, Z, F, Z) , $\cdots$, (F, F, F, F)} における差分要素の伝搬を XOR による差分要素の変化及び F 関数の入出力分要素の関係に従い、探索する。

Step3：暗号化方向と復号方向のそれぞれの出力差分要素を比較し、全単射型または不一致型により矛盾が生じている差分要素の組み合わせを不能差分特性として検出する。

5.2 CLEFIA の不能差分特性

F 関数の全単射性のみを利用した場合、CLEFIA には (Z, Z, Z, F)↛(Z, Z, Z, F) 及び (Z, F, Z, Z)↛(Z, F, Z, Z) の 9 ラウンドの不能差分特性（全単射型）が見つかった。(Z, Z, Z, F)↛(Z, Z, Z, F) の 9 ラウンドの不能差分特性を図 5.1 に示す。なお、F 関数の MDS 行列の分岐数を利用した場合も同じ結果となった。

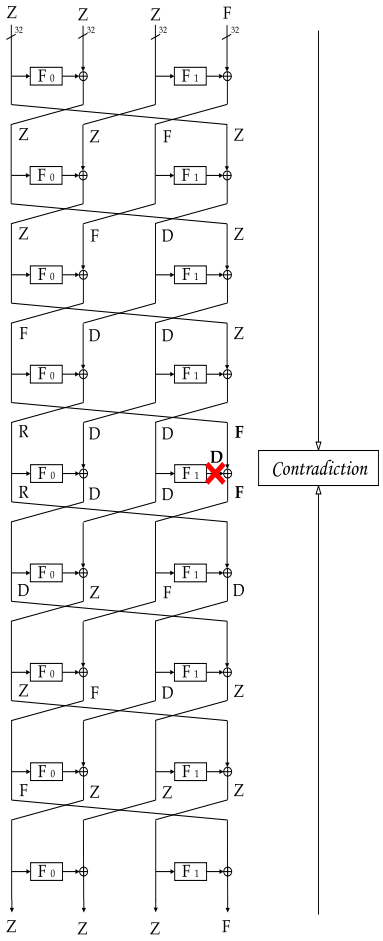


図 5.1: 9 ラウンド不能差分特性

辻原らの探索手法を用いた場合、(Z, Z, Z, F)↛(Z, Z, Z, F) 及び (Z, F, Z, Z)↛(Z, F, Z, Z) の 9 ラウンドの全単射型並びに不一致型が見つかった。結果として、辻原らの探索結果と一致する事を確認した。

5.3 まとめ

今回の調査では、既存結果と同じ不能差分特性が得られることが確認された。結果として、辻原らの不能差分パスが、現時点での最良パスである事及び彼らの評価が妥当である事を確認した。これより、CLEFIA は不能差分攻撃に対し、十分な耐性を持つと考えられる。

第6章　高階差分攻撃（飽和攻撃）

ここではCLEFIAの高階差分特性(飽和特性)をバイト単位で計算機探索し、これまでに報告されている特性及び未知の特性を調査する。さらに、これらの高階差分特性を用いて10段, 11段,12段それぞれの、CLEFIAを攻撃する際に要する平文数と計算量を解析する[15]。

また、n個の異なるバランス関数(入出力はm bit)のXOR和($\oplus$)を出力する回路である(m, n)モデルについて、飽和特性に関する角尾らの予想がある。これは、特殊なバランス状態と呼ばれる。ここでは(m, n)モデルの出力頻度分布に着目し、この分布とアダマール変換、拡大ハミング符号の検査行列を用いて、この予想が正しいことを証明する[16]。この特殊なバランス状態が、CLEFIAにおいて成立しており、素朴な視察で導かれる飽和攻撃特性を超える特性が計算機探索により発見される理由である。

6.1 CLEFIA の飽和攻撃耐性評価

ここでは CLEFIA のバイト単位の飽和特性を調査し、既存結果である 6 ラウンドの 32 階飽和特性を確認する。さらにこの特性を拡張した 8 ラウンド 96 階飽和特性を利用することにより、秘密鍵長 128bit の場合、10 ラウンドの CLEFIA に対して、選択平文数 $2^{97.6}$、計算量 2^{98} で攻撃可能であることを示す。秘密鍵長 192，256bit の場合は、それぞれ 11，12 ラウンドの CLEFIA に対して、選択平文数 $2^{98.3}$，$2^{98.8}$、計算量 2^{159}，2^{223} で攻撃可能であることを示す。

6.1.1 飽和攻撃

飽和攻撃は、1997 年に Daemen らによってブロック暗号 SQUARE に対する攻撃として最初に提案された攻撃法 [9] であり、飽和特性を利用し、ラウンド鍵を回復する手法である。典型的な飽和攻撃は、ブロック暗号のバイト指向構造を利用しており、AES に対しても有効である [10]。

高階差分

定義　入力 $X \in \mathrm{GF}(2)^n$ と鍵 $K \in \mathrm{GF}(2)^s$ から $Y \in \mathrm{GF}(2)^m$ を出力する暗号化関数を $Y = E(X;K)$ で表す。このとき、$E(X;K)$ の X に関する i 階差分は以下のように計算できる。

$$\Delta^{(i)}_{V^{(i)}} E(X;K) = \bigoplus_{\alpha \in V^{(i)}} E(X \oplus \alpha; K) \tag{6.1}$$

ここで、$V^{(i)}$ は $\mathrm{GF}(2)^n$ 上の i 次元部分空間 $V^{(i)}$ であり、その要素 α を入力差分という（以下、$\Delta^{(i)}_{V^{(i)}}$ を $\Delta^{(i)}$ と表記する）。

性質 1　暗号化関数 $E(X;K)$ の X に関するブール代数次数が N 次であるとき、以下が成り立つ。

$$\begin{cases} \Delta^{(N)} E(X;K) &= const \\ \Delta^{(N+1)} E(X;K) &= 0 \end{cases} \tag{6.2}$$

ここで、$const$ は定数である。

飽和特性

定義　$X_i \in \{0,1\}^j$ $(0 \le i < 2^j)$ を 2^j 個の j ビットデータ、X_i の出現度数を Y_i としたとき、X_i を以下の 5 つの状態に分類する。

Constant	:	$\forall i_0, i_1;\ X_{i_0} = X_{i_1}$ を満たす場合
All	:	$\forall i_0, i_1;\ i_0 \neq i_1$ ならば, $X_{i_0} \neq X_{i_1}$ を満たす場合
Even/odd	:	$\forall i_0, i_1;\ Y_{i_0} = Y_{i_1} \pmod 2$ を満たす場合
Balance	:	$\bigoplus_i X_i = 0$ を満たす場合
Unknown	:	不明な場合

以下、Constant を C、All を A、Even/odd を E、Balance を B 及び Unknown を U と表記する。

性質 2　暗号化関数 $E : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^s \to \mathrm{GF}(2)^n$ の出力 $Y \in \mathrm{GF}(2)^n$ が特性 C，A，E または B のとき、その n 階差分値は 0 となる。

攻撃の概要

R 段の暗号化関数 E_R を考える。入力 $X \in \mathrm{GF}(2)^n$ に対して、$(R-1)$ 段目の出力 $Y_{(R-1)}(X) \in \mathrm{GF}(2)^m$ を以下のように表す。

$$Y_{(R-1)}(X) = E_{(R-1)}(X; K_1, K_2, \cdots, K_{(R-1)}) \tag{6.3}$$

ここで、$K_i \in \mathrm{GF}(2)^s$ は i 段目に入力される副鍵である。また、暗号文 $C \in \mathrm{GF}(2)^n$ より、R 段目の鍵 K_R を用いて、$Y_{(R-1)}$ を求める関数 $\tilde{E}(\cdot) : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^s \to \mathrm{GF}(2)^m$ を以下のように表す。

$$Y_{(R-1)}(X) = \tilde{E}(C(X); K_R) \tag{6.4}$$

$E_{(R-1)}(\cdot)$ に対し、性質 1 または性質 2 が観測された場合、次式が成立する。

$$\Delta^{(N)} Y_{(R-1)}(X) = 0 \tag{6.5}$$

このとき、(6.4) 式及び (6.5) 式より以下の式が成り立つ。

$$\bigoplus_{a \in V^{(N)}} \tilde{E}(C(X \oplus \alpha); K_R) = 0 \tag{6.6}$$

(6.6) 式を攻撃方程式と呼び、(6.6) 式は最終段の鍵 K_R の推定が正しい場合は確率 1 で成立し、推定が誤りの場合は確率 2^{-m} で成立するので、攻撃者は正しい鍵 K_R を決定することができる。

6.1.2　CLEFIA の飽和特性

自己評価書 [12] には、以下に示す二つの 6 ラウンドの飽和特性、また、そのラウンド拡張 [11] を行った 8 ラウンドの飽和特性が報告されている。

$(\mathrm{C}, \mathrm{A}, \mathrm{C}, \mathrm{C}) \xrightarrow{6r} (\mathrm{B}, \mathrm{U}, \mathrm{U}, \mathrm{U})$,
$(\mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{A}) \xrightarrow{6r} (\mathrm{U}, \mathrm{U}, \mathrm{B}, \mathrm{U})$,
$(\mathrm{A}_0, \mathrm{C}, \mathrm{A}_1, \mathrm{A}_2) \xrightarrow{8r} (\mathrm{B}, \mathrm{U}, \mathrm{U}, \mathrm{U})$,
$(\mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2, \mathrm{C}) \xrightarrow{8r} (\mathrm{U}, \mathrm{U}, \mathrm{B}, \mathrm{U})$.

ここで、$\mathrm{A}_0 \parallel \mathrm{A}_1 \parallel \mathrm{A}_2$ は All 状態の 96bit である。

角尾らは、F 関数の構造が SP 構造であり、かつ、$m \leq 2n$ である 4 系列の Type-2 一般化 Feistel 構造には以下に示す二つの 6 ラウンドの飽和特性、また、そのラウンド拡張を行った 8 ラウンドの飽和特性が存在すると予想している [8]。なお、m は S-box のビット長、n は S-box の個数である。

$(\mathrm{C}, \mathrm{A}, \mathrm{C}, \mathrm{C}) \xrightarrow{6r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$,
$(\mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{A}) \xrightarrow{6r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$,
$(\mathrm{A}_0, \mathrm{A}_1, \mathrm{C}, \mathrm{A}_2) \xrightarrow{8r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$,
$(\mathrm{C}, \mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2) \xrightarrow{8r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$.

CLEFIA のデータ攪拌部は 4 系列の Type-2 一般化 Feistel 構造であり、1 ラウンドに二つの異なる F 関数を持っている。また、F 関数の構造はどちらも SP 構造である。したがって、CLEFIA においても、同様の飽和特性が存在する。さらに、角尾らは 6 系列以上の Type-2 一般化 Feistel 構造においても、同様の飽和特性が存在すると予想している。

計算機を用いた調査

次式に示すように CLEFIA のデータ攪拌部において、128bit の入力 X を 4 つのブロック $X_i \in \mathrm{GF}(2)^{32}$ $(1 \leq i \leq 4)$ に分割し、さらに、1 ブロックを 8bit の 4 つのサブブロック $X_{ij} \in \mathrm{GF}(2)^8$ $(1 \leq j \leq 4)$ に分割する。

$$X = (X_1, X_2, X_3, X_4), \quad X_i = (X_{i1}, X_{i2}, X_{i3}, X_{i4}).$$

サブブロックに 8 階差分、16 階差分、24 階差分及び 32 階差分を入力し、バイト単位での飽和特性を計算機を用い、調査した。なお、8 階差分及び 16 階差分はすべての入力パターン、24 階差分及び 32 階差分についてはブロックごとのすべての入力パターンに対し、調査を行った。また、$GFN_{8,10}$ に対し、8 階差分及び 32 階差分を用い、同様の調査を行った。

8 階差分，16 階差分及び 24 階差分を用いた飽和特性

8 階差分を用いた場合、5 ラウンド CLEFIA の入出力には以下の関係が見つかった。

(d – 1) ((C C C C) (A C C C) (C C C C) (C C C C))
$\xrightarrow{5r}$ ((U U U U) (U U U U) (B B B B) (U U U U))

(d – 2) ((C C C C) (C C C C) (C C C C) (A C C C))
$\xrightarrow{5r}$ ((B B B B) (U U U U) (U U U U) (U U U U))

入力パターン (A C C C) は (C A C C), (C C A C) 及び (C C C A) に置き換えても出力パターンは変化しない。

16 階差分及び 24 階差分を用いた場合、8 階差分において 5 ラウンドの飽和特性となる入力パターンを一つのみ含んでいれば、5 ラウンド CLEFIA の出力パターンは (d – 1) または (d – 2) となる。また、8 階差分において 5 ラウンドの飽和特性となる入力パターンを二つ含んでいる場合、以下のような関係が見つかった。

(d – 3) ((C C C C) (A C C C) (C C C C) (A C C C))
$\xrightarrow{5r}$ ((B B B B) (U U U U) (B B B B) (U U U U))

32 階差分を用いた飽和特性

32 階差分を用いた場合、6 ラウンド CLEFIA の入出力には以下の関係が見つかった。(d – 4) の飽和特性を図 6.1 に示す。この図において 4 ラウンド目出力の 3 ワード目の $(EEEE)$ が角尾らの予想する ”特殊なバランス状態 ” であり、これが発生する事により、素朴な飽和攻撃解析よりも 1 ラウンド長い飽和特性が得られる。なお、この特殊なバランス状態が成立する事の数学的証明を、第 6.2 節で、与える。

(d – 4) ((C C C C) (A A A A) (C C C C) (C C C C))
$\xrightarrow{6r}$ ((B B B B) (U U U U) (B B B B) (U U U U))

(d – 5) ((C C C C) (C C C C) (C C C C) (A A A A))
$\xrightarrow{6r}$ ((B B B B) (U U U U) (B B B B) (U U U U))

更に、6 ラウンドの飽和特性は 2 ラウンド拡張可能であり、次の 8 ラウンドの飽和特性が得られる。

(I) ((A A A A) (A A A A) (C C C C) (A A A A))
$\xrightarrow{8r}$ ((B B B B) (U U U U) (B B B B) (U U U U))

(II) ((C C C C) (A A A A) (A A A A) (A A A A))
$\xrightarrow{8r}$ ((B B B B) (U U U U) (B B B B) (U U U U))

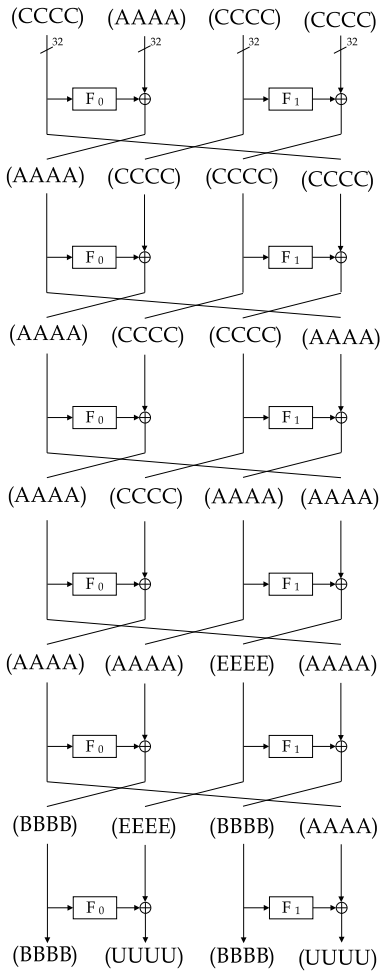


図 6.1: 6 ラウンド CLEFIA の飽和特性

以上より、CLEFIA の 8 ラウンド後の 4 系列中の 2 系列が B となるため、角尾らによって示された飽和特性 [8] と同様の結果が得られた。

6.1.3 CLEFIA の飽和攻撃

CLEFIA には 8 ラウンドの飽和特性が存在し、その中でも (I) または (II) の 96 階差分を用いた飽和特性を利用したとき、12 ラウンド CLEFIA に対し、飽和攻撃が適用可能である。ここでは、(I) の飽和特性を利用し、CLEFIA に対する飽和攻撃に必要な選択平文数及び計算量の見積もりを行う。なお、9 ラウンド鍵回復に必要な選択平文数と計算量については、自己評価書 [12] と同じであるため、説明を省略する。

10 ラウンド鍵回復　CLEFIA の 9 ラウンド目の F_0 関数を等価変形し、M_0 の位置を図 6.2 のように配置する。ただし、M_0^{-1} は M_0 の逆行列であり、$M_0^{-1} = M_0$ である。

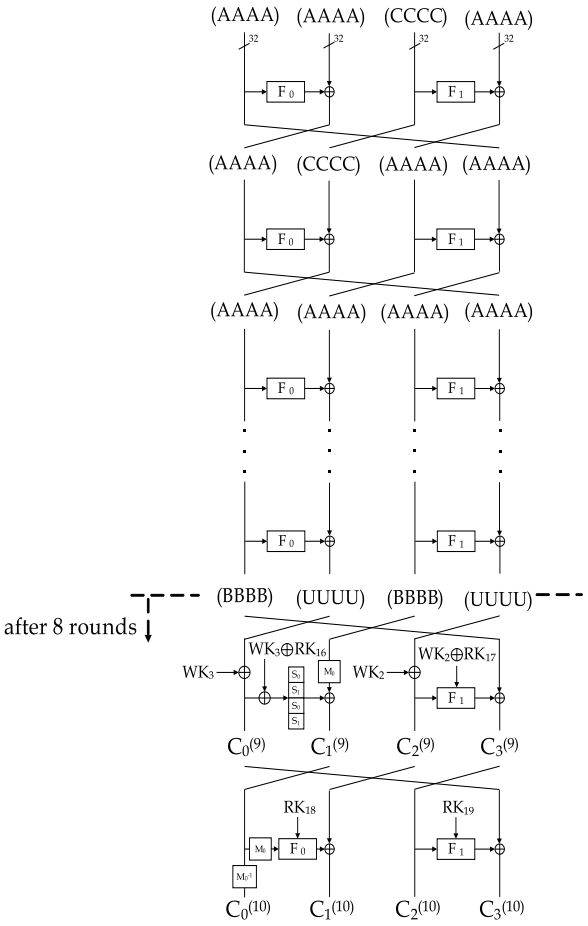


図 6.2: 10 ラウンド CLEFIA に対する鍵回復攻撃

i ラウンド出力の暗号文を $C^{(i)} = (C_0^{(i)}, C_1^{(i)}, C_2^{(i)}, C_3^{(i)})$ としたとき、図 6.2 及び (6.6) 式より、以下の攻撃方程式が得られる。

$$\bigoplus \mathrm{F}_0\left(\mathrm{F}_1\left(C_2^{(10)}; \mathrm{RK}_{19}\right) \oplus C_3^{(10)}; \mathrm{RK}_{16}^{'}\right) \oplus C_0^{(10)} = 0, \tag{6.7}$$

$$\bigoplus \mathrm{F}_1\left(\mathrm{F}_0\left(C_0^{(10)}; \mathrm{RK}_{18}\right) \oplus C_1^{(10)}; \mathrm{RK}_{17}^{'}\right) \oplus C_2^{(10)} = 0, \tag{6.8}$$

なお、$\mathrm{RK}_{16}^{'} = \mathrm{WK}_3 \oplus \mathrm{RK}_{16}$，$\mathrm{RK}_{17}^{'} = \mathrm{WK}_2 \oplus \mathrm{RK}_{17}$ である。(6.7) 式において、64bit の鍵 $\mathrm{RK}_{16}^{'}$=$\mathrm{RK}_{16(0)}^{'} \parallel \mathrm{RK}_{16(1)}^{'} \parallel \mathrm{RK}_{16(2)}^{'} \parallel \mathrm{RK}_{16(3)}^{'}$ 及び RK_{19} を以下の手順に従い、導出する。

1. $C_0^{(10)}$ において、すべての 2^{96} 個の値の XOR を計算し、その値に行列 M_0^{-1} を乗算した値を $Y = Y_0 \parallel Y_1 \parallel Y_2 \parallel Y_3$ とする。

2. (A_0 , A_1 , C , A_2) の形を持つ 2^{96} 個の平文を入力し、$C_2^{(10)} \parallel C_3^{(10)}$ の出現度数をカウントし、奇数回出現した 64bit 値のリスト $\mathrm{LIST}(C_2^{(10)}, C_3^{(10)}) = \mathrm{LIST}(C_2^{(10)}) \parallel \mathrm{LIST}(C_3^{(10)})$ を作成する。

3. すべての $l_{C_2^{(10)}} \in \mathrm{LIST}(C_2^{(10)})$，$l_{C_3^{(10)}} \in \mathrm{LIST}(C_3^{(10)})$ と推測した RK_{19} に対し、次式で表される X_i $(0 \leq i \leq 3)$ の出現度数をカウントし、奇数回出現した 8bit 値の $\mathrm{LIST}(X_i)$ を生成する。
$$l_{C_3^{(10)}} \oplus \mathrm{F}_1\left(l_{C_2^{(10)}}; \mathrm{RK}_{19}\right) = X, \tag{6.9}$$
$$X = X_0 \parallel X_1 \parallel X_2 \parallel X_3.$$

4. すべての $l_{X_i} \in \mathrm{LIST}(X_i)$ と観測した $\mathrm{RK}_{16(i)}^{'}$ に対し、
$$\bigoplus S_j\left(l_{X_i} \oplus \mathrm{RK}_{16(i)}^{'}\right) = Y_i \quad (0 \leq i \leq 3) \tag{6.10}$$
が成立するとき、推測した鍵 $\mathrm{RK}_{16}^{'}$ 及び RK_{19} は正しいと判定する。ここで、j は i が偶数のとき 0、奇数のとき 1 である。

同様に、(6.8) 式において、鍵 $\mathrm{RK}_{17}^{'}$ 及び RK_{18} を導出する。

(6.10) 式は 8bit の 4 つの方程式であるから、偽鍵に対し、成立する確率は $\left(2^{-8}\right)^4 = 2^{-32}$ である。(6.9) 式において、64bit の鍵 $\mathrm{RK}_{16}^{'}$ 及び RK_{19} の推定を行うには、$3(> \frac{64}{32})$ 組の 96 階差分を用意すれば十分である。よって、解読に必要な選択平文数は $3 \cdot 2^{96} \simeq 2^{97.6}$ である。

XOR 及び S-box の計算量を F 関数とみなし、鍵の推定に必要な計算量について考える。手順 1, 2 において、Y の算出及び $\mathrm{LIST}(C_2^{(10)}, C_3^{(10)})$ の作成にかかる計算量は $2 \cdot 2^{96} = 2^{97}$（F 関数）である。手順 3，4 において、一つの推測した 32bit の鍵 RK_{19} に対し、64bit 値の $\mathrm{LIST}(C_2^{(10)}, C_3^{(10)})$ を用い、X を計算する。次に、X を用い、8bit 値の $\mathrm{LIST}(X_i)$ $(0 \leq i \leq 3)$ を計算し、さらに、$\mathrm{LIST}(X_i)$ を用いて 8bit の鍵 $\mathrm{RK}_{16(i)}^{'}$ の推測を行うため、このときの計算量は 2^{32} $(2^{64} + 2^8 \cdot 4 + 2^8 \cdot 4 \cdot 2^8) \simeq 2^{96}$（F 関数）となる。以上より、64bit の鍵の推定に必要な計算量は $2^{97} + 2^{96} \simeq 2^{97.6}$（F 関数）となる。ただし、2 組目以降の鍵の推測に必要な計算量は 1 組目に比べ少ないため、全体の計算量に影響しないものとみなす。ここで、10 ラウンド CLEFIA には 20 個の F 関数が配置されているため、$2^{97.6}$（F 関数）$= \frac{2}{20} \cdot 2^{97.6} \simeq 2^{94.3}$（暗号化）である。また、(6.8) 式においても、同じ計算量で鍵の推定を行うことが可能であるので、10 ラウンドすべての鍵の推定に必要な計算量は $2 \cdot 2^{94.3} = 2^{95.3}$（暗号化）よって、暗号文を求める計算量 $2^{97.6}$（暗号化）、鍵を推定する計算量 $2^{95.3}$（暗号化）より、解読に必要な計算量は $2^{97.6} + 2^{95.3} \simeq 2^{98}$（暗号化）となる。したがって、この攻撃は、鍵長が 128，192 及び 256bit の 10 ラウンド CLEFIA いずれに対し、適用可能である。

作成するリストは 64bit 値の $\mathrm{LIST}(C_2^{(10)}, C_3^{(10)})$ 及び 32bit 値の $\mathrm{LIST}(X)$ である。よって、解読に使用するメモリは $2^{64} + 2^{32} \simeq 2^{64}$（bit）である。

11/12 ラウンド鍵回復　11/12 ラウンド鍵回復については、10 ラウンド鍵回復と同様の手法を用い、解読に必要な選択平文数及び計算量の導出する。すなわち、リスト LIST($C_2^{(10)}, C_3^{(10)}$)，LIST(X_i) により、それぞれ RK_{19}，$\mathrm{RK}'_{16(i)}$ を推定し、残りの鍵については全数探索により推定する。11/12 ラウンド鍵回復攻撃において、全数探索する鍵長はそれぞれ 128，192bit であるため、鍵の推定を行うには 96 階差分がそれぞれ $5(> \frac{128}{32}), 7(> \frac{192}{32})$ 組用意すれは十分である。よって、11/12 ラウンド CLEFIA の解読に必要な選択平文数はそれぞれ $5 \cdot 2^{96} \simeq 2^{98.3}, 7 \cdot 2^{96} \simeq 2^{98.8}$ である。また、10 ラウンド鍵回復において、64bit の鍵を推定するのに必要な計算量は $2^{94.3}$（暗号化）より、11 ラウンド鍵回復では、この操作を 11 ラウンド目の鍵の総数 2^{64} 回繰り返すので、計算量は $2^{94.3} \cdot 2^{64} \simeq 2^{159}$（暗号化）である。同様に、12 ラウンド鍵回復では $2^{94.3} \cdot 2^{128} \simeq 2^{223}$（暗号化）となる。

CLEFIA に対する飽和攻撃の攻撃可能段数、攻撃に必要な選択平文数及び計算量をまとめたものを表 6.1 に示す。

表 6.1: CLEFIA に対する飽和攻撃の結果

ラウンド数	鍵長	選択平文数	計算量
10	128,192,256	$2^{97.6}$	2^{98}
11	192,256	$2^{98.3}$	2^{159}
12	256	$2^{98.8}$	2^{223}

結果として、10 ラウンドのとき、RK'_{16} を 1 バイトごとに導出するため、自己評価書よりも計算量が 2^{-24} 倍程度に少なくなることが分かった。また、これらの結果と自己評価書に示された飽和特性を利用した攻撃に必要な選択平文数を比較した場合、自己評価書に示された飽和特性は 8 ラウンド後の 4 系列中の 1 系列のみ B であるので、各ラウンドにおいて、すべての鍵を回復するには、8 ラウンド後の 4 系列中の 2 系列が B となる飽和特性を利用した攻撃に必要な選択平文数よりも約 1.5∼2 倍の選択平文数が必要である。

6.1.4　まとめ

CLEFIA のバイト単位での飽和特性を調査した結果、CLEFIA には角尾らの予想した 6 ラウンドの飽和特性が存在し、これにラウンド拡張を適用することにより、8 ラウンドの飽和特性が存在することが確認された。この飽和特性を利用し、CLEFIA に飽和攻撃を適用した結果、鍵長 128bit の場合、10 ラウンドの CLEFIA に対して、選択平文数 $2^{97.6}$、計算量 2^{98} で飽和攻撃が可能である。鍵長 192，256bit の場合は、それぞれ 11，12 ラウンドの CLEFIA に対して、選択平文数 $2^{98.3}$，$2^{98.8}$、計算量 2^{159}，2^{223} で飽和攻撃が可能である。結果として、実際の CLEFIA のラウンド数は鍵長が 128bit の場合は 18、鍵長が 192/256bit の場合はそれぞれ 22，26 であるので、CLEFIA は飽和攻撃に対し、十分な耐性を持つと考えられる。

6.2 バランス関数の XOR 和における特殊な飽和特性

角尾らは (m, n) モデルを定義し、(m, n) モデルの飽和特性に関する予想を述べている。本稿ではその予想が正しいことを証明する。(m, n) モデルとは n 個の異なるバランス関数 (入出力は m bit) の XOR 和 ($\oplus$) を出力する回路モデルであり、バランス関数とはその出力全通りの XOR 和がゼロとなる関数である。角尾らの予想とは次の通りである。(m, n) モデルの出力頻度分布は、$m < 2n$ であれば全て偶数であり、$m = 2n$ であれば、奇数か偶数のいずれか一方のみである。そしてこのような予想が成り立つ関数は特殊なバランス関数と呼ばれている。我々は (m, n) モデルの出力頻度分布に着目し、この分布とアダマール変換、拡大ハミング符号の検査行列を用いて角尾らの予想が正しいことを証明する。

6.2.1 (m, n) モデルと角尾らの予想

角尾らは Type-2 一般化 Feistel 構造の飽和特性を検討した論文 [8] 中で (m, n) モデルを定義し、(m, n) モデルの飽和特性に関する予想を述べている。本節では (m, n) モデルの定義と角尾らの予想を示す。

図 6.3 に (m, n) モデルを示す $(m \geq 2, n \geq 1)$。データ線の bit 幅は全て m である。x_i と u_i は、それぞれ関数 $g_i (i = 0, 1, 2, \cdots, n-1)$ の入出力を表し、$X = x_0 \,||\, x_1 \,||\, \cdots \,||\, x_{n-1}$ は (m, n) モデルの入力を表す (|| はデータの連結を表す)。(m, n) モデルの出力 y は u_i の XOR 和 ($\oplus$) である。g_i はバランス関数と呼ばれ、その出力 u_i は次式を満たす。

$$\bigoplus_{i=0}^{2^m-1} u_i = 0. \tag{6.11}$$

次に u_i と y に関する統計量を定義する。初めに入力 $x_i = 0, 1, 2, \cdots, 2^m - 1$ に対する出力 u_i の出現回数の分布 (頻度分布) を $f_i(u)$ とし ($\sum f_i(u) = 2^m$)、$X = 0, 1, 2, \cdots, 2^{mn} - 1$ に対する出力 y の頻度分布を $f_y(y)$ とする ($\sum f_y(y) = 2^{mn}$)。さらに $f_i(u)$ に対して mod2 演算を適用した頻度分布を $f_i^{(2)}(u)$ とし、これを mod2 頻度分布と呼ぶ。つまり $f_i(u)$ が偶数であれば $f_i^{(2)}(u)$ は 0 となり、奇数であれば 1 となる。

次に角尾らの予想を示す。図 6.3 の (m, n) モデル $(m \geq 2, n \geq 1)$ において、

[予想 1] $m < 2n$ ならば $f_y(y)$ は任意の y に対して偶数である。

[予想 2] $m = 2n$ ならば $(f_y(0) + f_y(y))$ は任意の y に対して偶数である。

6.2.2 角尾らの予想の証明

初めに予想 1 から証明する。畳み込み演算を用いると $f_y(\tau)$ は次式で与えられる。

$$\begin{aligned} & f_y(\tau) \\ & = \sum_{u_0=0}^{2^m-1} \sum_{u_1=0}^{2^m-1} \cdots \sum_{u_{n-2}=0}^{2^m-1} f_0(u_0) f_1(u_1) \cdots f_{n-2}(u_{n-2}) \\ & \qquad \cdot f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2} \oplus \tau). \end{aligned} \tag{6.12}$$

ここで頻度分布 $f_i(u)$ の特性関数を $\varphi_i(t)$ として、$f_i(u)$ から $\varphi_i(t)$ への変換を H_{2^m} とする。さらに H_{2^m} の逆変換を $\mathrm{H}_{2^m}^{-1}$ とすると、式 (6.12) は次式で書き直せる。

$$f_y(\tau) = \mathrm{H}_{2^m}^{-1}\big\{\varphi_0(t)\varphi_1(t)\cdots\varphi_{n-2}(t)\varphi_{n-1}(t)\big\}. \tag{6.13}$$

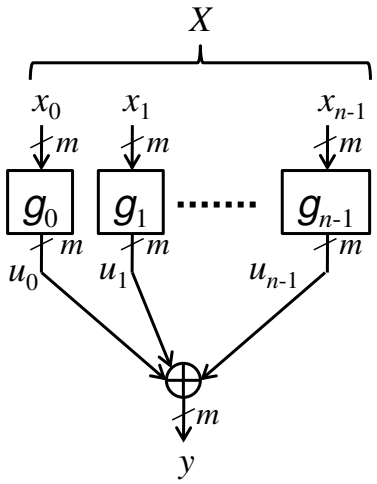


図 6.3: $(m,\, n)$ モデル.

本稿ではアダマール変換とその逆変換をそれぞれ H_{2^m}, $\mathrm{H}_{2^m}^{-1}$ とする (アダマール変換と畳み込みの関係は節 6.2.3 に示す)。

次に $f_i^{(2)}(u)$ 及び $f_i(u)$ の性質を考察する。関数 g_i はバランス関数なので、$f_i^{(2)}(u)$ は符号長 2^m, 情報ビット数 $2^m - m - 1$ の拡大ハミング符号の符号語である。従ってその検査行列を $\mathcal{H}_{2^m}$ とすると次式が成り立つ。

$$\begin{aligned} \mathcal{H}_{2^m}\mathbf{f}_i^{(2)} &= \mathbf{0}, \\ \mathbf{f}_i^{(2)} &= (f_i^{(2)}(0), f_i^{(2)}(1), \cdots, f_i^{(2)}(2^m-1))^t. \end{aligned} \tag{6.14}$$

$(\cdot)^t$ はベクトル及び行列の転置を表す。式 (6.14) では各々の要素の加算は XOR である。式 (6.14) から直ちに次式が導かれる。

$$\begin{aligned} \mathcal{H}_{2^m}\mathbf{f}_i &= (\text{全ての要素が偶数であるベクトル}), \\ \mathbf{f}_i &= (f_i(0), f_i(1), \cdots, f_i(2^m-1))^t. \end{aligned} \tag{6.15}$$

尚、式 (6.15) において、各々の要素の加算は算術加算である。これより、特性関数 $\varphi_i(t) = \mathrm{H}_{2^m} f_i(u)$ の性質として次が導かれる (導出過程は節 6.2.4 に示す)。

$$\varphi_i(t) \text{ は } 4 \text{ を因数に持つ.} \tag{6.16}$$

これより、式 (6.13) において $\mathrm{H}_{2^m}^{-1} = \frac{1}{2^m}\,\mathrm{H}$ に注意すれば、$f_y(\tau)$ は $4^n/2^m = 2^{2n-m}$ を因数に持つことが分かる。従って、$m < 2n$ ならば $f_y(\tau)$ は任意の τ に対して偶数である。

(予想 1 の証明終わり)

次に予想 2 を証明する。尚、$y = 0$ の場合、予想 2 は自明なので証明は省く。畳み込み演算を用

いると $f_y(0) + f_y(\tau)$ は次式で与えられる。

$$\begin{aligned} & f_y(0) + f_y(\tau) \\ & = \sum_{u_0=0}^{2^m-1} \sum_{u_1=0}^{2^m-1} \cdots \sum_{u_{n-2}=0}^{2^m-1} f_0(u_0) f_1(u_1) \cdots f_{n-2}(u_{n-2}) \\ & \qquad \cdot \big\{ f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2}) \\ & \qquad\quad + f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2} \oplus \tau) \big\}. \end{aligned} \tag{6.17}$$

これを等価変形すると次式となる。

$$\begin{aligned} & f_y(0) + f_y(\tau) \\ & = \sum_{u_0 \in U(\tau)} \sum_{u_1 \in U(\tau)} \cdots \sum_{u_{n-2} \in U(\tau)} \\ & \quad \{ f_0(u_0) + f_0(u_0 \oplus \tau) \} \{ f_1(u_1) + (f_1(u_1 \oplus \tau) \} \cdots \\ & \quad \cdot \{ f_{n-2}(u_{n-2}) + f_{n-2}(u_{n-2} \oplus \tau) \} \\ & \quad \cdot \big\{ f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2}) \\ & \qquad + f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2} \oplus \tau) \big\}. \end{aligned} \tag{6.18}$$

ここで集合 $U = \{0, 1, 2, \cdots, 2^m - 1\}$ とすると、τ に依存する集合 $U(\tau)$ は $\{U(\tau), U(\tau) \oplus \tau\} = U$ を満たす。

次に $F_i(u) = (f_i(u) + f_i(u \oplus \tau))$ $(u \neq u \oplus \tau)$ とおき、そのアダマール変換を $\varphi_{F_i}(t) = \mathrm{H}_{2^{m-1}} F_i(u)$ とすると、性質 (6.16) から次の性質が導かれる。

$$\varphi_{F_i}(t) \text{ は } 4 \text{ を因数に持つ.} \tag{6.19}$$

$\varphi_{F_i}(t)$ を用いると、$f_y(0) + f_y(\tau)$ は次式で与えられる。

$$\begin{aligned} f_y(0) + f_y(\tau) &= \frac{1}{2^{m-1}} \mathrm{H}_{2^{m-1}} \varphi(t), \\ \varphi(t) &= \varphi_{F_0}(t) \varphi_{F_1}(t) \cdots \varphi_{F_{n-1}}(t). \end{aligned} \tag{6.20}$$

尚、上段左辺の第 1 項の引数は 0 に固定されているので、式 (6.22) により逆変換する際には $\mathbf{x}$=$\mathbf{0}$ に固定する。ここで性質 (6.19) より $\varphi(t)$ は 4^n を因数に持つ。従って $f_y(0) + f_y(\tau)$ は $4^n/2^{m-1} = 2^{2n-m+1}$ を因数に持つ。これより、$m \leq 2n$ ならば頻度分布 $f_y(0) + f_y(\tau)$ は任意の τ において偶数となる。

(予想 2 の証明終わり)

6.2.3 アダマール変換と畳み込み演算の関係

定義域が $\mathrm{GF}(2)^m$ である任意の 2 つの実数関数を $p_i(\mathbf{x})$ $(i = 0, 1)$ とする ($\mathbf{x} \in \mathrm{GF}(2)^m$, $p_i(\mathbf{x}) \in R$)。さらに $p_i(\mathbf{x})$ をアダマール変換した関数を $P_i(\mathbf{t}) = \mathrm{H}_{2^m}\ p_i(\mathbf{x})$ とする ($\mathbf{t} \in \mathrm{GF}(2)^m$, $P_i(\mathbf{t}) \in R$) と、アダマール変換 H_{2^m} は次式で定義される。

$$P_i(\mathbf{t}) = \sum_{\mathbf{x}} p_i(\mathbf{x}) (-1)^{\mathbf{x} \cdot \mathbf{t}}. \tag{6.21}$$

また逆変換 $\mathrm{H}_{2^m}^{-1}$ は次式で定義される。

$$p_i(\mathbf{x}) = \frac{1}{2^m} \sum_{\mathbf{t}} P_i(\mathbf{t})(-1)^{\mathbf{x}\cdot\mathbf{t}}. \tag{6.22}$$

次に $p_0(\mathbf{x})$ と $p_1(\mathbf{x})$ の畳み込みを $q(\boldsymbol{\tau})$ とすると、$q(\boldsymbol{\tau})$ は次式で与えられる。

$$\begin{aligned} q(\boldsymbol{\tau}) &= p_0(\mathbf{x}) * p_1(\mathbf{x}) \\ &= \sum_{\mathbf{x}} p_0(\mathbf{x}) p_1(\mathbf{x} \oplus \boldsymbol{\tau}). \end{aligned} \tag{6.23}$$

さらに $q(\boldsymbol{\tau})$ のアダマール変換を $Q(\mathbf{t})$ とすれば、$Q(\mathbf{t})$ は次式で与えられる。

$$\begin{aligned} Q(\mathbf{t}) &= \sum_{\boldsymbol{\tau}} q(\boldsymbol{\tau})(-1)^{\boldsymbol{\tau}\cdot\mathbf{t}} \\ &= \sum_{\boldsymbol{\tau}} \sum_{\mathbf{x}} p_0(\mathbf{x}) p_1(\mathbf{x} \oplus \boldsymbol{\tau})(-1)^{\boldsymbol{\tau}\cdot\mathbf{t}}. \end{aligned} \tag{6.24}$$

式 (6.24) において $\boldsymbol{\sigma} = \mathbf{x} \oplus \boldsymbol{\tau}$ として整理すると次式となる。

$$\begin{aligned} Q(\mathbf{t}) &= \sum_{\boldsymbol{\sigma}} \sum_{\mathbf{x}} p_0(\mathbf{x}) p_1(\boldsymbol{\sigma})(-1)^{(\mathbf{x}\oplus\boldsymbol{\sigma})\cdot\mathbf{t}} \\ &= \sum_{\boldsymbol{\sigma}} \sum_{\mathbf{x}} p_0(\mathbf{x}) p_1(\boldsymbol{\sigma})(-1)^{\mathbf{x}\cdot\mathbf{t}\oplus\boldsymbol{\sigma}\cdot\mathbf{t}}. \end{aligned} \tag{6.25}$$

$(-1)^0 = 1$ なので式 (6.25) において $(-1)^{\mathbf{x}\cdot\mathbf{t}\oplus\boldsymbol{\sigma}\cdot\mathbf{t}} = (-1)^{\mathbf{x}\cdot\mathbf{t}} \cdot (-1)^{\boldsymbol{\sigma}\cdot\mathbf{t}}$ と変形できる。これを適用すると式 (6.25) は次式となる。

$$\begin{aligned} Q(\mathbf{t}) &= \sum_{\boldsymbol{\sigma}} \sum_{\mathbf{x}} p_0(\mathbf{x})(-1)^{\mathbf{x}\cdot\mathbf{t}} \cdot p_1(\boldsymbol{\sigma})(-1)^{\boldsymbol{\sigma}\cdot\mathbf{t}} \\ &= \sum_{\mathbf{x}} p_0(\mathbf{x})(-1)^{\mathbf{x}\cdot\mathbf{t}} \cdot \sum_{\boldsymbol{\sigma}} p_1(\boldsymbol{\sigma})(-1)^{\boldsymbol{\sigma}\cdot\mathbf{t}} \\ &= P_0(\mathbf{t}) \cdot P_1(\mathbf{t}). \end{aligned} \tag{6.26}$$

故に式 (6.22) を用いて式 (6.26) を逆変換することにより、式 (6.23) に示された畳み込み $q(\boldsymbol{\tau})$ が得られる。$i = 0, 1, 2, \cdots$ と増えた場合も同様である。

6.2.4 性質 (6.16) の導出

性質 (6.16) 導出の理解を助けるために、ここではアダマール変換を行列として定義し、変換前後の関数はベクトルとする。アダマール行列 $\mathbf{H}$ とその逆行列 $\mathbf{H}^{-1}$ は次式で定義される。

$$\begin{aligned} &\mathbf{H}_1 = 1, \\ &\mathbf{H}_{2^m} = \begin{bmatrix} \mathbf{H}_{2^{m-1}} & \mathbf{H}_{2^{m-1}} \\ \mathbf{H}_{2^{m-1}} & -\mathbf{H}_{2^{m-1}} \end{bmatrix}, \\ &\mathbf{H}_{2^m}^{-1} = \frac{1}{2^m}\mathbf{H}_{2^m}, \quad (1 \le m \in N). \end{aligned} \tag{6.27}$$

ここで $\mathbf{H}$ の下付き添え字は行列の次数を表す。$\mathbf{H}_{2^m}$ の第 1 行目の要素は全て 1 である。その他の行は 1 の要素数が 2^{m-1} であり、-1 の要素数が 2^{m-1} である。また $\mathbf{H}_{2^m}$ の任意の行は符号長 2^m である拡大ハミング符号の検査行列 $\mathcal{H}_{2^m}$ の行の線形和となっている (但し、$\mathbf{H}_{2^m}$ の要素-1 は 0 で

置き換える)。従って $\boldsymbol{\varphi} = \mathbf{H}_{2^m}\ \mathbf{f}_i$ とすると、ベクトル $\boldsymbol{\varphi}$ の第 0 要素 φ_0 は 2^m である ($\because \sum_{u=0}^{2^m-1} f_i(u) = 2^m$. $f_i(u)$ は $\mathbf{f}_i$ の要素)。第 1 要素 φ_1 は次式となる。

$$\varphi_1 = \left(\sum_{u=0}^{2^{m-1}-2} f_i(2u) \right) - \left(\sum_{u=0}^{2^{m-1}-2} f_i(2u+1) \right). \tag{6.28}$$

ここで右辺の第 1 項と第 2 項は共に拡大ハミング符号の符号語の要素の総和であるので式 (6.15) よりそれらの値は偶数となる。これより式 (6.28) は次式で書き直せる。

$$\begin{aligned} \varphi_1 &= \left(2^{m-1} + 2a\right) - \left(2^{m-1} - 2a\right) \\ &= 4a, \quad (a \in Z). \end{aligned} \tag{6.29}$$

従ってベクトル $\boldsymbol{\varphi}$ の第 1 要素 φ_1 は 4 を因数に持つ。第 2 要素以降についても式 (6.29) と同様なので、4 を因数に持つ。これより、性質 (6.16) が導かれる。

第7章　補間攻撃及び代数攻撃

一般には、ある関数について次数が高く、項数も多ければ、補間攻撃や代数攻撃に対して安全であると考えられている。ここでは S-box S_0 と S_1 のブール多項式及び GF(2^8) 上の補間多項式を解析し、多項式の次数や項数を調査した。これらを第 7.1 節、第 7.4 節に示す。結果は、自己評価書と同一である。

CLEFIA のデータランダム化部は、図 7.1 の様に、F_0 関数の出力が、次段 F_0 関数に、F_1 関数の出力が、次段 F_1 関数に入る構造を持つ。$\mathbf{P_0}$ を入力変数とすれば、3 段目までは、F_0 関数の直列接続で、次数の上昇が起きる。同様に、$\mathbf{P_1}$ を入力とした場合 F_1 関数の直列接続である。ここでは、3 段までの F_i 関数の直列接続のブール多項式の解析を行った。結果は、第 7.2 節、第 7.3 節に示すが、2 段接続で、28 次 3 段接続で 31 次である。S-box の次数である 7 次 (又は 6 次) のべき乗で、次数上昇が起きる訳ではないが、フルラウンドでは、18 段以上あり、自己評価書の主張と同じ理由でブール多項式に基づく補間攻撃や代数攻撃は、困難と考える。

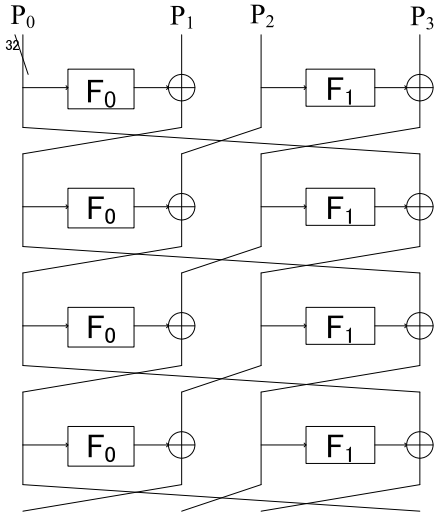


図 7.1: CLEFIA データランダム化部

7.1 F_i 関数 1 段の解析

7.1.1 S-box S_0 と S_1 のブール多項式の解析

図 7.2 に 8bit 入出力 S-box: $\mathrm{GF}(2)^8 \to \mathrm{GF}(2)^8$ の模式図を示す。x_i, $y_i \in \mathrm{GF}(2)$ ($i = 0, 1, 2, \cdots, 7$) はそれぞれ入出力ビットを表す。自己評価書においては y_i の最小次数は S_0 の場合 6 であり、S_1 の場合 7 であると報告されている。

次に我々が S_i のブール多項式を解析した結果を表 7.1, 7.2 に示す。表の見方は次の通りである。例えば表 7.1 の y_0 の列の場合、y_0 のブール多項式は 6 次項を 10 個持ち、5 次項を 28 個持ち、$\cdots$、定数項を 1 個持ち、合計 118 個の項を持つことを表している。表 7.1, 7.2 の右端の列は (i 次項の総種類数)/2 を示している。これらの値はブール多項式の各項がランダムに出現していると仮定した時に期待される項数である。表 7.1, 7.2 より、S_0 では全ての出力ビット y_i の次数が 6 であることが分かり、S_1 では全ての y_i の次数が 7 であることが分かる。また最高次の項数が多く、項の総数も多いものを代数攻撃に" 強い" と仮定すると、S_0 における y_1 と S_1 における y_3 は相対的に強く、S_0 における y_6 と S_1 における y_4 は相対的に弱いと見える。

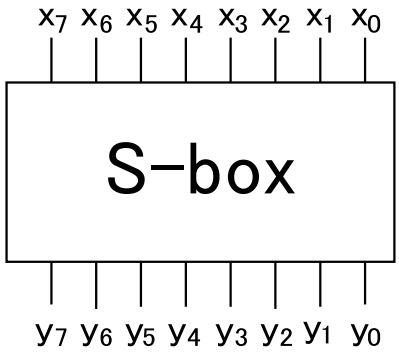


図 7.2: 8bit 入出力 S-box の模式図

表 7.1: S_0 のブール多項式の次数及び項数

	y0	y1	y2	y3	y4	y5	y6	y7	項数期待値
6 次項	10	10	8	6	8	8	5	9	${}_8\mathrm{C}_6/2$=14
5 次項	28	32	16	29	29	26	24	22	${}_8\mathrm{C}_5/2$=28
4 次項	38	37	32	39	37	37	26	42	${}_8\mathrm{C}_4/2$=35
3 次項	25	23	28	23	25	31	29	20	${}_8\mathrm{C}_3/2$=28
2 次項	14	14	13	13	11	14	12	13	${}_8\mathrm{C}_2/2$=14
1 次項	2	6	3	5	5	4	5	5	${}_8\mathrm{C}_1/2$=4
定数項	1	1	1	0	1	0	1	0	${}_8\mathrm{C}_0/2$=0.5
総和	118	123	101	115	116	120	102	111	123.5

表 7.2: S_1 のブール多項式の次数及び項数

	y0	y1	y2	y3	y4	y5	y6	y7	項数期待値
7 次項	2	3	2	6	1	3	4	4	${}_8\mathrm{C}_7/2$=4
6 次項	12	14	14	14	9	12	13	15	${}_8\mathrm{C}_6/2$=14
5 次項	25	26	31	34	25	31	28	28	${}_8\mathrm{C}_5/2$=28
4 次項	45	36	38	39	32	30	39	36	${}_8\mathrm{C}_4/2$=35
3 次項	31	32	22	39	23	22	20	27	${}_8\mathrm{C}_3/2$=28
2 次項	16	13	16	15	13	16	14	13	${}_8\mathrm{C}_2/2$=14
1 次項	4	6	5	3	6	5	3	5	${}_8\mathrm{C}_1/2$=4
定数項	0	0	1	1	0	1	1	0	${}_8\mathrm{C}_0/2$=0.5
総和	135	130	129	151	109	120	122	128	127.5

7.1.2 F_i 関数のブール多項式

第 1 段目の F 関数は前節での $S-box$ の多項式次数の解析及び F_i 関数内で使用される $4 \times 4 MDS$ 行列の性質より、F_i 関数出力ビットは 32 元 7 次多項式 (i=0,1) になる。

7.2 2 段直列 F_i 関数のブール多項式の解析

ここでは 2 段直列 F_i 関数のブール多項式の解析について述べる。CLEFIA の鍵スケジュール部、データ処理部に見られる 2 段直列 F_i 関数の構造を図 7.3 に示す。x_i ($i = 0, 1, 2, \cdots, 63$) は入力ビットを表し、y_i ($i = 0, 1, 2, \cdots, 31$) は出力ビットを表す。$\oplus$ は XOR を表す。

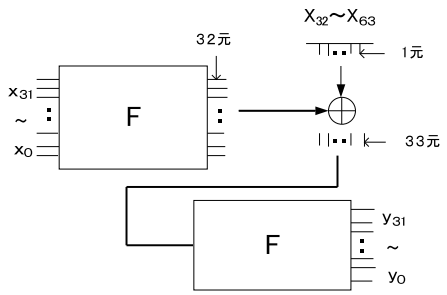


図 7.3: 64 ビット入力 32 ビット出力の 2 段直列 F_i 関数

7.2.1 形式的次数解析

前節より第 1 段目の F_i 関数出力ビットは 32 元 7 次多項式 ($i = 0,1$) である。XOR 演算後では各ビットにおいて新たな 1 次項が追加されるので、33 元 7 次多項式 ($i = 0,1$) になる。更に第 2 段目の F_i 関数出力ビットは、64 元の多項式であり、形式上の最高次数は、自己評価書の様に、$7^2 = 49$ 次となる。

7.2.2 F_i 関数の 1 対 1 特性を考慮した解析

図 7.3 の x_i $(i = 32, 33, \cdots, 63)$ を固定すると、2 段接続の F_i 関数は、入力 x_i $(i = 0, 1, \cdots, 31)$ と出力 y_i $(i = 0, 1, \cdots, 31)$ の間で、1 対 1 写像となっている。従って、出力 y_i のどのビットも、x_i $(i = 0, 1, \cdots, 31)$ に関し、高々31 次式となる。出力 y_i は、x_i $(i = 32, 33, \cdots, 63)$ に対しては高々7 次式であり、結局、F_i 関数出力ビットは、64 元の高々31 次式である。

7.2.3 $S-box$ の 1 対 1 特性を考慮した解析

図 7.3 の出力 y_i の最高次数を求める為には、入力 x_i $(i = 32, 33, \cdots, 63)$ を省いた 32 ビット入出力の 2 段直列 F_i 関数 (図 7.4) の次数を考えれば十分である。図において、$\mathbf{x_i}, \mathbf{y_i}, \mathbf{z_i}$ は 1 バイト変数である。

ここで、$\mathbf{x_0} = (x_{00}, x_{01}, \cdots, x_{07})$ を変数とし、$\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$ を定数と考えると、$\mathbf{x_0}$ と $\mathbf{z_0}$ は、1 対 1 対応となる。従って、$\mathbf{z_0}$ の各ビットのブール多項式表現は、x_{0j} に関し、高々7 次式である。同じく、出力バイト $\mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}$ においても、x_{0j} に関し、高々7 次式である。

同様の考察が、他の入力バイト $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$ に対しても成立するので、変数 $\mathbf{z_i}$ $(i = 0, 1, 2, 3)$ の各ビットのブール多項式表現は、入力変数 x_{ij} $(i = 0, 1, 2, 3, j = 0, 1, \cdots, 7)$ に対し高々$7 * 4 = 28$ 次式である。MDS 行列は、線形変換であり、最高次数には影響しないから、図の出力 $\mathbf{y_i}$ の各ビットのブール代数次数も、高々28 次と推定される。

7.2.4 高階差分による次数の確認

ここで高階差分の次の性質を用いる。

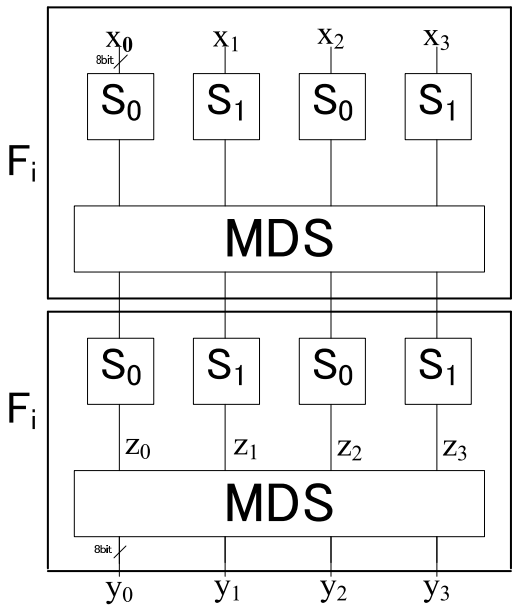


図 7.4: 32 ビット入出力の 2 段直列 F_i 関数

関数 $F(X;K)$ の X に関するブール代数次数が N に等しい時、X,K によらず以下の式が成立する。

$$\deg_X \{F(X;K)\}{=}N \rightarrow \begin{cases} \Delta^{(N+1)}F(X;K){=}0 \\ \Delta^{(N)}F(X;K){=}const \end{cases}$$

第 7.2.3 節の考察を計算機により確認すべく、図 7.4 に示す 2 段直列 F_i 関数の 29 階差分を求めた。29 階差分においては、入力変数 $\mathbf{x_i}$ $(i = 0, 1, 2, 3)$ のどれか 1 バイトに関しては、x_{ij} $(j = 0, 1, \cdots, 7)$ の 8 階差分が適用されている。推論通り、どの 29 階差分も高階差分値は 0 となり、最高次数は 28 以下で有る事が、確認された。

また、入力の 4 バイト $\mathbf{x_i}$ $(i = 0, 1, 2, 3)$ のどれか 1 バイトに関し、8 階差分を含むような高階差分も 0 である。従って、ブール代数式において、8 次項 $x_{i0}x_{i1}x_{i2}x_{i3}x_{i4}x_{i5}x_{i6}x_{i7}$ を因数に含む項は存在しない。

従って、存在しうる 28 次項は、${}_{32}C_{28}$ 通りではなく、4 バイト $\mathbf{x_i}$ $(i = 0, 1, 2, 3)$ の各バイト内の変数に対し 7 次項を構成する組み合わせ $({}_8C_7)^4 = 4096$ 通りである。これら全てに対し、28 階差分を確認し[1]、出力 y_i $(i = 0, 1, \cdots, 31)$ の 32 ビットどれかの高階差分値は 1 であり、対応する 28 次項が存在する事を確認した。4096 通りのデータを示すのは、スペースの無駄であるので、この結果を、32 ビットの出力 y_i $(i = 0, 1, \cdots, 31)$ には、何らかの 28 次項が含まれている事を示す意図をで、まとめたのが表 7.3, 表 7.4 である。

表には、4 バイトの入力 $(\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3})$ を 32 ビット入力 $(x_0, x_1, \cdots, x_{31})$ で表し、x_1 の入っていない 7 次項 $x_0x_2x_3x_4x_5x_6x_7$ を $x^7_{[1]}$ と表記する。表の見方は次の通りである。第 1 列には、この表記法で 28 次項を表している。例えば $x^{28}_{[2,11,18,24]}$ は、$x_2, x_{11}, x_{18}, x_{24}$ の入っていない 28 次項である。第 2 列には、それを 16 進数表示で示す。$x^{28}_{[2,11,18,24]}$ では、$FDEF7BDD$ となる。第 3 列には、その 28 次項が、出力 y_i $(i = 0, 1, 2, \cdots, 31)$ に存在している所を 1, 存在していない所を 0

[1]MDS 行列のある種の対称性を利用すると、$(36)^2 = 1296$ 通りを調べる事で全てを尽くす事になる

と表している。この表を数行辿れば、F_i 関数の 2 段接続においては、F_0、F_1 のいずれでも 32 ビット出力には何らかの 28 次項が現れている事が確認出来る。従って F_i 関数の 2 段接続は、すべて最高次数 28 次をもつ。

なお、表の最下行には、入力 $\mathbf{x_i}$ の同一バイト内の 8 ビットで構成される 8 次項を因数に持つ高次項は存在しない事を示すデータ例として、$x^{28}_{[14,18,22,26]}$ の結果も示してある。

表 7.3: 28 階差分 F_0 2 段接続

28 次項	28 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{28}_{[2,11,18,24]}$	DFEFDF7F	00111001100010011010000001101001
$x^{28}_{[1,13,23,31]}$	BFFBFEFE	11000100000000010101100010111010
$x^{28}_{[5,10,19,29]}$	FBDFF7FB	10001101000011011011001010111110
$x^{28}_{[5,13,17,26]}$	FBFBFBDF	10110010111101101001001010110110
$x^{28}_{[5,10,17,26]}$	FBDFBFDF	10010001001010111001110101010000
	⋮	
$x^{28}_{[1,10,17,26]}$	BFDFBFDF	10101101111101011010110111110101
$x^{28}_{[1,9,19,24]}$	BFBFEF7F	10101101010110010000101000111101
$x^{28}_{[14,18,22,26]}$	FFFDDDDF	00000000000000000000000000000000

表 7.4: 28 階差分 F_1 2 段接続

28 次項	28 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{28}_{[2,11,18,24]}$	DFEFDF7F	00101011010010101101100010101100
$x^{28}_{[1,13,23,31]}$	BFFBFEFE	11001000011010100010101101101001
$x^{28}_{[5,10,19,29]}$	FBDFF7FB	00110000110111100001010000100100
$x^{28}_{[5,13,17,26]}$	FBFBFBDF	01010111011010110011000000110001
$x^{28}_{[5,10,17,26]}$	FBDFBFDF	10111101010100000100101100010000
	⋮	
$x^{28}_{[1,10,17,26]}$	BFDFBFDF	01110110011101110111011001110111
$x^{28}_{[1,9,19,24]}$	BFBFEF7F	00111111110010101010101011101011
$x^{28}_{[14,18,22,26]}$	FFFDDDDF	00000000000000000000000000000000

また、同一バイト内の 8 次項を因数に持たない 27 次、26 次項が、全ての出力ビットに存在する事を、同様に確認した。結果を表 7.5、表 7.6、表 7.7、表 7.8 に示す。

7.3 3 段接続 F_i 関数のブール多項式の解析

F_i 関数の 3 段接続においても、32 ビット入出力の 1 対 1 対応は、成り立つ。従ってブール多項式の最高次数は、高々31 次である。前節と同様に、計算機により高階差分を調査し、最高次数が 31 次であった事を確認した。

表 7.5: 26 階差分 F_0 2 段接続

26 次項	26 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{26}_{[6,11,16,21,26,30]}$	FDEF7BDD	10000101011110011011110101111111
$x^{26}_{[3,7,11,16,20,27]}$	EEEF77EF	00010110000001110100010000110011
$x^{26}_{[2,6,12,18,21,26]}$	DDF7DBDF	00010110000001110100010000110011
$x^{26}_{[4,14,21,25,27,30]}$	F7FDFBAD	01110010000100100010100111101001
$x^{26}_{[1,2,9,17,21,29]}$	9FBFBBFB	01010111010100011011011101110000
⋮		
$x^{26}_{[1,4,8,11,23,29]}$	B76FFEFB	01011011001101001111001100110101
$x^{26}_{[3,8,15,20,23,31]}$	EF7EF6FE	10101000110110100111101100101001
$x^{26}_{[1,6,7,10,24,28]}$	BCDFFF77	00000000000000000000000000000000

表 7.6: 27 階差分 F_0 2 段接続

27 次項	27 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{27}_{[5,10,16,23,27]}$	FBDF7EEF	01101101011001100010111100111011
$x^{27}_{[4,9,15,21,30]}$	F7BEFBFD	11101001001011000000110101110100
$x^{27}_{[2,10,23,26,29]}$	DFDFFEDB	00010000100011101011111110101000
$x^{27}_{[6,8,13,20,27]}$	FD7BF7EF	01000110111111011000001001010110
$x^{27}_{[4,7,12,21,24]}$	F6F7FB7F	11000110110011001111110101101101
⋮		
$x^{27}_{[1,9,16,20,26]}$	BFBF77DF	01010010100010000010011100110100
$x^{27}_{[4,12,18,23,25]}$	F7F7DEBF	01010010100010000010011100110100
$x^{27}_{[1,2,3,15,16]}$	8FFE7FFF	00000000000000000000000000000000

表 7.7: 26 階差分 F_1 2 段接続

26 次項	26 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{26}_{[6,11,16,21,26,30]}$	FDEF7BDD	01010101101001010001101000101111
$x^{26}_{[3,7,11,16,20,27]}$	EEEF77EF	01000011110101010111100101011001
$x^{26}_{[2,6,12,18,21,26]}$	DDF7DBDF	01100100111010010111000000000101
$x^{26}_{[4,14,21,25,27,30]}$	F7FDFBAD	00111100101100111001100001010100
$x^{26}_{[1,2,9,17,21,29]}$	9FBFBBFB	11101010110100000011111100000011
$x^{26}_{[1,4,8,11,23,29]}$	B76FFEFB	00110110011101100010011100111101
⋮		
$x^{26}_{[3,8,15,20,23,31]}$	EF7EF6FE	01110011001110110100001001001001
$x^{26}_{[0,4,11,19,23,27]}$	77EFEEEF	01111001010110010100001111010101
$x^{26}_{[1,6,7,10,24,28]}$	BCDFFF77	00000000000000000000000000000000

表 7.8: 27 階差分 F_1 2 段接続

27 次項	27 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{27}_{[5,10,16,23,27]}$	FBDF7EEF	01101101010011111110000100111101
$x^{27}_{[4,9,15,21,30]}$	F7BEFBFD	00000011100001011001110001111100
$x^{27}_{[2,10,23,26,29]}$	DFDFFEDB	10010111000010101000010010100010
$x^{27}_{[6,8,13,20,27]}$	FD7BF7EF	01111101000010000111000011011100
$x^{27}_{[4,7,12,21,24]}$	F6F7FB7F	10011000001100110001010101001001
	⋮	
$x^{27}_{[1,9,16,20,26]}$	BFBF77DF	11110111101100101010101001111001
$x^{27}_{[4,12,18,23,25]}$	F7F7DEBF	11000000010100010101111111111011
$x^{27}_{[1,2,3,15,16]}$	8FFE7FFF	00000000000000000000000000000000

結果を表 7.9,7.10 に示す。これより、全ての 31 次項が現れている事、出力 y_i $(i = 0, 1, \cdots, 31)$ の 32 ビットには、いずれかの 31 次項が現れている事がわかる。

表 7.9: 31 階差分 F_0 3 段

31 次項	31 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{31}_{[0]}$	7FFFFFFF	01100110001111000110010010100000
$x^{31}_{[1]}$	BFFFFFFF	10011000111111100110101111010111
$x^{31}_{[2]}$	DFFFFFFF	10111111001001101100110011011000
$x^{31}_{[3]}$	EFFFFFFF	10101111010001101100000000101010
$x^{31}_{[4]}$	F7FFFFFF	11010011101001100011001010100000
$x^{31}_{[5]}$	FBFFFFFF	10110000101100001000010011010100
$x^{31}_{[6]}$	FDFFFFFF	00011011011001100100011100101111
$x^{31}_{[7]}$	FEFFFFFF	01011000010110011000000001100010
$x^{31}_{[8]}$	FF7FFFFF	01001110101101100100100011001000
$x^{31}_{[9]}$	FFBFFFFF	01111011001100011101001110110101
$x^{31}_{[10]}$	FFDFFFFF	01010111101011111011000000001000
$x^{31}_{[11]}$	FFEFFFFF	01000100101110011011001110001001
$x^{31}_{[12]}$	FFF7FFFF	01110010010011101110110101110101
$x^{31}_{[13]}$	FFFBFFFF	10000100011011010011111000010111
$x^{31}_{[14]}$	FFFDFFFF	00000000001011010111011101101110
$x^{31}_{[15]}$	FFFEFFFF	10110111010101000011101111111001
$x^{31}_{[16]}$	FFFF7FFF	01100100101000000110011000111100
$x^{31}_{[17]}$	FFFFBFFF	01101011110101111001100011111110
$x^{31}_{[18]}$	FFFFDFFF	11001100110110001011111100100110
$x^{31}_{[19]}$	FFFFEFFF	11000000001010101010111101000110
$x^{31}_{[20]}$	FFFFF7FF	00110010101000001101001110100110
$x^{31}_{[21]}$	FFFFFBFF	10000100110101001011000010110000
$x^{31}_{[22]}$	FFFFFDFF	01000111001011110001101101100110
$x^{31}_{[23]}$	FFFFFEFF	10000000011000100101100001011001
$x^{31}_{[24]}$	FFFFFF7F	01001000110010000100111010110110
$x^{31}_{[25]}$	FFFFFFBF	11010011101101010111101100110001
$x^{31}_{[26]}$	FFFFFFDF	10110000000010000101011110101111
$x^{31}_{[27]}$	FFFFFFEF	10110011100010010100010010111001
$x^{31}_{[28]}$	FFFFFFF7	11101101011101010111001001001110
$x^{31}_{[29]}$	FFFFFFFB	00111110000101111000010001101101
$x^{31}_{[30]}$	FFFFFFFD	01110111011011100000000000101101
$x^{31}_{[31]}$	FFFFFFFE	00111011111110011011011101010100

表 7.10: 31 階差分 F_1 3 段

31 次項	31 次項の 16 進数表示	y_i $(i = 31, 30, \cdots, 1, 0)$
$x^{31}_{[0]}$	7FFFFFFF	10110110010011101100100001001000
$x^{31}_{[1]}$	BFFFFFFF	00110001011110111011010111010011
$x^{31}_{[2]}$	DFFFFFFF	10101111010101110000100010110000
$x^{31}_{[3]}$	EFFFFFFF	10111001010001001000100110110011
$x^{31}_{[4]}$	F7FFFFFF	01001110011100100111010111101101
$x^{31}_{[5]}$	FBFFFFFF	01101101100001000001011100111110
$x^{31}_{[6]}$	FDFFFFFF	00101101000000000110111001110111
$x^{31}_{[7]}$	FEFFFFFF	01010100101101111111100100111011
$x^{31}_{[8]}$	FF7FFFFF	00111100011001101010000001100100
$x^{31}_{[9]}$	FFBFFFFF	11111110100110001101011101101011
$x^{31}_{[10]}$	FFDFFFFF	00100110101111111101100011001100
$x^{31}_{[11]}$	FFEFFFFF	01000110101011110010101011000000
$x^{31}_{[12]}$	FFF7FFFF	10100110110100111010000000110010
$x^{31}_{[13]}$	FFFBFFFF	10110000101100001101010010000100
$x^{31}_{[14]}$	FFFDFFFF	01100110000110110010111101000111
$x^{31}_{[15]}$	FFFEFFFF	01011001010110000110001010000000
$x^{31}_{[16]}$	FFFF7FFF	11001000010010001011011001001110
$x^{31}_{[17]}$	FFFFBFFF	10110101110100110011000101111011
$x^{31}_{[18]}$	FFFFDFFF	00001000101100001010111101010111
$x^{31}_{[19]}$	FFFFEFFF	10001001101100111011100101000100
$x^{31}_{[20]}$	FFFFF7FF	01110101111011010100111001110010
$x^{31}_{[21]}$	FFFFFBFF	00010111001111100110110110000100
$x^{31}_{[22]}$	FFFFFDFF	01101110011101110010110100000000
$x^{31}_{[23]}$	FFFFFEFF	11111001001110110101010010110111
$x^{31}_{[24]}$	FFFFFF7F	10100000011001000011110001100110
$x^{31}_{[25]}$	FFFFFFBF	11010111011010111111111010011000
$x^{31}_{[26]}$	FFFFFFDF	11011000110011000010011010111111
$x^{31}_{[27]}$	FFFFFFEF	00101010110000000100011010101111
$x^{31}_{[28]}$	FFFFFFF7	10100000001100101010011011010011
$x^{31}_{[29]}$	FFFFFFFB	11010100100001001011000010110000
$x^{31}_{[30]}$	FFFFFFFD	00101111010001110110011000011011
$x^{31}_{[31]}$	FFFFFFFE	01100010100000000101100101011000

7.4 S-box S_0 と S_1 に対する GF(2^8) 上の補間多項式

ここでは CLEFIA の S-box S_0 と S_1 に対する GF(2^8) 上の補間多項式について、自己評価書 [1] における記載事項を確認し、我々の解析結果を示す。

表 7.11, 7.12 にそれぞれ S-box S_i ($i = 0, 1$): GF(2^8) $\rightarrow$ GF(2^8) の入出力を示す。入出力値は 16 進数表記であり、8 ビットの入力に対して上位 4 ビットと下位 4 ビットがそれぞれ表の行と列に対応し、行と列の交点となる要素がその出力となる。ここで $y = S_i(x)$ として、y を x の補間多項式で表した時、あらゆる既約多項式に対して最小項数は 244 ($i = 0$), 252 ($i = 1$) になると報告されている。

次に我々が補間多項式の項数と係数がゼロである項を解析した結果を表 7.13, 7.14 に示す。GF(2^8) の特性多項式として 8 次の全ての既約多項式について解析した。表の見方は次の通りである。例えば表 7.13 の第 2 行目については特性多項式 $cp(x)$: 0x11b ($= x^8 + x^4 + x^3 + x + 1$) を用いた場合、$S_0$ の補間多項式は 247 個の非ゼロの項を持つことを表し、残りの 9 つの項 (x^{255}, x^{254}, x^{253}, x^{251}, x^{247}, x^{239}, x^{223}, x^{191}, x^{127}) の係数はゼロであることを表している。

表 7.13 より、特性多項式としてどのような 8 次既約多項式を仮定しても、得られる補間多項式は 252 次多項式となることが分かる。また項数は仮定した特性多項式に依存し、その最小値は 244 となる。この値は自己評価書 [1] のそれと一致している。

表 7.14 より、特性多項式としてどのような 8 次既約多項式を仮定しても、得られる補間多項式は 254 次多項式となることが分かる。また項数は仮定した特性多項式に依存し、その最小値は 252 となる。この値は自己評価書 [1] のそれと一致している。

表 7.11: S_0 の入出力

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	57	49	d1	c6	2f	33	74	fb	95	6d	82	ea	0e	b0	a8	1c
1.	28	d0	4b	92	5c	ee	85	b1	c4	0a	76	3d	63	f9	17	af
2.	bf	a1	19	65	f7	7a	32	20	06	ce	e4	83	9d	5b	4c	d8
3.	42	5d	2e	e8	d4	9b	0f	13	3c	89	67	c0	71	aa	b6	f5
4.	a4	be	fd	8c	12	00	97	da	78	e1	cf	6b	39	43	55	26
5.	30	98	cc	dd	eb	54	b3	8f	4e	16	fa	22	a5	77	09	61
6.	d6	2a	53	37	45	c1	6c	ae	ef	70	08	99	8b	1d	f2	b4
7.	e9	c7	9f	4a	31	25	fe	7c	d3	a2	bd	56	14	88	60	0b
8.	cd	e2	34	50	9e	dc	11	05	2b	b7	a9	48	ff	66	8a	73
9.	03	75	86	f1	6a	a7	40	c2	b9	2c	db	1f	58	94	3e	ed
a.	fc	1b	a0	04	b8	8d	e6	59	62	93	35	7e	ca	21	df	47
b.	15	f3	ba	7f	a6	69	c8	4d	87	3b	9c	01	e0	de	24	52
c.	7b	0c	68	1e	80	b2	5a	e7	ad	d5	23	f4	46	3f	91	c9
d.	6e	84	72	bb	0d	18	d9	96	f0	5f	41	ac	27	c5	e3	3a
e.	81	6f	07	a3	79	f6	2d	38	1a	44	5e	b5	d2	ec	cb	90
f.	9a	36	e5	29	c3	4f	ab	64	51	f8	10	d7	bc	02	7d	8e

表 7.12: S_1 の入出力

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	6c	da	c3	e9	4e	9d	0a	3d	b8	36	b4	38	13	34	0c	d9
1.	bf	74	94	8f	b7	9c	e5	dc	9e	07	49	4f	98	2c	b0	93
2.	12	eb	cd	b3	92	e7	41	60	e3	21	27	3b	e6	19	d2	0e
3.	91	11	c7	3f	2a	8e	a1	bc	2b	c8	c5	0f	5b	f3	87	8b
4.	fb	f5	de	20	c6	a7	84	ce	d8	65	51	c9	a4	ef	43	53
5.	25	5d	9b	31	e8	3e	0d	d7	80	ff	69	8a	ba	0b	73	5c
6.	6e	54	15	62	f6	35	30	52	a3	16	d3	28	32	fa	aa	5e
7.	cf	ea	ed	78	33	58	09	7b	63	c0	c1	46	1e	df	a9	99
8.	55	04	c4	86	39	77	82	ec	40	18	90	97	59	dd	83	1f
9.	9a	37	06	24	64	7c	a5	56	48	08	85	d0	61	26	ca	6f
a.	7e	6a	b6	71	a0	70	05	d1	45	8c	23	1c	f0	ee	89	ad
b.	7a	4b	c2	2f	db	5a	4d	76	67	17	2d	f4	cb	b1	4a	a8
c.	b5	22	47	3a	d5	10	4c	72	cc	00	f9	e0	fd	e2	fe	ae
d.	f8	5f	ab	f1	1b	42	81	d6	be	44	29	a6	57	b9	af	f2
e.	d4	75	66	bb	68	9f	50	02	01	3c	7f	8d	1a	88	bd	ac
f.	f7	e4	79	96	a2	fc	6d	b2	6b	03	e1	2e	7d	14	95	1d

表 7.13: S_0 の補間多項式の項数と係数がゼロである項

$cp(x)$	項数	係数がゼロである項											
11b	247	255	254	253	251	247	239	223	191	127			
11d	246	255	254	253	251	247	239	223	191	127	2		
12b	245	255	254	253	251	247	239	223	203	194	191	127	
12d	247	255	254	253	251	247	239	223	191	127			
139	247	255	254	253	251	247	239	223	191	127			
13f	247	255	254	253	251	247	239	223	191	127			
14d	247	255	254	253	251	247	239	223	191	127			
15f	247	255	254	253	251	247	239	223	191	127			
163	245	255	254	253	251	247	245	239	235	223	191	127	
165	247	255	254	253	251	247	239	223	191	127			
169	247	255	254	253	251	247	239	223	191	127			
171	244	255	254	253	251	247	239	235	223	191	127	50	45
177	245	255	254	253	251	247	243	239	223	191	172	127	
17b	247	255	254	253	251	247	239	223	191	127			
187	244	255	254	253	251	247	239	223	200	191	127	123	100
18b	246	255	254	253	251	247	239	223	191	127	97		
18d	247	255	254	253	251	247	239	223	191	127			
19f	247	255	254	253	251	247	239	223	191	127			
1a3	245	255	254	253	251	250	247	239	223	191	163	127	
1a9	246	255	254	253	251	247	239	223	191	127	32		
1b1	246	255	254	253	251	247	239	223	191	131	127		
1bd	247	255	254	253	251	247	239	223	191	127			
1c3	247	255	254	253	251	247	239	223	191	127			
1cf	246	255	254	253	251	247	239	234	223	191	127		
1d7	246	255	254	253	251	247	239	223	191	127	1		
1dd	246	255	254	253	251	247	239	223	191	163	127		
1e7	246	255	254	253	251	247	239	227	223	191	127		
1f3	247	255	254	253	251	247	239	223	191	127			
1f5	247	255	254	253	251	247	239	223	191	127			
1f9	246	255	254	253	251	247	239	223	191	145	127		

表 7.14: S_1 の補間多項式の項数と係数がゼロである項

$cp(x)$	項数	係数がゼロである項
11b	253	255 153 57
11d	254	255 219
12b	254	255 30
12d	254	255 198
139	253	255 209 13
13f	254	255 173
14d	252	255 186 59 55
15f	255	255
163	255	255
165	254	255 148
169	255	255
171	254	255 247
177	255	255
17b	255	255
187	254	255 219
18b	255	255
18d	255	255
19f	253	255 198 28
1a3	255	255
1a9	252	255 96 42 17
1b1	253	255 230 48
1bd	253	255 82 1
1c3	255	255
1cf	253	255 234 121
1d7	253	255 209 131
1dd	252	255 248 241 65
1e7	255	255
1f3	254	255 87
1f5	254	255 141
1f9	255	255

第8章　関連鍵攻撃

鍵関連攻撃の実装の為には、鍵処理部の差分特性の把握が必要である。128 ビット鍵の場合、鍵処理部は、データ攪拌部と同じ、4 系列 Type2 一般化 Feistel 構造 12 段であり、データ攪拌部の差分攻撃耐性評価がそのまま使用できる。192 及び 256 ビット鍵で使用される鍵処理部は、8 系列 Type2 一般化 Feistel 構造 10 段であり、ここでは、その最大差分特性確率の上界を、第 8.1 節で述べる。その値は、$2^{-151.72}$ である。128,192,256 ビット鍵、何れも、十分小さな最大差分特性確率であり、関連鍵攻撃に対する耐性に問題は無いと考える。

また、鍵処理部の高階差分特性を、第 8.2 節で評価し、192、256 ビット鍵の場合、フルラウンドの 10 段に渡る 32 階の高階差分特性（飽和特性）が存在する事、9 段に対し、8 階の高階差分特性（256 種類の鍵の組）が存在する事を示した。この特性が直接に攻撃に結びつくとは、考えにくいが、CLEFIA をそのまま、ハッシュ関数の構成部品として使う場合、注意が必要であろう。なお、128 ビット鍵の場合、このような特性は存在しない。

8.1 鍵スケジュール部の差分特性

本節では関連鍵攻撃耐性評価の一環として、鍵スケジュール部の差分特性確率 (DCP) の上界を評価する。一般に DCP の上界が十分に小さければ、差分特性を利用した関連鍵攻撃に対して安全であると考えられる。図 8.1 に 192/256bit 秘密鍵の鍵スケジュール部における非線形関数部分 $GFN_{8,10}$(等価変形を適用) を示す。データ線は 32bit で構成されていて、秘密鍵 k_i ($i = 0, 1, \cdots, 7$) を入力にとり、中間鍵 ℓ_i ($i = 0, 1, \cdots, 7$) を出力する。ここで $k_0 = k'_0$ であり、$\ell_0 = \ell'_0$ である。また x_j と y_j ($j = 0, 1, \cdots, 39$) はそれぞれ F_i 関数 ($i = 0, 1$) への入出力データを表す。ここで $x_j = x'_j$ ($j = 0, 1, 2, 3, 4$) である。鍵スケジュール部において図 8.1 以外の部分は線形関数であるので、図 8.1 の DCP の上界は、鍵スケジュール部の DCP の上界と等価である。故にここでは図 8.1 の DCP の上界を評価する。尚、128bit 秘密鍵の鍵スケジュール部における非線形関数部分 $GFN_{4,12}$ はラウンド構造がデータ処理部と同一である。従って $GFN_{4,12}$ の DCP の上界はデータ処理部の差分特性の解析結果から分かる。具体的には表 3.6 より、$GFN_{4,12}$ の DCP の上界は自己評価書では $2^{-130.76}$ であり、本研究結果では $2^{-144.39}$ である。いずれの評価も (DCP の上界) $< 2^{-\text{秘密鍵長}}$ であり、128bit 秘密鍵の鍵スケジュール部は差分特性を利用した関連鍵攻撃に対して安全であると考えられる。評価精度については、自己評価書よりも本研究の方が高い。これは本研究では 2 種類の S-box (S_0, S_1) の差分確率の違いを評価しているためである。

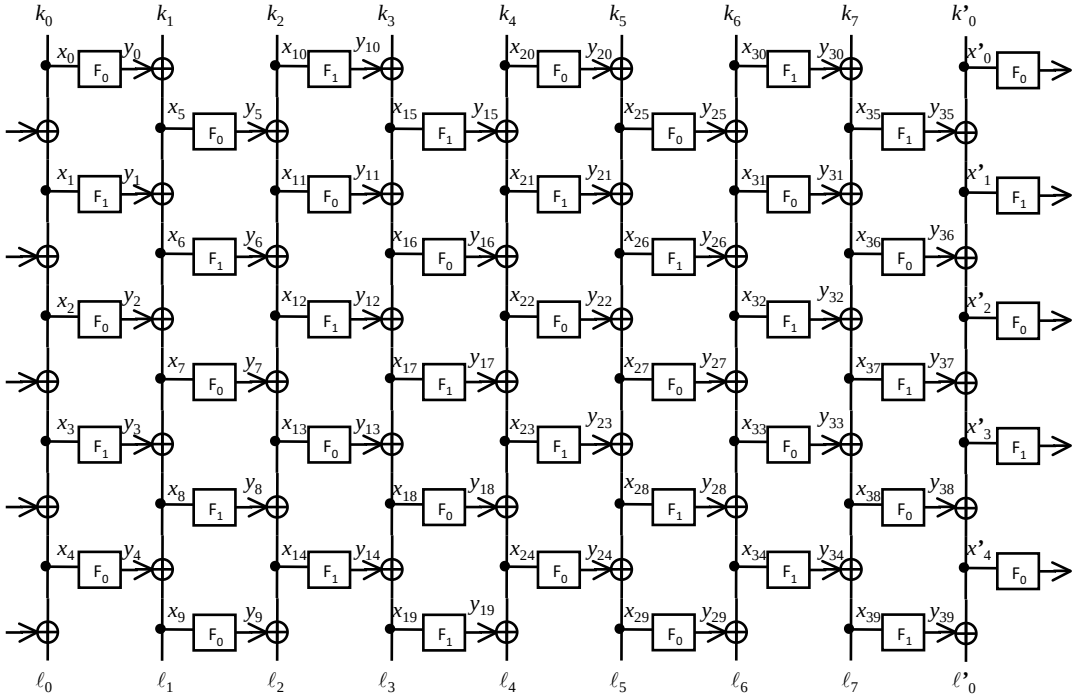


図 8.1: 等価変形した $GFN_{8,10}$.

8.1.1 $GFN_{8,10}$ の DCP の上界評価について

データ処理部の DCP の探索手法と同様に、ここでも 8bit truncation を用いてバイト単位の差分の有無に着目し、DSM の効果を織り込み、DCP の上界を Vterbi アルゴリズムを用いて探索

する。但しデータ処理部とは異なり、鍵スケジュール部では全ての F_i 関数内において独立鍵の加算が存在しない。しかしながら本研究では評価の簡単化のため、全ての F_i 関数内において独立鍵の加算が存在すると仮定する。また、データ処理部の場合と同様に、入力 k_i から出力 ℓ_i へ向かって探索を進めようと考えると、遷移状態として 2 ラウンド分の F_i 関数入力バイト差分 Δx_i(計 8 個) とそれ以前の 2 ラウンド分の F_i 関数入力差分のバイト重み $|\Delta x_i|$(計 8 個)、つまり ($|\Delta x_i|$, $|\Delta x_{i+10}|$, $|\Delta x_{i+20}|$, $|\Delta x_{i+30}|$, $|\Delta x_{i+5}|$, $|\Delta x_{i+15}|$, $|\Delta x_{i+25}|$, $|\Delta x_{i+35}|$, Δx_{i+1}, Δx_{i+11}, Δx_{i+21}, Δx_{i+31}, Δx_{i+6}, Δx_{i+16}, Δx_{i+26}, Δx_{i+36}) が必要となる。Δx_i については 0 から 0xf まで全 16 通りあり、$|\Delta x_i|$ については 0 から 4 まで全 5 通りある。従って遷移状態については全部で $2^{50.58}$ ($= 16^8 \times 5^8$) 通り存在することとなる。このような莫大な数の状態を現実的な時間で探索することは困難であるため、別の探索方法を検討する。

8.1.2 $GFN_{8,10}$ の DCP の上界探索手法

ここでは $GFN_{8,10}$ の DCP の上界を現実的な時間で探索する手法を述べる。初めに探索の概要を述べる。まず第一に、図 8.1 において探索開始状態として (Δx_0, Δx_1, Δx_2, Δx_3, Δx_4) の縦に 5 つ並ぶ変数を状態にとり、列に沿って右方向に探索を進め (DCP が高くなるように遷移し)、($\Delta x'_0$, $\Delta x'_1$, $\Delta x'_2$, $\Delta x'_3$, $\Delta x'_4$) を探索終了状態とする。尚、$\Delta x_i = \Delta x'_i$ ($i = 0, 1, 2, 3, 4$) なので開始状態と終了状態は同一である。図 8.2 に探索の詳細を示す。

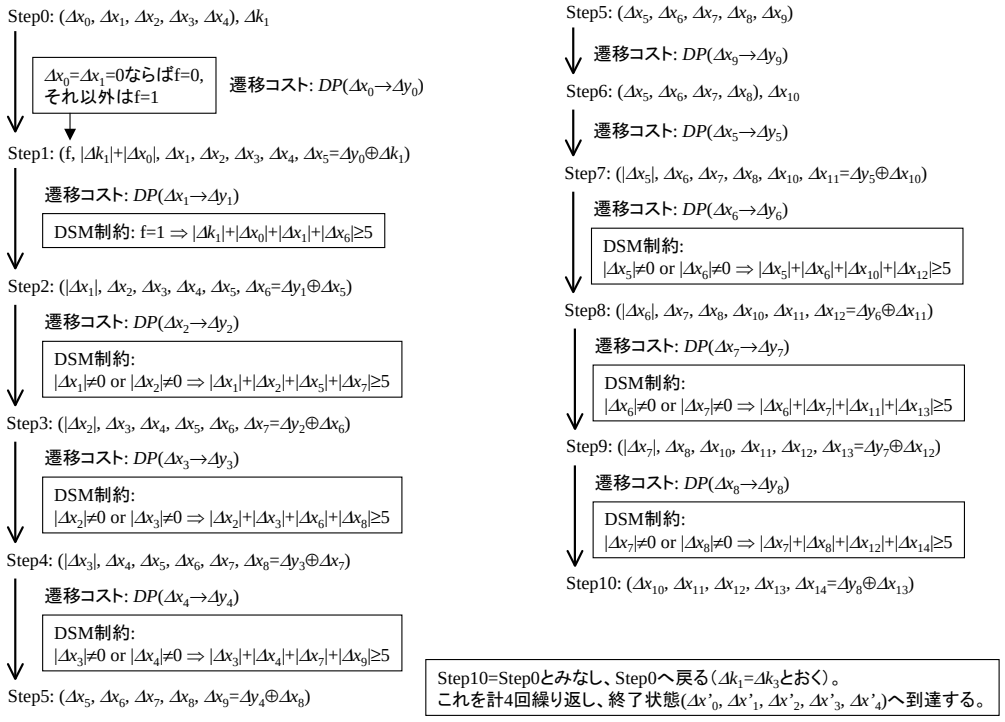


図 8.2: $GFN_{8,10}$ の DCP の上界探索法.

Step0 では状態 (Δx_0, Δx_1, Δx_2, Δx_3, Δx_4) と Δk_1 から Δx_5 を生成し、Step1 へ遷移する。こ

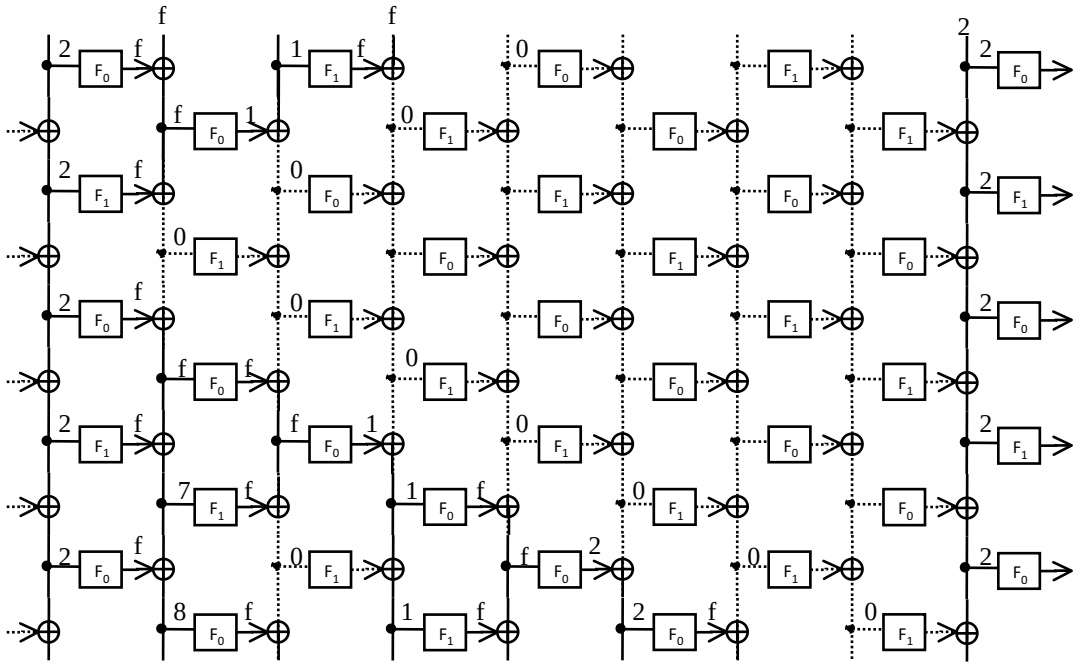


図 8.3: $GFN_{8,10}$ の DCP の上界探索結果.

のときの遷移コストは差分確率 $DP(\Delta x_0 \to \Delta y_0)$ である。また Step2 への遷移時に必要となる付帯情報として f とバイト重みの和 $|\Delta k_1|+|\Delta x_0|$ (5 以上の場合は 5 を代入) を生成する。Step1 から Step2 へ遷移するためには新たに Δx_6 を生成する。この時の遷移コストは $DP(\Delta x_1 \to \Delta y_1)$ である。また DSM の特性により次式に示す制約が生じる。

$$(\Delta x_0 \neq 0 \text{ or } \Delta x_1 \neq 0) \ \to \ |\Delta k_1| + |\Delta x_0| + |\Delta x_1| + |\Delta x_6| \geq 5. \tag{8.1}$$

つまり Step1 から Step2 への遷移においては式 (8.1) を満たす遷移のみが許される。以降の Step においても同様にして遷移を繰り返し、Step40 にて終了状態 ($\Delta x'_0$, $\Delta x'_1$, $\Delta x'_2$, $\Delta x'_3$, $\Delta x'_4$) に到達する。本探索手法においては、全ての Step の中で Step1 の状態数が $2^{23.59}$ $(= 2 \times 6 \times 16^5)$ で最大となるが、入力 k_i から出力 ℓ_i へ向かって探索を進める場合の状態数 $2^{50.58}$ と比べて十分小さく、現実的な時間で探索することが可能となる。一方、本探索アルゴリズムでは開始状態 (全 2^{20} 通り) と終了状態を同一にしなければならないという制約があるため、単純には図 8.2 に示した手順を 2^{20}(開始状態全通り) 回繰り返さなければならない。しかしながら、開始状態と終了状態が同一である差分パスを少なくとも 1 つ見つけ、その時の DCP を上界の候補値とし、上界の候補値を用いて開始状態の候補数を絞り込むことにより図 8.2 の手順の繰り返し回数を大幅に削減できる。具体的には上界の候補値を 1 つ用いることにより、開始状態の候補数は 1/2 になると期待されるので、上界の候補値を 20 回更新すれば、候補数は 2^{-20} となり真の上界を発見できると考えられる。

8.1.3 $GFN_{8,10}$ の DCP の上界探索結果

図 8.3 に $GFN_{8,10}$ の DCP の上界を探索した結果の一例を示す。実線は非零差分の伝播を表し、点線は差分伝播がない事 (零差分の伝播) を表す。図中の数値はバイト truncated 差分の 16 進数表

記である。この時の DCP の上界は $2^{-151.72}$ であり、S-box に着目すると S_0 は 17 個が”Active” となっていて、S_1 は 12 個が”Active” となっている。ここから分かることは、「攻撃者は 192/256bit 秘密鍵に対して鍵差分を入力できる場合、確率 $2^{-151.72}$ でラウンド鍵を制御できる。」であるが、現在のところこの特性を利用した攻撃は発見されていない。また自己評価書には示されていないが、より広範囲のラウンド (5 ラウンド) に影響する DSM による制約条件が SONY から新たに示されている [13]。この新たな制約を図 8.3 に適用すると、一箇所のみがこの制約 (次式に示す) を満たしていないことが分かる。

$$|\Delta x_2| + |\Delta x_3| + |\Delta x_4| + |\Delta x_6| + |\Delta x_9| \geq 5 \tag{8.2}$$

図 8.3 では式 (8.2) の左辺が 4 となっている。式 (8.2) を満たすためには $\Delta x_9 = 8$(バイト重み 1) ではなく、例えば $\Delta x_9 = \mathrm{a}$(バイト重み 2) と修正すればよく、DCP は $2^{-4.67}$ 低下する程度である。これは DSM による制約 [13] を満たし、bit 単位でも接続する可能性がある差分パスである。

8.2 鍵処理部の飽和特性

鍵処理部は、128 ビット鍵の場合、データ攪拌部と同じ、4 系列 Type2 一般化 Feistel 構造であり、第 6 章に記した高階差分攻撃（飽和攻撃）耐性評価が使用できる。192 及び 256 ビット鍵で使用される鍵処理部は、8 系列 Type2 一般化 Feistel 構造（$GFN_{8,10}$）であり、ここでは $GFN_{8,10}$ の高階差分特性（飽和特性）を述べる。計算機により、8 階及び 32 階差分特性を評価し以下の特性を発見した。

8.2.1 $GFN_{8,10}$ の飽和特性

8 階差分を用いた飽和特性

8 階差分を用いた場合、9 ラウンド $GFN_{8,10}$ の入出力には以下の関係が見つかった。表記は 6 章と同じく飽和特性の記法である。

(k − 1) ((C C C C) (A C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C))
$\xrightarrow{9r}$ ((U U U U) (U U U U) (B B B B) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U))

(k − 2) ((C C C C) (C C C C) (C C C C) (A C C C) (C C C C) (C C C C) (C C C C) (C C C C))
$\xrightarrow{9r}$ ((U U U U) (U U U U) (U U U U) (U U U U) (B B B B) (U U U U) (U U U U) (U U U U))

(k − 3) ((C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (A C C C) (C C C C) (C C C C))
$\xrightarrow{9r}$ ((U U U U) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U) (B B B B) (U U U U))

(k − 4) ((C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (A C C C))
$\xrightarrow{9r}$ ((B B B B) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U))

入力パターン (A C C C) は (C A C C), (C C A C) 及び (C C C A) に置き換えても出力パターンは変化しない。

32 階差分を用いた飽和特性

32 階差分を用いた場合、10 ラウンド $GFN_{8,10}$ の入出力には以下の関係が見つかった。(k − 5) の飽和特性を図 8.4 に示す。

(k − 5) ((C C C C) (A A A A) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C))
$\xrightarrow{10r}$ ((B B B B) (U U U U) (B B B B) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U))

(k − 6) ((C C C C) (C C C C) (C C C C) (A A A A) (C C C C) (C C C C) (C C C C) (C C C C))
$\xrightarrow{10r}$ ((U U U U) (U U U U) (B B B B) (U U U U) (B B B B) (U U U U) (U U U U) (U U U U))

(k − 7) ((C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (A A A A) (C C C C) (C C C C))
$\xrightarrow{10r}$ ((U U U U) (U U U U) (U U U U) (U U U U) (B B B B) (U U U U) (B B B B) (U U U U))

(k − 8) ((C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (C C C C) (A A A A))
$\xrightarrow{10r}$ ((B B B B) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U) (B B B B) (U U U U))

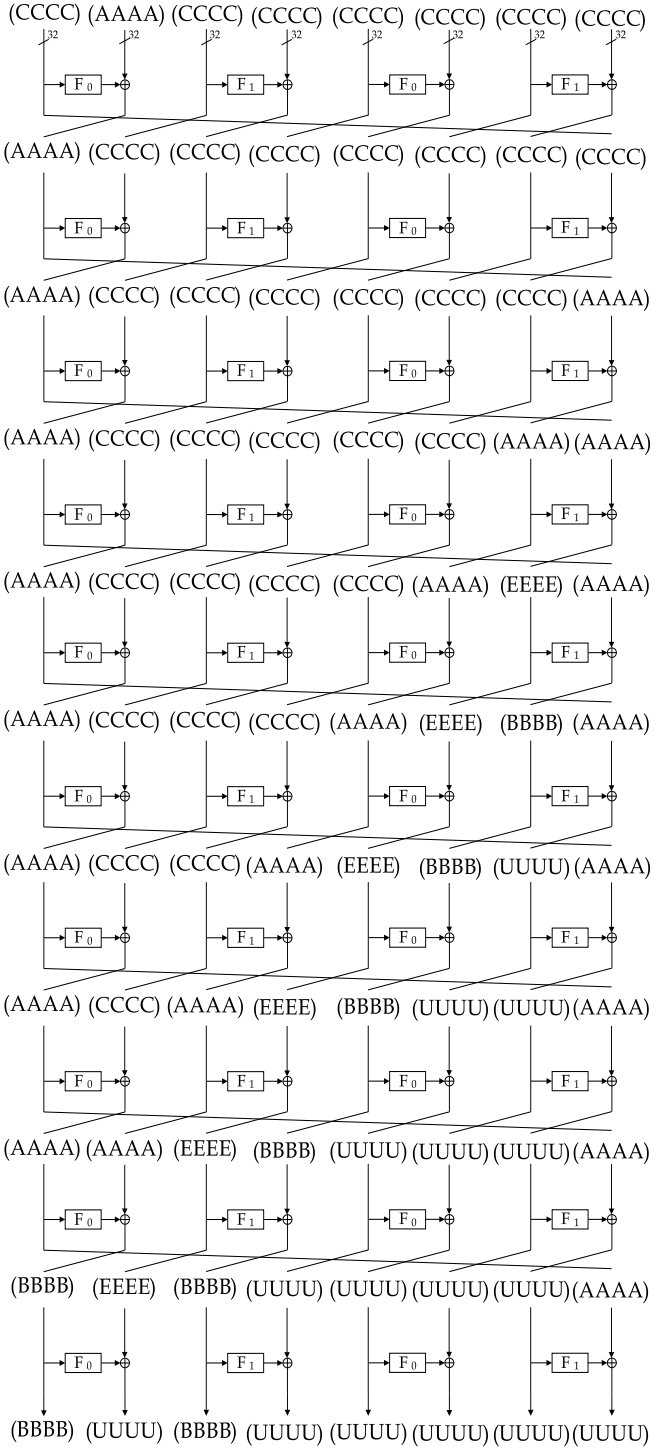


図 8.4: 10 ラウンド $GFN_{8,10}$ の飽和特性

更に、8 系列の Type-2 一般化 Feistel 構造において、10 ラウンドの飽和特性は 4 ラウンド拡張可能であり、次の 14 ラウンドの飽和特性が得られる。

(III) ((A A A A) (A A A A) (C C C C) (C C C C) (C C C C) (A A A A) (A A A A) (A A A A))
$\xrightarrow{14r}$ ((B B B B) (U U U U) (B B B B) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U))

(VI) ((A A A A) (A A A A) (A A A A) (A A A A) (C C C C) (C C C C) (C C C C) (A A A A))
$\xrightarrow{14r}$ ((U U U U) (U U U U) (B B B B) (U U U U) (B B B B) (U U U U) (U U U U) (U U U U))

(V) ((C C C C) (A A A A) (A A A A) (A A A A) (A A A A) (A A A A) (C C C C) (C C C C))
$\xrightarrow{14r}$ ((U U U U) (U U U U) (U U U U) (U U U U) (B B B B) (U U U U) (B B B B) (U U U U))

(VI) ((C C C C) (C C C C) (C C C C) (A A A A) (A A A A) (A A A A) (A A A A) (A A A A))
$\xrightarrow{14r}$ ((B B B B) (U U U U) (U U U U) (U U U U) (U U U U) (U U U U) (B B B B) (U U U U))

これまでの結果より、F 関数の構造が SP 構造であり $m \leq 2n$ である ℓ 系列の Type-2 一般化 Feistel 構造において、$\ell \geq 4$ のとき、$\ell + 2$ ラウンド後の ℓ 系列中の 2 系列が B となる飽和特性が存在し、ラウンド拡張を適用することにより $\ell/2$ ラウンド拡張された $(3\ell/2 + 2)$ ラウンドの飽和特性が存在するものと予想される。ここで、ℓ は偶数である。

参考文献

[1] ソニー株式会社, “128 ビットブロック暗号 CLEFIA 自己評価書 Version 1.0,” 平成 22 年 1 月 29 日, http://www.cryptrec.go.jp/topics/cryptrec_20101001_callforattack.html, http://www.sony.co.jp/Products/cryptography/clefia/technical/index.html

[2] A.Biryukov and D.Khovratovich, ”Related-Key Cryptanalysis of the Full AES-192 and AES-256”, ASIACRYPT 2009, LNCS 5912, pp. 1-18, 2009.

[3] 情報処理振興事業協会、通信・放送機構, ”暗号技術評価報告書 (2001 年度版) CRYPTREC Report 2001”, http://www.ipa.go.jp/security/fy13/report/cryptrec/c01.pdf, 2002

[4] “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 (Telecommunications Advancement Organization of Japan), pp. 109-110, June 2000.

[5] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “The 128-bit Blockcipher CLEFIA”, FSE 2007, LNCS 4593, pp.181-195, Springer-Verlag, 2007.

[6] 辻原悦子，茂真紀，洲崎智保，川崎剛嗣，角尾幸保，“CLEFIA の新たな不能差分”，信学技法，vol.108, no38, ISEC2008-3, pp15-22, 2008 年 5 月.

[7] 角尾幸保，辻原悦子，中嶋浩貴，久保博靖，“変形 Feistel 構造を持つブロック暗号の不可能差分”，SCIS2007-4A2-2, 2007.

[8] 角尾幸保，辻原悦子，久保博靖，茂真紀，川崎剛嗣，“一般化 Feistel 構造の飽和特性”，電子情報通信学会論文誌 A，vol.J93-A, No.4, pp269-276, 電子情報通信学会，2010.

[9] J. Daemen, L.R. Knudsen, and V. Rijmen, “The block cipher SQUARE”, FSE’97, LNCS 1267, pp.149-165, Springer-Verlag, 1997.

[10] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, “Improved Cryptanalysis of Rijndael”, in Proceedings of Fast Software Encryption-FSE2000, vol.1987 of Lecture Notes in Computer Science, pp.213-230, Springer, 2001.

[11] K. Hwang, W. Lee, S. Lee, and J. Lim, “Saturation attacks on reduced round Skipjack”, FSE2002, LNCS 2365, pp.100-111, Springer-Verlag, 2002.

[12] Sony Corporation, The 128-bit blockcipher CLEFIA, security and performance evaluations, revision 1.0, June 1(2007), http://www.sony.co.jp/Products/cryptography/clefia/.

[13] 三津田敦司, 白井太三,“CLEFIA の差分攻撃及び線形攻撃に対する安全性評価の更新”. SCIS2011, 2B1-2, (2011.1).

[14] 小林直登, 五十嵐保隆, 金子敏信, “CLEFIA の差分攻撃耐性”,SCIS2011, 2B1-3, (2011.1).

[15] 芝山直喜, 五十嵐保隆, 金子敏信, 半谷精一郎, “共通鍵ブロック暗号 CLEFIA の飽和攻撃耐性評価”, SCIS2011, 2B1-4, (2011.1).

[16] 五十嵐保隆, 金子敏信, “バランス関数の XOR 和における特殊な飽和特性”, SCIS2011, 3B2-1, (2011.1)

付 録 A　資料:委託研究に関わる学会発表論文

1 小林直登, 五十嵐保隆, 金子敏信: “CLEFIA の差分攻撃耐性,” SCIS 2011, 2B1-3, (2011.1)

2 芝山直喜, 五十嵐保隆, 金子敏信, 半谷精一郎: “共通鍵ブロック暗号 CLEFIA の飽和攻撃耐性評価,” SCIS 2011, 2B1-4, (2011.1)

3 五十嵐保隆, 金子敏信: “バランス関数の XOR 和における特殊な飽和特性,” SCIS 2011, 3B2-1, (2011.1)

SCIS 2011 The 2011 Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 25-28, 2011
The Institute of Electronics, Information and Communication Engineers

CLEFIA の差分攻撃耐性
Differential characteristic property of CLEFIA

小林　直登 *　　五十嵐　保隆 *　　金子　敏信 *
Naoto Kobayashi　　Yasutaka Igarashi　　Toshinobu Kaneko

あらまし　CLEFIA は FSE2007 において SONY によって提案された共通鍵ブロック暗号である [1]。自己評価書 [2] では、差分攻撃耐性指標である最大差分特性確率の上界が報告されている。この上界は CLEFIA の特徴である拡散行列切り替え法 (DSM) に着目して導出されているが、CLEFIA のもう一つの特徴である異なる 2 種類の Sbox が使われているという点には着目されていない。本稿ではこれら 2 つの特徴に着目して、自己評価書よりも精密な最大差分特性確率の上界を導出する。結果として 128,192,256bit 秘密鍵の CLEFIA の場合、自己評価書での上界はそれぞれ $2^{-205.48}$,$2^{-256.85}$,$2^{-303.55}$ であるが、我々の評価法ではこの上界が $2^{-227.42}$,$2^{-282.78}$,$2^{-338.46}$ となり、より精密になることを示す。

キーワード　CLEFIA、差分特性確率、拡散行列切り替え法 (MDS)、viterbi 探索

1　はじめに

CLEFIA は FSE2007 において SONY によって提案された共通鍵ブロック暗号である。CLEFIA の自己評価書 [2] では、差分攻撃耐性指標である最大差分特性確率の上界が報告されている。この上界は CLEFIA の特徴である拡散行列切り替え法 (DSM) に着目して導出されているが、 CLEFIA のもう一つの特徴である異なる 2 種類の Sbox が使われているという点には着目されていない。そこで本稿ではこれら 2 つの特徴に着目して、自己評価書よりも精密な最大差分特性確率の上界を導出する。

2　CLEFIA の構造 [4]

2.1　データ処理部

本稿では CLEFIA の構造のうちデータ処理部のみ説明する。データ処理部は 32bit ワード 4 系列の r ラウンド一般化 Feistel 構造をもち、ラウンド数である r は秘密鍵長 128,192,256bit に対してそれぞれ 18,22,26 である。

P, C を 128 ビットの平文、暗号文とし、i ラウンド目で処理される入力データを $P^{(i)}$、出力データを $C^{(i)}$ とする。P=$P^{(i)}$,C=$C^{(i)}$ である。$P^{(i)}$ を 32 ビット毎 4 つに分割したデータを上位から $P^{\{i,0\}}, P^{\{i,1\}}, P^{\{i,2\}}, P^{\{i,3\}}$、同様に $C^{(i)}$ を 32 ビット毎 4 つに分割したデータを上位から $C^{\{i,0\}}, C^{\{i,1\}}, C^{\{i,2\}}, C^{\{i,3\}}$ とする。また、WK_i $(0 \leq i < 4)$ を 32 ビットのホワイトニング鍵、$RK_i (0 \leq i < 2r)$ を 32 ビットのラウンド鍵とする。これを用い暗号化関数 ENC_r は次のように定義される。これを図 1 に示す。

$Step1.\ T_0|T_1|T_2|T_3 \leftarrow$
$\quad P^{\{0,0\}}|(P^{\{0,1\}} \oplus WK_0)|P^{\{0,2\}}|(P^{\{0,3\}} \oplus WK_1)$
$Step2.\ i = 0$ から $r-1$ に対して以下を実行 :
$Step2.1\ T_1 \leftarrow T_1 \oplus F_0(RK_{2i,T_0})$
$\quad T_3 \leftarrow T_3 \oplus F_1(RK_{2i+1}, T_2)$
$Step2.2\ T_0|T_1|T_2|T_3 \leftarrow T_1|T_2|T_3|T_0$
$Step3.\ C^{\{r,0\}}|C^{\{r,1\}}|C^{\{r,2\}}|C^{\{r,3\}} \leftarrow$
$\quad T_3|T_0 \oplus WK_2|T_1|T_2 \oplus WK_3$

2.1.1　F 関数

図 2、図 3 に F 関数の構成を示す。
S_0, S_1 はそれぞれ 8 ビットの入出力の Sbox を表し、M_0, M_1 はそれぞれ 4×4 の MDS 行列を表している。$k_i (i = 0, 1, 2, 3)$ はラウンド鍵を示し、$Rk_i = k_0|k_1|k_2|k_3$ となる。(| はデータの連結を表す。) F_0 は次のように定義される[1]。F_1 は F_0 の定義で、S_0 と S_1 を入れ替え、M_0 を M_1 に置き換えたもので定義される。

* 東京理科大学理工学研究科電気工学専攻, 〒 278-8510 千葉県野田市山崎 2641, Department of Electrical Engineering, Faculty of Science and Technology, Tokyo University of Science, 2641 Yamazaki, Noda, Chiba, 278-8510 JAPAN, j7307058@ed.noda.tus.ac.jp, yasutaka@rs.noda.tus.ac.jp, kaneko@ee.noda.tus.ac.jp

[1] ${}^t a$ はベクトルまたは行列 a の転置を表す

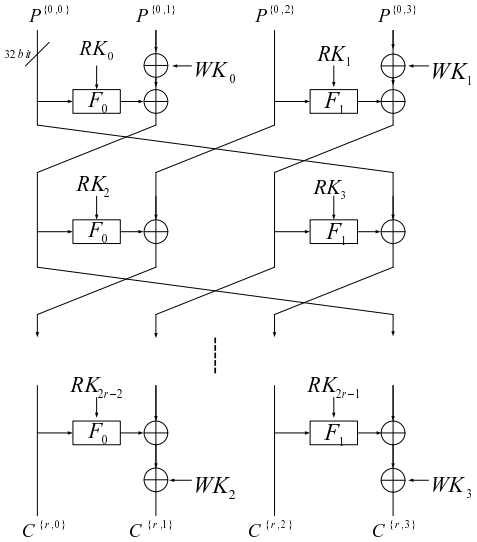


図 1: データ処理部の構造

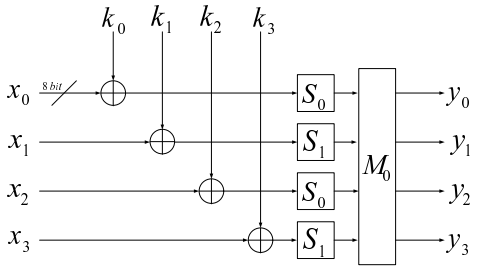


図 2: F0 関数

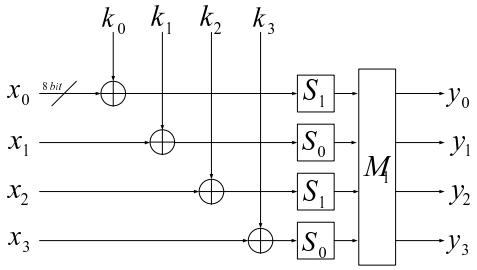


図 3: F1 関数

F_0 :
$Step1.T \leftarrow RK \oplus x$
$Step2.T = T_0|T_1|T_2|T_3, T_i \in \{0,1\}^8$とする
$T_0 \leftarrow S_0(T_0), T_1 \leftarrow S_1(T_1),$
$T_2 \leftarrow S_0(T_2), T_3 \leftarrow S_1(T_3),$
$Step3.y = y_0|y_1|y_2|y_3, y_i \in \{0,1\}^8$とする
${}^t(y_0, y_1, y_2, y_3) = M_0 \ {}^t(T_0, T_1, T_2, T_3)$

2.1.2 Sbox

CLEFIA は 2 種類の S-box を採用している。一つは、ランダムに選択された 4 種の 4 ビット入出力 Sbox をベースとした S_0 であり、もうひとつは、$GF(2^8)$ 上の逆元関数をベースとした S_1 である。

2.1.3 拡散行列

F 関数で用いられる 2 つの MDS 行列 M_0,M_1 は以下で定義される。

$$M_0 = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x06 \\ 0x02 & 0x01 & 0x06 & 0x04 \\ 0x04 & 0x06 & 0x01 & 0x02 \\ 0x06 & 0x04 & 0x02 & 0x01 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} 0x01 & 0x08 & 0x02 & 0x0a \\ 0x08 & 0x01 & 0x0a & 0x02 \\ 0x02 & 0x0a & 0x01 & 0x08 \\ 0x0a & 0x02 & 0x08 & 0x01 \end{pmatrix}$$

行列とベクトル間で実行される乗算は、辞書的順序で最初となる原始多項式 $z^8 + z^4 + z^3 + z^2 + 1$ で定義される GF(2^8) 上の演算として実行される。

3 差分攻撃

差分攻撃とは、差分伝搬を観測することによりある入力差分がどのような出力差分に高確率で伝搬するかどうかを探索する攻撃方法である。ここでは差分攻撃とその耐性を考える際に必要な事柄をまとめる。

3.1 差分確率

関数 $f(x)$ に対して、入力差分 Δx と出力差分 Δy が与えられたとき、差分確率 DP_f は次式のように定義される。ここで n は x のビット長を示す。

$$DP_f(\Delta x, \Delta y) = \frac{\sharp\{x \in \{0,1\}^n \mid f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \quad (1)$$

差分攻撃に対する関数 $f(x)$ の強度は、式 (1) で定義される最大差分確率 DP_{max} で評価され、確率が小さいほど強度が強い。

$$DP_{max} = \max_{\Delta x \neq 0, \Delta y} DP_f(\Delta x, \Delta y) \quad (2)$$

3.2 最大差分特性確率

暗号化関数全体に対しても、式 (2) を用いて最大差分確率を計算して強度指標とすることが正確な評価となるが、それは計算量の問題で困難である。その場合最大差分特性確率を強度指標とする。$f(x)$ が R ラウンド繰り返される暗号系では、i 段目の入力差分を Δx_i、出力差分を Δx_{i+1} としたとき、最大差分特性確率 DCP_{max} は以下の式で定義される。

$$DCP_{max} = \max_{\substack{\Delta x \neq 0, \Delta y \\ \Delta x_1, \cdots, \Delta x_R}} \prod_{i=0}^{R} DP_{f_i}(\Delta x_{i-1}, \Delta x_i) \qquad (3)$$

ここで途中段の差分の伝搬状況 $\Delta_{x_0} \rightarrow \Delta_{x_1} \rightarrow \cdots \rightarrow \Delta_{x_R}$ を差分パスという。

3.3 truncate 差分解析

実際に最大差分特性確率を求めることも計算量的に困難である場合、truncate 差分解析を用いて、最大差分特性確率の上界である最大 truncate 差分確率 DCP_{Tmax} を求める。この手法は、複数 bit の差分の有無を 1bit で表す。すなわち差分有の場合"1"、無の場合を"0"と表記し、"1"の差分を active 差分と呼ぶ。Sbox の bit 幅である 8bit の truncate 解析を行う場合,4 バイト差分はバイト単位に差分の有無を考えるので次式のように 4 ビットで表記される。

$$\Delta = (\Delta_0, \Delta_1, \Delta_2, \Delta_3) \qquad (4)$$

差分の伝播は分岐においては図 4 で示され、排他的論理和においては図 5 のいずれかで示される。ここで

$$\Delta' = (\Delta_0', \Delta_1', \Delta_2', \Delta_3') \qquad (5)$$

$$\Delta\text{"} = (\Delta_0\text{"}, \Delta_1\text{"}, \Delta_2\text{"}, \Delta_3\text{"}) \qquad (6)$$

とすれば truncate 差分ベクトルの成分毎に次式の演算規則が適用される。

$$\Delta_i' = \Delta_i \bar{\oplus} \Delta_i\text{"} \qquad (7)$$

ただし $\bar{\oplus}$ は truncate 差分としての排他的論理和であり下表に従う。

$\bar{\oplus}$	0	1
0	0	1
1	1	0or1

図中の太線は active 差分が伝播しているパスを示している。

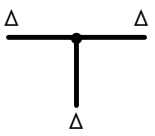


図 4: 分岐における差分の伝播

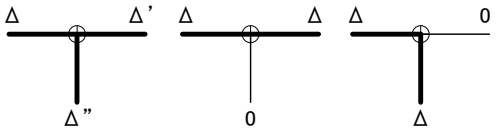


図 5: 排他的論理和における差分の伝播

3.4 active Sbox

Sbox の入力差分が active であれば、その Sbox を active Sbox と呼ぶ。以降、ラウンド t の処理で生じる active Sbox の数を $as^{(t)}$ と書き、active Sbox 数の合計を AS($=\Sigma as^{(t)}$) と表す。さらに AS のうち最小のものを AS_{min} と表す。

4 CLEFIA の解析及び結果

ここでは、CLEFIA の差分攻撃耐性を評価するために最大差分特性確率の上界を 8bit truncate 差分解析により導出するアルゴリズムを説明し、解析結果を示す。

4.1 Sbox

2 種類の Sbox、S_0, S_1 の最大差分確率を計算すると

$$S_0 : DP_{max} = 2^{-4.67} \qquad (8)$$

$$S_1 : DP_{max} = 2^{-6.00} \qquad (9)$$

となった。これは自己評価書と同一の結果である。S_0, S_1 それぞれについて DP_{max} を与える入出力差分の例を表 1,2 に示す。

表 1: $S_0 : DP_{\max}$ を与える入出力差分

Δ_x	0x97	0xE2	0xE3	0x32	0xC6	0xF3	0x8
Δ_y	0xA	0xD	0x1E	0x20	0x30	0x7D	0x7E

表 2: $S_1 : DP_{\max}$ を与える入出力差分

Δ_x	0xA6	0xBE	0xEA	0xA2	0xFB	0xA9	0x2
Δ_y	0x14	0x1	0x13	0x2	0x3	0xFF	0xE9

4.2 F_i 関数の差分特性

F 関数の最大 truncate 差分確率 DP_{FTmax} を評価するために分岐数条件と 2 種類の Sbox の最大差分確率の最大差分確率を与える $(S_0 : DP = 2^{-4.67}, S_1 : DP = 2^{-6.00})$ の出力差分を用いて実際に計算機探索を行った。その結果、存在しえない値があったため S_0 に関しては次点の確率 $DP = 2^{-5.00}$ を考慮して評価した。この結果を表 3.4 に示す。なお、DP_{FTmax} は
$(2^{-4.67})^{AS_0'} \times (2^{-5.00})^{AS_0"} \times (2^{-6.00})^{AS_1}$
と表すことができる。ここで $(AS_0', AS_0", AS_1)$ はそれぞれ差分確率 $(2^{-4.67}, 2^{-5.00}, 2^{-6.00})$ を与える active Sbox の数のことをいう。例として F_0 の (入力バイト差分)=0xB,(出力バイト差分)=0x3 の場合、DP_{FTmax} は $(AS_0' = 2, AS_1 = 1)$ より $2^{-15.34}$ となるが実際の DP_{FTmax} は $(AS_0' = 1, AS_0" = 1, AS_1 = 1)$ より $2^{-15.67}$ となる。

但し、表中の数値は $(log_2 DCP_{FTmax})$ の指数部を示す。

表中の横線はそのような入出力差分の入力がないパターンを示しており、最小分岐数の条件から存在しえないパスである。

4.3 CLEFIA の truncate 差分解析

4.3.1 解析方法

データ処理部における非線形関数は Sbox のみであることから、最大差分特性確率の上界を求めるためには、すべての差分パスに対する DCP_{Tmax} を求めればよい。そこでまず、F_i 関数の入出力差分対に対し差分確率の最大値を求める。次に S_0,S_1 の AS 数を求め、その結果を元に CLEFIA 全体の差分パスに対する特性確率の最大値を探索アルゴリズムである Viterbi アルゴリズムで導出する。

4.3.2 探索アルゴリズム

差分パス及び最大差分特性確率の上界を探索する手法として Viterbi アルゴリズムを用いる。各段の入力を状態と考え、トレリス線図を用いて状態遷移を解析することにより、DCP_{Tmax} とその差分パスを導出できる。解析で用いる状態及び状態遷移を図 6 を用いて説明する。各ラウンドにおける F 関数の 8 ビット truncate 入力差分を $\Delta\mathscr{X}_i$ とし、$\Delta\mathscr{X}_i$ における active バイト数を D_i と表す。$\Delta\mathscr{X}_i$ は 4 ビットベクトルであり D_i は $(0 \le D_i \le 4)$ の値を持つ。

状態遷移は初めの 2 段の状態変数である次式

$$st(0) = (\Delta\mathscr{X}_0, \Delta\mathscr{X}_1, \Delta\mathscr{X}_2, \Delta\mathscr{X}_3) \tag{10}$$

からスタートする。ここではこれをラウンド 0 の状態と呼ぶ。次のラウンド $t = 1$ では状態変数として次式をとる。

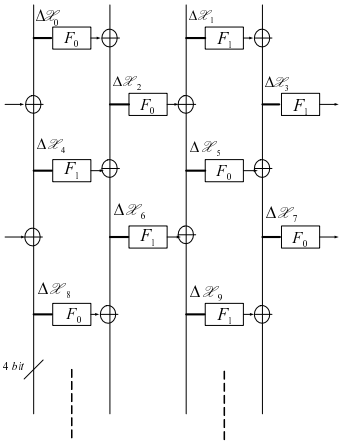


図 6: データ処理部の構造

$$st(1) = (\Delta\mathscr{X}_2, \Delta\mathscr{X}_3, \Delta\mathscr{X}_4, \Delta\mathscr{X}_5, D_0, D_1) \tag{11}$$

$st(0)$ から $st(1)$ への可能な遷移は 3.3 節で述べた truncate 差分伝播規則に従って決定される。また、D_0、D_1 は次節に述べる DSM の効果を状態遷移に反映させるためのものである。さらに次のラウンド $t = 2$ では次式を状態変数にとる。

$$st(2) = (\Delta\mathscr{X}_4, \Delta\mathscr{X}_5, \Delta\mathscr{X}_6, \Delta\mathscr{X}_7, D_0, D_1, D_2, D_3) \tag{12}$$

同様に D_2、D_3 は DSM の効果を反映したものである。$t \ge 2$ においては次式の状態変数をとる。

$$st(t) = (\Delta\mathscr{X}_{2t}, \Delta\mathscr{X}_{2t+1}, \Delta\mathscr{X}_{2t+2}, \Delta\mathscr{X}_{2t+3}, \quad D_{2t-4}, D_{2t-3}, D_{2t-2}, D_{2t-1}) \tag{13}$$

4.3.3 拡散行列 M(DSM) の取り扱い

ここでは CLEFIA の拡散行列の MDS 特性を viterbi 探索にいかに反映させるかを述べる。M_0, M_1 の最小分岐数は、M_0, M_1 の入出力における非零バイト差分の数の総和のうち、最小のものを指す。ただし、入力バイト差分がオールゼロの場合を除く。本稿で用いる $(M_0), (M_1), (M_0|M_1)$ の最小分岐数はいずれの場合も 5 である。ここで $(M_0|M_1)$ は行列 (M_0) と (M_1) の連結である 4 行 8 列の行列である。このような拡散行列が使われている場合、次のような性質[2]が成り立つ。

- 性質 1

  $t = 2a - 1 (a \ge 1)$ のとき
  $D_{2t+2} + D_{2t+1} + D_{2t-2} \ge 5 \ \ (D_{2t+1} \ne 0)$
  $D_{2t+3} + D_{2t} + D_{2t-1} \ge 5 \ \ (D_{2t} \ne 0)$

  $t = 2a (a \ge 1)$ のとき
  $D_{2t+2} + D_{2t} + D_{2t-2} \ge 5 \ \ (D_{2t} \ne 0)$
  $D_{2t+3} + D_{2t+1} + D_{2t-1} \ge 5 \ \ (D_{2t+1} \ne 0)$

[2] 自己評価書 [2] p 24, 性質 2.2、2,3

- 性質 2

$t = 2a - 1 (a \geq 2)$ のとき
$D_{2t+2} + D_{2t+1} + D_{2t-3} + D_{2t-6} \geq 5 \ \ (D_{2t+1} + D_{2t-3} \neq 0)$
$D_{2t+3} + D_{2t} + D_{2t-4} + D_{2t-5} \geq 5 \ \ (D_{2t} + D_{2t-4} \neq 0)$

$t = 2a (a \geq 2)$ のとき
$D_{2t+2} + D_{2t} + D_{2t-4} + D_{2t-6} \geq 5 \ \ (D_{2t} + D_{2t-4} \neq 0)$
$D_{2t+3} + D_{2t+1} + D_{2t-3} + D_{2t-5} \geq 5 \ \ (D_{2t+1} + D_{2t-3} \neq 0)$

性質 1 は F 関数単体の最小分岐数特性に関するものであり、すでに 4.2 節で述べた F 関数差分特性のデータとして組み込まれている。$st(t)$ から $st(t+1)$ の状態遷移では $st(t)$ が truncate 差分 $\Delta\mathscr{X}_{2t}, \Delta\mathscr{X}_{2t+1}, \Delta\mathscr{X}_{2t+2}, \Delta\mathscr{X}_{2t+3}$ を状態として持っているので性質 1 の代りに 4.2 節の DP_{FTmax} を用いる。探索ではこれと図 4,5 に示した差分の伝播条件、性質 2 を満足するパスの中から最良パスを探索する。

例えば、$st(5) = (\Delta\mathscr{X}_{10}, \Delta\mathscr{X}_{11}, \Delta\mathscr{X}_{12}, \Delta\mathscr{X}_{13}, D_6, D_7, D_8, D_9)$ =(0xB,0x7,0x1,0xA,0x0,0x4,0x2,0x1) のとき $\Delta\mathscr{X}_{14}$ は $(\Delta\mathscr{X}_{14} = F(\Delta\mathscr{X}_{12}) \bar{\oplus} \Delta\mathscr{X}_{10})$ で表される。但し $F(\Delta\mathscr{X}_{12})$ は F 関数の出力バイト差分となる。ここで 3.3 節で記した伝播条件から (0x4,0x5,0x6,0x7,0xC,0xD,0xE,0xF) の 8 通りの差分が存在することがわかる。しかしながら、性質 2 より式 (14) を満たさない差分は伝播しないため実際に伝搬する差分は (0x5,0x6,0x7,0xC,0xD,0xE,0xF) の 7 通りとなる。

$$D_{14} \geq (D_{12} + D_8 + D_6) = 2 \tag{14}$$

4.3.4 最大差分特性確率の上界

4.3.2 の探索アルゴリズムを適用し計算機で解析した結果、128bit CLEFIA では $DCP_{Tmax}^{18round} = 2^{-227.42}$、192bit CLEFIA では $DCP_{Tmax}^{22round} = 2^{-282.78}$、256bit CLEFIA では $DCP_{Tmax}^{26round} = 2^{-338.46}$ となった。表 5 に AS_{min} を示し、その確率を自己評価書と本稿とを比較する。なお、確率は $log_2(DCP_{Tmax})$ の値をとる。自己評価書 [2] では S_0 の DCP_{Tmax} のみを考慮した確率であり、本稿では S_0、S_1 の確率の違いを評価したものである。また、DCP_{Tmax} を与えるパスにそって active Sbox 数をカウントしたものが AS_{min} の欄に示してある。結果として、自己評価書と同じ個数の最小 Sbox 数となっている。

表 6 に本稿の結果ををを与える差分パスの一例を示す。

5 結論

本稿では、共通鍵ブロック暗号 CLEFIA の 2 種類の Sbox と DSM を考慮し、差分攻撃耐性評価を行った。Viterbi アルゴリズムを用いて truncate 差分パスを探索し、最大 truncate 差分特性確率の上界を導出した。その結果、1 種類の Sbox のみを考慮した自己評価書の評価と比べ、2 種類の Sbox を考慮した本評価では上界値が低下した。これは本評価が自己評価書よりも精密であることを示している。これにより、CLEFIA の差分攻撃に対するセキュリティーマージンが増加した。

表 5: 本稿での解析と自己評価書での解析による DCP_{Tmax} の比較

r	AS_{min}	自己評価書 [2]	本稿
1	0	0	0
2	1	-4.67	-4.67
3	2	-9.34	-10.67
4	6	-28.02	-32.01
5	8	-37.36	-41.35
6	12	-56.04	-62.69
7	14	-65.38	-73.69
8	18	-84.06	-92.37
9	20	-93.40	-102.04
10	22	-102.74	-113.71
11	24	-112.08	-123.05
12	28	-130.76	-144.39
13	30	-141.10	-155.06
14	34	-158.78	-176.40
15	36	-168.12	-185.07
16	38	-177.46	-196.41
17	40	-186.80	-206.74
18	44	-205.48	-227.42
19	46	-214.82	-237.42
20	50	-233.50	-256.77
21	52	-242.84	-269.43
22	55	-256.85	-282.78
23	56	-261.52	-290.10
24	59	-275.53	-306.45
25	62	-289.54	-320.78
26	65	-303.55	-338.46

表 6: 18 ラウンドにおける本稿の結果を与える差分の一例

t	F 関数の入力差分
1	$(\Delta\mathscr{X}_0 \| \Delta\mathscr{X}_1) =$ 0x01
2	$(\Delta\mathscr{X}_2 \| \Delta\mathscr{X}_3) =$ 0x00
3	$(\Delta\mathscr{X}_4 \| \Delta\mathscr{X}_5) =$ 0x01
4	$(\Delta\mathscr{X}_6 \| \Delta\mathscr{X}_7) =$ 0x0f
5	$(\Delta\mathscr{X}_8 \| \Delta\mathscr{X}_9) =$ 0xb1
6	$(\Delta\mathscr{X}_{10} \| \Delta\mathscr{X}_{11}) =$ 0x67
7	$(\Delta\mathscr{X}_{12} \| \Delta\mathscr{X}_{13}) =$ 0x0b
8	$(\Delta\mathscr{X}_{14} \| \Delta\mathscr{X}_{15}) =$ 0x60
9	$(\Delta\mathscr{X}_{16} \| \Delta\mathscr{X}_{17}) =$ 0x00
10	$(\Delta\mathscr{X}_{18} \| \Delta\mathscr{X}_{19}) =$ 0x60
11	$(\Delta\mathscr{X}_{20} \| \Delta\mathscr{X}_{21}) =$ 0x0b
12	$(\Delta\mathscr{X}_{22} \| \Delta\mathscr{X}_{23}) =$ 0x6a
13	$(\Delta\mathscr{X}_{24} \| \Delta\mathscr{X}_{25}) =$ 0xb5
14	$(\Delta\mathscr{X}_{26} \| \Delta\mathscr{X}_{27}) =$ 0x07
15	$(\Delta\mathscr{X}_{28} \| \Delta\mathscr{X}_{29}) =$ 0x05
16	$(\Delta\mathscr{X}_{30} \| \Delta\mathscr{X}_{31}) =$ 0x00
17	$(\Delta\mathscr{X}_{32} \| \Delta\mathscr{X}_{33}) =$ 0x05
18	$(\Delta\mathscr{X}_{34} \| \Delta\mathscr{X}_{35}) =$ 0x07

参考文献

[1] Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, Tetsu Iwata ”The 128-bit Blockcipher CLEFIA.”FSE 2007, LNCS 4593, pp. 181-195, 2007.

[2] SONY 株式会社,“128bit ブロック暗号 CLEFIA 自己評価書,”

[3] “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 (Telecommunications Advancement Organization of Japan), pp.109-110, 1994.

[4] SONY 株式会社,“128bit ブロック暗号 CLEFIA 暗号技術仕様書,”

表 3: F_0 の差分確率の上界 $[log_2]$

		出力バイト差分															
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
入力バイト差分	0x0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x3	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x5	-	-	-	-	-	-	-	-12	-	-	-	-12	-	-12	-12	-12
	0x6	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x7	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0x8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x9	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xA	-	-	-	-	-	-	-	-9.67	-	-	-	-9.34	-	-9.67	-9.34	-9.34
	0xB	-	-	-	-15.67	-	-15.67	-15.34	-15.34	-	-15.34	-15.67	-15.34	-15.34		-15.34	-15.34
	0xC	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xD	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0xE	-	-	-	-15.34	-	-15.67	-15.34	-15.34	-	-15.34	-15.67	-15.34	-15.67	-15.34	-15.34	-15.34
	0xF	-	-21.34	-21.67	-21.34	-21.34	-21.34	-21.34	-21.34	-21.67	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34

表 4: F_1 の差分確率の上界 $[log_2]$

		出力バイト差分															
		0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
入力バイト差分	0x0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0x1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x3	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.67
	0x5	-	-	-	-	-	-	-	-9.67	-	-	-	-9.67	-	-9.67	-9.67	-9.34
	0x6	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0x7	-	-	-	-15.67	-	-15.67	-15.34	-15.34	-	-15.67	-15.67	-15.34	-15.67	-15.34	-15.34	-15.34
	0x8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-6
	0x9	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xA	-	-	-	-	-	-	-	-12	-	-	-	-12	-	-12	-12	-12
	0xB	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67		-16.67	-16.67
	0xC	-	-	-	-	-	-	-	-10.67	-	-	-	-10.67	-	-10.67	-10.67	-10.67
	0xD	-	-	-	-15.67	-	-15.67	-15.67	-15.34	-	-15.34	-15.67	-15.34	-15.67	-15.34	-15.34	-15.34
	0xE	-	-	-	-16.67	-	-16.67	-16.67	-16.67	-	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67	-16.67
	0xF	-	-21.67	-21.67	-21.34	-21.67	-21.34	-21.34	-21.34	-21.67	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34	-21.34

SCIS 2011 The 2011 Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 25-28, 2011
The Institute of Electronics, Information and Communication Engineers

共通鍵ブロック暗号 CLEFIA の飽和攻撃耐性評価
Security Evaluation of CLEFIA against Saturation Cryptanalysis

芝山 直喜* 五十嵐 保隆† 金子 敏信† 半谷 精一郎*
Naoki Shibayama Yasutaka Igarashi Toshinobu Kaneko Seiichiro Hangai

あらまし　本稿では，FSE2007 において SONY の白井らによって提案された 128bit ブロック暗号 CLEFIA[1] の飽和攻撃耐性評価について報告する．CLEFIA のバイト単位での飽和特性を調査した結果，既存結果 [2] と同じ 32 階差分を用いた 6 ラウンドの飽和特性が存在し，これを 2 ラウンド拡張した，96 階差分を用いた 8 ラウンドの飽和特性が存在することが確認された．この飽和特性を利用することにより，鍵長 128bit の場合，10 ラウンドの CLEFIA に対して，選択平文数 $2^{97.6}$，計算量 2^{98} で飽和攻撃が可能である．鍵長 192，256bit の場合は，それぞれ 11，12 ラウンドの CLEFIA に対して，選択平文数 $2^{98.3}$，$2^{98.8}$，計算量 2^{159}，2^{223} で飽和攻撃が可能である．また，CLEFIA の鍵長 192/256bit の鍵スケジュール部である 8 系列の Type-2 一般化 Feistel 構造において，10 ラウンドの飽和特性が発見された．

キーワード　CLEFIA，一般化 Feistel 構造，飽和特性，安全性評価

1　はじめに

CLEFIA は 2007 年に（株）SONY の白井らによって提案された一般化 Feistel 構造のブロック暗号である [1]．CLEFIA は 1 ラウンドで 2 つの関数を並列に配置した 4 系列の一般化 Feistel 構造を採用しており，データブロック長は 128bit，秘密鍵長は 128，192 及び 256bit をサポートしている．

飽和攻撃は，1997 年に Daemen らによってブロック暗号 SQUARE に対する攻撃として最初に提案された攻撃法 [3] であり，飽和特性を利用し，ラウンド鍵を回復する手法である．CLEFIA には 32 階差分を用いた 6 ラウンドの飽和特性が見つかっており，この 6 ラウンドの飽和特性の 2 ラウンド拡張を行った 8 ラウンドの飽和特性が示されている [2][6]．

本稿は，CLEFIA のバイト単位での飽和特性を調査し，飽和攻撃に対する安全性を評価する．

* 東京理科大学工学研究科電気工学専攻，〒 102-0073　東京都千代田区九段北 1-14-6，Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science, 1-14-6 Kudankita, Chiyoda-ku, Tokyo 102-0073 JAPAN，j4310642@ed.kagu.tus.ac.jp，hangai@ee.kagu.tus.ac.jp
† 東京理科大学理工学研究科電気工学専攻，〒 278-8510　千葉県野田市山崎 2641，Department of Electrical Engineering, Faculty of Science and Technology, Tokyo University of Science, 2641 Yamazaki, Noda, Chiba, 278-8510　JAPAN，yasutaka@rs.noda.tus.ac.jp，kaneko@ee.noda.tus.ac.jp

2　CLEFIA の仕様

2.1　データ攪拌部

図 1 に CLEFIA のデータ攪拌部を示す．CLEFIA は 4 系列の一般化 Feistel 構造で，1 ラウンドで F_0 及び F_1 の 2 種類の F 関数が並列に配置されている．CLEFIA のデータ攪拌部では 128bit の平文 $X_0^{(0)} \| X_1^{(0)} \| X_2^{(0)} \| X_3^{(0)}$ と 32bit のラウンド鍵 $2r$ 個（$RK_0, \cdots, RK_{2r-1}$）及びホワイトニング鍵 4 個（$WK_0, \cdots, WK_3$）から 128bit の暗号文を生成する．なお，ラウンド数 r は鍵長 128，192，256bit それぞれで 18，22，26 である．

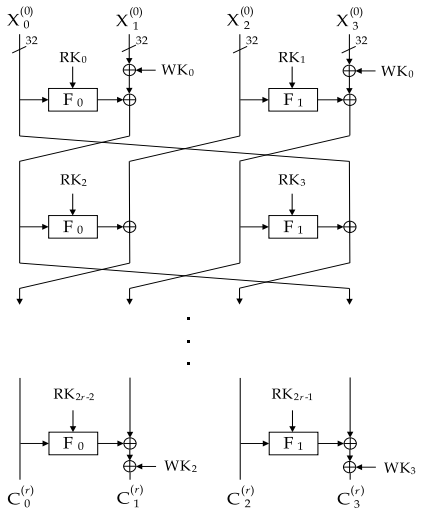
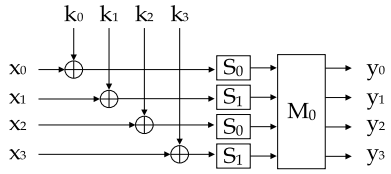
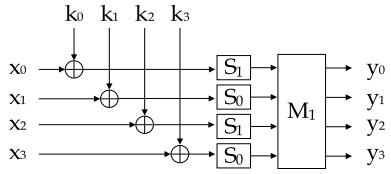


図 1: CLEFIA のデータ攪拌部



(a) F_0関数



(b) F_1関数

図 2: F_0, F_1 関数

図 2 の F_0，F_1 関数において，S_0，S_1 は 8 ビット入出力の S-box であり，2 つの行列 M_0，M_1 は次のように定義される．

$$M_0 = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x06 \\ 0x02 & 0x01 & 0x06 & 0x04 \\ 0x04 & 0x06 & 0x01 & 0x02 \\ 0x06 & 0x04 & 0x02 & 0x01 \end{pmatrix}, \quad (1)$$

$$M_1 = \begin{pmatrix} 0x01 & 0x08 & 0x02 & 0x0a \\ 0x08 & 0x01 & 0x0a & 0x02 \\ 0x02 & 0x0a & 0x01 & 0x08 \\ 0x0a & 0x02 & 0x08 & 0x01 \end{pmatrix}. \quad (2)$$

M_0 と M_1 の最小分岐数は 5 である．なお，これらの行列とベクトル間で実行される乗算は，原始多項式 $z^8 + z^4 + z^3 + z^2 + 1$ で定義される GF(2^8) 上の演算である．

2.2 鍵スケジュール部

鍵スケジュール部における中間鍵 L の生成は CLEFIA の鍵長が 128bit のときの鍵スケジュール部は鍵ホワイトニングのないラウンド数 r' が 12 のデータ攪拌部と同様である．ただし，ラウンド鍵は定数である．また，鍵長 192,256bit のときは 8 系列の一般化 Feistel 構造により中間鍵 L を生成する．なお，ラウンド数 r' は鍵長 192, 256bit それぞれで 10，12 である．図 3 に鍵長 192bit の鍵スケジュール部（以下,「$GFN_{8,10}$」という.）を示す．ホワイトニング鍵 WK_i $(0 \le i < 4)$ 及びラウンド鍵 RK_j $(0 \le j < 2r)$ は秘密鍵 K 及び中間鍵 L を使用し，生成する．なお，その細部についての説明は省略する．

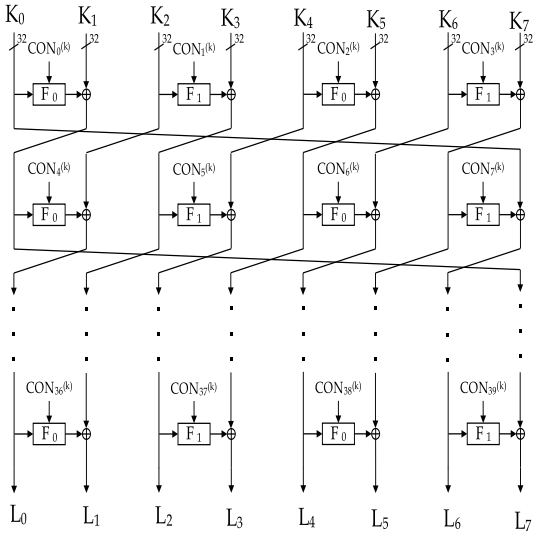


図 3: $GFN_{8,10}$

3 飽和攻撃

飽和攻撃は，1997 年に Daemen らによってブロック暗号 SQUARE に対する攻撃として最初に提案された攻撃法 [3] であり，飽和特性を利用し，ラウンド鍵を回復する手法である．典型的な飽和攻撃は，ブロック暗号のバイト指向構造を利用しており，AES に対しても有効である [4].

3.1 高階差分

定義 入力 $X \in \mathrm{GF}(2)^n$ と鍵 $K \in \mathrm{GF}(2)^s$ から $Y \in \mathrm{GF}(2)^m$ を出力する暗号化関数を $Y = E(X;K)$ で表す．このとき，$E(X;K)$ の X に関する i 階差分は以下のように計算できる．

$$\Delta^{(i)}_{V^{(i)}} E(X;K) = \bigoplus_{\alpha \in V^{(i)}} E(X \oplus \alpha; K) \quad (3)$$

ここで，$V^{(i)}$ は $\mathrm{GF}(2)^n$ 上の i 次元部分空間 $V^{(i)}$ であり，その要素 α を入力差分という（以下，$\Delta^{(i)}_{V^{(i)}}$ を $\Delta^{(i)}$ と表記する）.

性質 1 暗号化関数 $E(X;K)$ の X に関するブール代数次数が N 次であるとき，以下が成り立つ.

$$\begin{cases} \Delta^{(N)} E(X;K) &= const \\ \Delta^{(N+1)} E(X;K) &= 0 \end{cases} \quad (4)$$

ここで，$const$ は定数である．

3.2 飽和特性

定義　$X_i \in \{0,1\}^j$ $(0 \le i < 2^j)$ を 2^j 個の j ビットデータ，X_i の出現度数を Y_i としたとき，X_i を以下の 5 つの状態に分類する．

Constant	:	$\forall i_0, i_1;\ X_{i_0} = X_{i_1}$を満たす場合
All	:	$\forall i_0, i_1;\ i_0 \neq i_1$ならば，$X_{i_0} \neq X_{i_1}$ を満たす場合
Even/odd	:	$\forall i_0, i_1;\ Y_{i_0} = Y_{i_1} \pmod 2$ を満たす場合
Balance	:	$\bigoplus_i X_i = 0$ を満たす場合
Unknown	:	不明な場合

以下，Constant を C，All を A，Even/odd を E，Balance を B 及び Unknown を U と表記する．

性質 2　暗号化関数 $E : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^s \to \mathrm{GF}(2)^n$ の出力 $Y \in \mathrm{GF}(2)^n$ が特性 C，A，E または B のとき，その n 階差分値は 0 となる．

3.3 攻撃の概要

R 段の暗号化関数 E_R を考える．入力 $X \in \mathrm{GF}(2)^n$ に対して，$(R-1)$ 段目の出力 $Y_{(R-1)}(X) \in \mathrm{GF}(2)^m$ を以下のように表す．

$$Y_{(R-1)}(X) = E_{(R-1)}(X; K_1, K_2, \cdots, K_{(R-1)}) \quad (5)$$

ここで，$K_i \in \mathrm{GF}(2)^s$ は i 段目に入力される副鍵である．また，暗号文 $C \in \mathrm{GF}(2)^n$ より，R 段目の鍵 K_R を用いて，$Y_{(R-1)}$ を求める関数 $\tilde{E}(\cdot) : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^s \to \mathrm{GF}(2)^m$ を以下のように表す．

$$Y_{(R-1)}(X) = \tilde{E}(C(X); K_R) \quad (6)$$

$E_{(R-1)}(\cdot)$ に対し，性質 1 または性質 2 が観測された場合，次式が成立する．

$$\Delta^{(N)} Y_{(R-1)}(X) = 0 \quad (7)$$

このとき，(6) 式及び (7) 式より以下の式が成り立つ．

$$\bigoplus_{a \in V^{(N)}} \tilde{E}(C(X \oplus \alpha); K_R) = 0 \quad (8)$$

(8) 式を攻撃方程式と呼び，(8) 式は最終段の鍵 K_R の推定が正しい場合は確率 1 で成立し，推定が誤りの場合は確率 2^{-m} で成立するので，攻撃者は正しい鍵 K_R を決定することができる．

4 CLEFIA の飽和特性

4.1 自己評価書及び角尾らの飽和特性

自己評価書 [6] には，以下に示す二つの 6 ラウンドの飽和特性，また，そのラウンド拡張 [5] を行った 8 ラウンドの飽和特性が存在することが報告されている．

$$(\mathrm{C}, \mathrm{A}, \mathrm{C}, \mathrm{C}) \xrightarrow{6r} (\mathrm{B}, \mathrm{U}, \mathrm{U}, \mathrm{U})$$
$$(\mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{A}) \xrightarrow{6r} (\mathrm{U}, \mathrm{U}, \mathrm{B}, \mathrm{U})$$
$$(\mathrm{A}_0, \mathrm{C}, \mathrm{A}_1, \mathrm{A}_2) \xrightarrow{8r} (\mathrm{B}, \mathrm{U}, \mathrm{U}, \mathrm{U})$$
$$(\mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2, \mathrm{C}) \xrightarrow{8r} (\mathrm{U}, \mathrm{U}, \mathrm{B}, \mathrm{U})$$

ここで，$\mathrm{A}_0 \parallel \mathrm{A}_1 \parallel \mathrm{A}_2$ は All 状態の 96bit である．

角尾らは，F 関数の構造が SP 構造であり，かつ，$m \le 2n$ である 4 系列の Type-2 一般化 Feistel 構造には以下に示す二つの 6 ラウンドの飽和特性，また，そのラウンド拡張を行った 8 ラウンドの飽和特性が存在すると予想している [2]．なお，m は S-box のビット長，n は S-box の個数である．

$$(\mathrm{C}, \mathrm{A}, \mathrm{C}, \mathrm{C}) \xrightarrow{6r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$$
$$(\mathrm{C}, \mathrm{C}, \mathrm{C}, \mathrm{A}) \xrightarrow{6r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$$
$$(\mathrm{A}_0, \mathrm{A}_1, \mathrm{C}, \mathrm{A}_2) \xrightarrow{8r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$$
$$(\mathrm{C}, \mathrm{A}_0, \mathrm{A}_1, \mathrm{A}_2) \xrightarrow{8r} (\mathrm{B}, \mathrm{U}, \mathrm{B}, \mathrm{U})$$

CLEFIA のデータ攪拌部は 4 系列の Type-2 一般化 Feistel 構造であり，1 ラウンドに二つの異なる F 関数を持っている．また，F 関数の構造はどちらも SP 構造であり，$m \le 2n$ である．したがって，CLEFIA においても，同様の飽和特性が存在する．さらに，角尾らは F 関数の構造が SP 構造であり $m \le 2n$ である 6 系列以上の Type-2 一般化 Feistel 構造においても，同様の飽和特性が存在すると予想している．

4.2 計算機を用いた調査

次式に示すように CLEFIA のデータ攪拌部において，128bit の入力 X を 4 つのブロック $X_i \in \mathrm{GF}(2)^{32}$ $(1 \le i \le 4)$ に分割し，さらに，1 ブロックを 8bit の 4 つのサブブロック $X_{ij} \in \mathrm{GF}(2)^8$ $(1 \le j \le 4)$ に分割する．

$$X = (X_1, X_2, X_3, X_4),$$

$$X_i = (X_{i1}, X_{i2}, X_{i3}, X_{i4}).$$

サブブロックに 8 階差分，16 階差分，24 階差分及び 32 階差分を入力し，バイト単位での飽和特性を計算機を用い，調査した．なお，8 階差分及び 16 階差分はすべての入力パターン，24 階差分及び 32 階差分についてはブロックごとのすべての入力パターンに対し，調査を行った．また，$GFN_{8,10}$ に対し，8 階差分及び 32 階差分を用い，同様の調査を行った．

4.2.1 データ攪拌部

8 階差分，16 階差分及び 24 階差分を用いた飽和特性　8 階差分を用いた場合，5 ラウンド CLEFIA の入出力には以下の関係が見つかった．

(d – 1) ((CCCC) (ACCC) (CCCC) (CCCC))
$\xrightarrow{5r}$ ((UUUU) (UUUU) (BBBB) (UUUU))

(d – 2) ((CCCC) (CCCC) (CCCC) (ACCC))
$\xrightarrow{5r}$ ((BBBB) (UUUU) (UUUU) (UUUU))

入力パターン (ACCC) は (CACC), (CCAC) 及び (CCCA) に置き換えても出力パターンは変化しない.

16 階差分及び 24 階差分を用いた場合，8 階差分において 5 ラウンドの飽和特性となる入力パターンを一つのみ含んでいれば，5 ラウンド CLEFIA の出力パターンは (d – 1) または (d – 2) となる．また，8 階差分において 5 ラウンドの飽和特性となる入力パターンを二つ含んでいる場合，以下のような関係が見つかった.

(d – 3) ((CCCC) (ACCC) (CCCC) (ACCC))
$\xrightarrow{5r}$ ((BBBB) (UUUU) (BBBB) (UUUU))

32 階差分を用いた飽和特性　32 階差分を用いた場合，6 ラウンド CLEFIA の入出力には以下の関係が見つかった．(d – 4) の飽和特性を図 4 に示す.

(d – 4) ((CCCC) (AAAA) (CCCC) (CCCC))
$\xrightarrow{6r}$ ((BBBB) (UUUU) (BBBB) (UUUU))

(d – 5) ((CCCC) (CCCC) (CCCC) (AAAA))
$\xrightarrow{6r}$ ((BBBB) (UUUU) (BBBB) (UUUU))

更に，6 ラウンドの飽和特性は 2 ラウンド拡張可能であり，次の 8 ラウンドの飽和特性が得られる [2].

(I) ((AAAA) (AAAA) (CCCC) (AAAA))
$\xrightarrow{8r}$ ((BBBB) (UUUU) (BBBB) (UUUU))

(II) ((CCCC) (AAAA) (AAAA) (AAAA))
$\xrightarrow{8r}$ ((BBBB) (UUUU) (BBBB) (UUUU))

以上より，CLEFIA の 8 ラウンド後の 4 系列中の 2 系列が B となるため，角尾らによって示された飽和特性 [2] と同様の結果が得られた.

4.2.2 $GFN_{8,10}$

8 階差分を用いた飽和特性　8 階差分を用いた場合，9 ラウンド $GFN_{8,10}$ の入出力には以下の関係が見つかった.

(k – 1) ((CCCC) (ACCC) (CCCC) (CCCC)
(CCCC) (CCCC) (CCCC) (CCCC))
$\xrightarrow{9r}$ ((UUUU) (UUUU) (BBBB) (UUUU)
(UUUU) (UUUU) (UUUU) (UUUU))

(k – 2) ((CCCC) (CCCC) (CCCC) (ACCC)
(CCCC) (CCCC) (CCCC) (CCCC))
$\xrightarrow{9r}$ ((UUUU) (UUUU) (UUUU) (UUUU)
(BBBB) (UUUU) (UUUU) (UUUU))

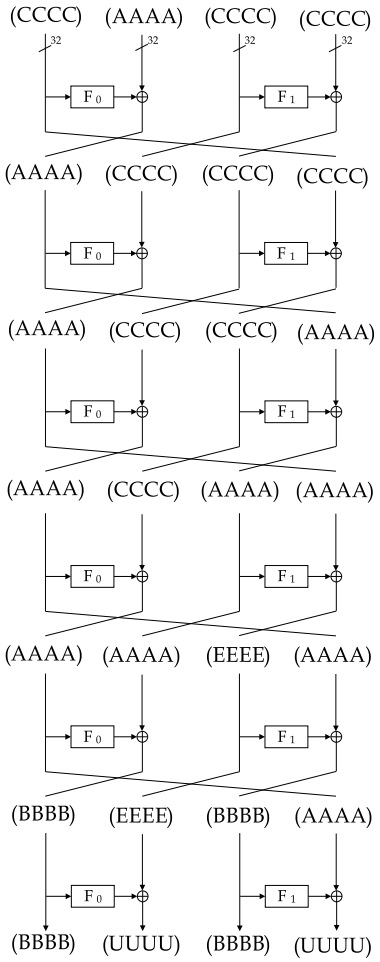


図 4: 6 ラウンド CLEFIA の飽和特性

(k – 3) ((CCCC) (CCCC) (CCCC) (CCCC)
(CCCC) (ACCC) (CCCC) (CCCC))
$\xrightarrow{9r}$ ((UUUU) (UUUU) (UUUU) (UUUU)
(UUUU) (UUUU) (BBBB) (UUUU))

(k – 4) ((CCCC) (CCCC) (CCCC) (CCCC)
(CCCC) (CCCC) (CCCC) (ACCC))
$\xrightarrow{9r}$ ((BBBB) (UUUU) (UUUU) (UUUU)
(UUUU) (UUUU) (UUUU) (UUUU))

入力パターン (ACCC) は (CACC), (CCAC) 及び (CCCA) に置き換えても出力パターンは変化しない.

32 階差分を用いた飽和特性　32 階差分を用いた場合，10 ラウンド $GFN_{8,10}$ の入出力には以下の関係が見つかった．(k – 5) の飽和特性を図 5 に示す.

(k − 5) ((C C C C) (A A A A) (C C C C) (C C C C)
(C C C C) (C C C C) (C C C C) (C C C C))
$\xrightarrow{10r}$ ((B B B B) (U U U U) (B B B B) (U U U U)
(U U U U) (U U U U) (U U U U) (U U U U))

(k − 6) ((C C C C) (C C C C) (C C C C) (A A A A)
(C C C C) (C C C C) (C C C C) (C C C C))
$\xrightarrow{10r}$ ((U U U U) (U U U U) (B B B B) (U U U U)
(B B B B) (U U U U) (U U U U) (U U U U))

(k − 7) ((C C C C) (C C C C) (C C C C) (C C C C)
(C C C C) (A A A A) (C C C C) (C C C C))
$\xrightarrow{10r}$ ((U U U U) (U U U U) (U U U U) (U U U U)
(B B B B) (U U U U) (B B B B) (U U U U))

(k − 8) ((C C C C) (C C C C) (C C C C) (C C C C)
(C C C C) (C C C C) (C C C C) (A A A A))
$\xrightarrow{10r}$ ((B B B B) (U U U U) (U U U U) (U U U U)
(U U U U) (U U U U) (B B B B) (U U U U))

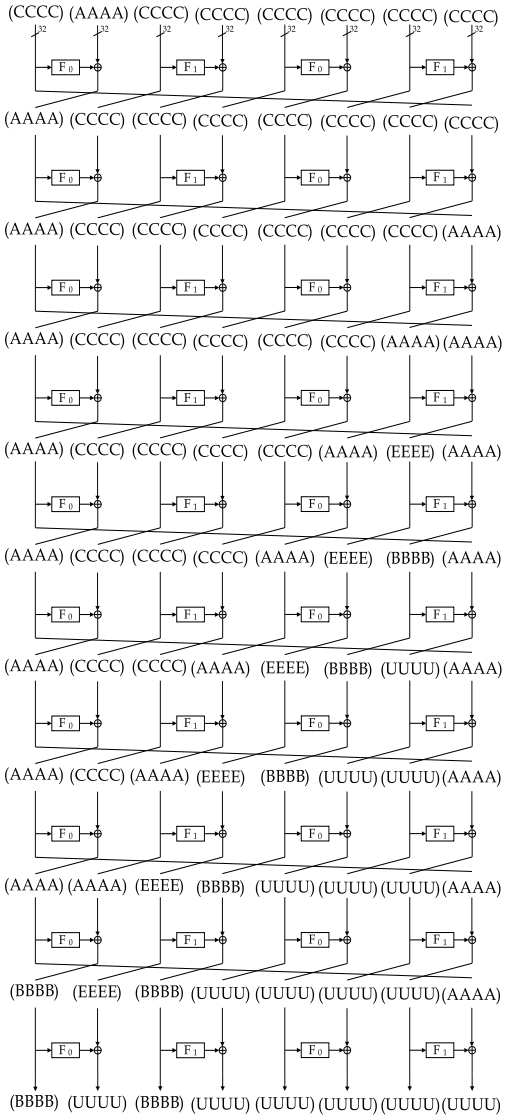


図 5: 10 ラウンド $GFN_{8,10}$ の飽和特性

更に，8 系列の Type-2 一般化 Feistel 構造において，10 ラウンドの飽和特性は 4 ラウンド拡張可能であり，次の 14 ラウンドの飽和特性が得られる．

(III) ((A A A A) (A A A A) (C C C C) (C C C C)
(C C C C) (A A A A) (A A A A) (A A A A))
$\xrightarrow{14r}$ ((B B B B) (U U U U) (B B B B) (U U U U)
(U U U U) (U U U U) (U U U U) (U U U U))

(VI) ((A A A A) (A A A A) (A A A A) (A A A A)
(C C C C) (C C C C) (C C C C) (A A A A))
$\xrightarrow{14r}$ ((U U U U) (U U U U) (B B B B) (U U U U)
(B B B B) (U U U U) (U U U U) (U U U U))

(V) ((C C C C) (A A A A) (A A A A) (A A A A)
(A A A A) (A A A A) (C C C C) (C C C C))
$\xrightarrow{14r}$ ((U U U U) (U U U U) (U U U U) (U U U U)
(B B B B) (U U U U) (B B B B) (U U U U))

(VI) ((C C C C) (C C C C) (C C C C) (A A A A)
(A A A A) (A A A A) (A A A A) (A A A A))
$\xrightarrow{14r}$ ((B B B B) (U U U U) (U U U U) (U U U U)
(U U U U) (U U U U) (B B B B) (U U U U))

以上より，F 関数の構造が SP 構造であり $m \leq 2n$ である 8 系列の Type-2 一般化 Feistel 構造においても，14 ラウンド後の 8 系列中の 2 系列が B となるため，角尾らによって予想された飽和特性が成立することが確認された．また，これまでの結果より，F 関数の構造が SP 構造であり $m \leq 2n$ である ℓ 系列の Type-2 一般化 Feistel 構造において，$\ell \geq 4$ のとき，$\ell + 2$ ラウンド後の ℓ 系列中の 2 系列が B となる飽和特性が存在し，ラウンド拡張を適用することにより $\ell/2$ ラウンド拡張された $(3\ell/2 + 2)$ ラウンドの飽和特性が存在するものと予想される．ここで，ℓ は偶数である．

5 CLEFIA の飽和攻撃

CLEFIA には 8 ラウンドの飽和特性が存在し，その中でも (I) または (II) の 96 階差分を用いた飽和特性を利用したとき，12 ラウンド CLEFIA に対し，飽和攻撃が適用可能である．ここでは，(I) の飽和特性を利用し，CLEFIA に対する飽和攻撃に必要な選択平文数及び計算量の見積もりを行う．なお，9 ラウンド鍵回復に必要な選択平文数と計算量については，自己評価書 [6] と同じであるため，説明を省略する．

10 ラウンド鍵回復　CLEFIA の 9 ラウンド目の F_0 関数を等価変形し，M_0 の位置を図 6 のように配置する．ただし，M_0^{-1} は M_0 の逆行列であり，$M_0^{-1} = M_0$ である．i ラウンド出力の暗号文を $C^{(i)} = (C_0^{(i)}, C_1^{(i)}, C_2^{(i)}, C_3^{(i)})$ としたとき，図 6 及び (8) 式より，以下の攻撃方程式が得られる．

$$\bigoplus \mathrm{F}_0 \left(\mathrm{F}_1 \left(C_2^{(10)}; \mathrm{RK}_{19} \right) \oplus C_3^{(10)}; \mathrm{RK}'_{16} \right) \oplus C_0^{(10)} = 0, \tag{9}$$

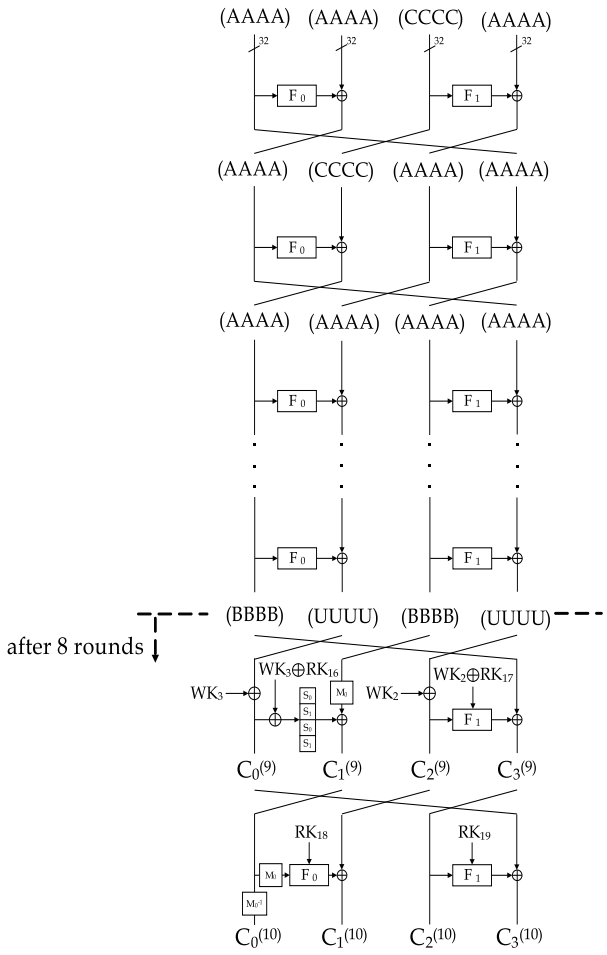


図 6: 10 ラウンド CLEFIA に対する鍵回復攻撃

$$\bigoplus \mathrm{F}_1\left(\mathrm{F}_0\left(C_0^{(10)};\mathrm{RK}_{18}\right)\oplus C_1^{(10)};\mathrm{RK}_{17}^{'}\right)\oplus C_2^{(10)}=0, \tag{10}$$

なお，$\mathrm{RK}_{16}^{'}=\mathrm{WK}_3\oplus\mathrm{RK}_{16}$，$\mathrm{RK}_{17}^{'}=\mathrm{WK}_2\oplus\mathrm{RK}_{17}$ である．(9) 式において，64bit の鍵 $\mathrm{RK}_{16}^{'}$=$\mathrm{RK}_{16(0)}^{'}\parallel\mathrm{RK}_{16(1)}^{'}\parallel\mathrm{RK}_{16(2)}^{'}\parallel\mathrm{RK}_{16(3)}^{'}$ 及び RK_{19} を以下の手順に従い，導出する．

1. $C_0^{(10)}$ において，すべての 2^{96} 個の値の XOR を計算し，その値に行列 M_0^{-1} を乗算した値を $Y=Y_0\parallel Y_1\parallel Y_2\parallel Y_3$ とする．

2. $(\mathrm{A}_0,\mathrm{A}_1,\mathrm{C},\mathrm{A}_2)$ の形を持つ 2^{96} 個の平文を入力し，$C_2^{(10)}\parallel C_3^{(10)}$ の出現度数をカウントし，奇数回出現した 64bit 値のリスト $\mathrm{LIST}(C_2^{(10)},C_3^{(10)})=\mathrm{LIST}(C_2^{(10)})\parallel\mathrm{LIST}(C_3^{(10)})$ を作成する．

3. すべての $l_{C_2^{(10)}}\in\mathrm{LIST}(C_2^{(10)})$，$l_{C_3^{(10)}}\in\mathrm{LIST}(C_3^{(10)})$ と推測した RK_{19} に対し，次式で表される X_i $(0\le i\le 3)$ の出現度数をカウントし，奇数回出現した 8bit 値の $\mathrm{LIST}(X_i)$ を作成する．

$$l_{C_3^{(10)}}\oplus\mathrm{F}_1\left(l_{C_2^{(10)}};\mathrm{RK}_{19}\right)=X, \tag{11}$$
$$X=X_0\parallel X_1\parallel X_2\parallel X_3.$$

4. すべての $l_{X_i}\in\mathrm{LIST}(X_i)$ と観測した $\mathrm{RK}_{16(i)}^{'}$ に対し，

$$\bigoplus S_j\left(l_{X_i}\oplus\mathrm{RK}_{16(i)}^{'}\right)=Y_i\quad(0\le i\le 3) \tag{12}$$

が成立するとき，推測した鍵 $\mathrm{RK}_{16}^{'}$ 及び RK_{19} は正しいと判定する．ここで，j は i が偶数のとき 0，奇数のとき 1 である．

同様に，(10) 式において，鍵 $\mathrm{RK}_{17}^{'}$ 及び RK_{18} を導出する．

(12) 式は 8bit の 4 つの方程式であるから，偽鍵に対し，成立する確率は $\left(2^{-8}\right)^4=2^{-32}$ である．(11) 式において，64bit の鍵 $\mathrm{RK}_{16}^{'}$ 及び RK_{19} の推定を行うには，$3(>\frac{64}{32})$ 組の 96 階差分を用意すれば十分である．よって，解読に必要な選択平文数は $3\cdot 2^{96}\simeq 2^{97.6}$ である．

XOR 及び S-box の計算量を F 関数とみなし，鍵の推定に必要な計算量について考える．手順 1，2 において，Y の算出及び $\mathrm{LIST}(C_2^{(10)},C_3^{(10)})$ の作成にかかる計算量は $2\cdot 2^{96}=2^{97}$（F 関数）である．手順 3，4 において，一つの推測した 32bit の鍵 RK_{19} に対し，64bit 値の $\mathrm{LIST}(C_2^{(10)},C_3^{(10)})$ を用い，X を計算する．次に，X を用い，8bit 値の $\mathrm{LIST}(X_i)$ $(0\le i\le 3)$ を計算し，さらに，$\mathrm{LIST}(X_i)$ を用いて 8bit の鍵 $\mathrm{RK}_{16(i)}^{'}$ の推測を行うため，このときの計算量は $2^{32}\left(2^{64}+2^8\cdot 4+2^8\cdot 4\cdot 2^8\right)\simeq 2^{96}$（F 関数）となる．以上より，64bit の鍵の推定に必要な計算量は $2^{97}+2^{96}\simeq 2^{97.6}$（F 関数）となる．ただし，2 組目以降の鍵の推測に必要な計算量は 1 組目に比べ少ないため，全体の計算量に影響しないものとみなした．ここで，10 ラウンド CLEFIA には 20 個の F 関数が配置されているため，$2^{97.6}$（F 関数）$=\frac{2}{20}\cdot 2^{97.6}\simeq 2^{94.3}$（暗号化）である．また，(10) 式においても，同じ計算量で鍵の推定を行うことが可能であるので，10 ラウンドすべての鍵の推定に必要な計算量は $2\cdot 2^{94.3}=2^{95.3}$（暗号化）よって，暗号文を求める計算量 $2^{97.6}$（暗号化），鍵を推定する計算量 $2^{95.3}$（暗号化）より，解読に必要な計算量は $2^{97.6}+2^{95.3}\simeq 2^{98}$（暗号化）となる．したがって，この攻撃は，鍵長が 128，192 及び 256bit の 10 ラウンド CLEFIA いずれに対し，適用可能である．

作成するリストは 64bit 値の $\mathrm{LIST}(C_2^{(10)},C_3^{(10)})$ 及び 32bit 値の $\mathrm{LIST}(X)$ である．よって，解読に使用するメモリは $2^{64}+2^{32}\simeq 2^{64}$（bit）である．

11/12 ラウンド鍵回復　11/12 ラウンド鍵回復については，10 ラウンド鍵回復と同様の手法を用い，解読に必要

な選択平文数及び計算量の導出する．すなわち，リスト $\mathrm{LIST}(C_2^{(10)}, C_3^{(10)})$，$\mathrm{LIST}(X_i)$ により，それぞれ RK_{19}，$\mathrm{RK}'_{16(i)}$ を推定し，残りの鍵については全数探索により推定する．

11/12 ラウンド鍵回復において，推定する鍵長はそれぞれ 128，192bit であるため，鍵の推定を行うには 96 階差分がそれぞれ $5(> \frac{128}{32})$, $7(> \frac{192}{32})$ 組用意すれは十分である．よって，11/12 ラウンド CLEFIA の解読に必要な選択平文数はそれぞれ $5 \cdot 2^{96} \simeq 2^{98.3}$, $7 \cdot 2^{96} \simeq 2^{98.8}$ である．また，10 ラウンド鍵回復において，64bit の鍵を推定するのに必要な計算量は $2^{94.3}$（暗号化）より，11 ラウンド鍵回復では，この操作を 11 ラウンド目の鍵の総数 2^{64} 回繰り返すので，計算量は $2^{94.3} \cdot 2^{64} \simeq 2^{159}$（暗号化）である．同様に，12 ラウンド鍵回復では $2^{94.3} \cdot 2^{128} \simeq 2^{223}$（暗号化）となる．

CLEFIA に対する飽和攻撃の攻撃可能段数，攻撃に必要な選択平文数及び計算量をまとめたものを表 1 に示す．

表 1: CLEFIA に対する飽和攻撃の結果

ラウンド数	鍵長	選択平文数	計算量
10	128,192,256	$2^{97.6}$	2^{98}
11	192,256	$2^{98.3}$	2^{159}
12	256	$2^{98.8}$	2^{223}

結果として，10 ラウンドのとき，RK'_{16} を 1 バイトごとに導出するため，自己評価書よりも計算量が 2^{24} 程度少なくなることが分かった．また，これらの結果と自己評価書に示された飽和特性を利用した攻撃に必要な選択平文数を比較した場合，自己評価書に示された飽和特性は 8 ラウンド後の 4 系列中の 1 系列のみ B であるので，各ラウンドにおいて，すべての鍵を回復するには，8 ラウンド後の 4 系列中の 2 系列が B となる飽和特性を利用した攻撃に必要な選択平文数よりも約 1.5〜2 倍の選択平文数が必要である．

6 まとめ

CLEFIA のバイト単位での飽和特性を調査した結果，CLEFIA には角尾らの予想した 6 ラウンドの飽和特性が存在し，これにラウンド拡張を適用することにより，8 ラウンドの飽和特性が存在することが確認された．この飽和特性を利用し，CLEFIA に飽和攻撃を適用した結果，鍵長 128bit の場合，10 ラウンドの CLEFIA に対して，選択平文数 $2^{97.6}$，計算量 2^{98} で飽和攻撃が可能である．鍵長 192，256bit の場合は，それぞれ 11，12 ラウンドの CLEFIA に対して，選択平文数 $2^{98.3}$，$2^{98.8}$，計算量 2^{159}，2^{223} で飽和攻撃が可能である．結果として，実際の CLEFIA のラウンド数は鍵長が 128bit の場合は 18，鍵長が 192/256bit の場合はそれぞれ 22，26 であるので，CLEFIA は飽和攻撃に対し，十分な耐性を持つと考えられる．また，$GFN_{8,10}$ において，10 ラウンドの飽和特性が存在することが分かった．この結果より，F 関数の構造が SP 構造であり $m \leq 2n$ である 8 系列の Type-2 一般化 Feistel 構造においても，8 系列中の 2 系列が B となる角尾らによって予想された飽和特性が成立することが確認された．

参考文献

[1] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, “The 128-bit Blockcipher CLEFIA”, FSE 2007, LNCS 4593, pp.181-195, Springer-Verlag, 2007.

[2] 角尾幸保，辻原悦子，久保博靖，茂真紀，川崎剛嗣，“一般化 Feistel 構造の飽和特性”，電子情報通信学会論文誌 A，vol.J93-A, No.4, pp269-276, 電子情報通信学会，2010.

[3] J. Daemen, L.R. Knudsen, and V. Rijmen, “The block cipher SQUARE”, FSE’97, LNCS 1267, pp.149-165, Springer-Verlag, 1997.

[4] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, “Improved Cryptanalysis of Rijndael”, in Proceedings of Fast Software Encryption-FSE2000, vol.1987 of Lecture Notes in Computer Science, pp.213-230, Springer, 2001.

[5] K. Hwang, W. Lee, S. Lee, and J. Lim, “Saturation attacks on reduced round Skipjack”, FSE2002, LNCS 2365, pp.100-111, Springer-Verlag, 2002.

[6] Sony Corporation, The 128-bit blockcipher CLEFIA, security and performance evaluations, revision 1.0, June 1(2007), `http://www.sony.co.jp/Products/cryptography/clefia/`.

SCIS 2011 The 2011 Symposium on Cryptography and Information Security
Kokura, Japan, Jan. 25-28, 2011
The Institute of Electronics, Information and Communication Engineers

バランス関数の XOR 和における特殊な飽和特性
Saturation characteristics of XOR sum of balance functions

五十嵐保隆* 金子敏信 *
Yasutaka Igarashi Toshinobu Kaneko

あらまし 角尾らは (m, n) モデルを定義し、(m, n) モデルの飽和特性に関する予想を述べている。本稿ではその予想が正しいことを証明する。(m, n) モデルとは n 個の異なるバランス関数 (入出力は m bit) の XOR 和 ($\oplus$) を出力する回路モデルであり、バランス関数とはその出力全通りの XOR 和がゼロとなる関数である。角尾らの予想とは次の通りである。(m, n) モデルの出力頻度分布は、$m < 2n$ であれば全て偶数であり、$m = 2n$ であれば、奇数か偶数のいずれか一方のみである。そしてこのような予想が成り立つ関数は特殊なバランス関数と呼ばれている。我々は (m, n) モデルの出力頻度分布に着目し、この分布とアダマール変換、拡大ハミング符号の検査行列を用いて角尾らの予想が正しいことを証明する。

キーワード バランス関数, 飽和特性, アダマール変換, 拡大ハミング符号, 畳み込み

1 (m, n) モデルと角尾らの予想

角尾らは Type-2 一般化 Feistel 構造の飽和特性を検討した論文 [1] 中で (m, n) モデルを定義し、(m, n) モデルの飽和特性に関する予想を述べている。本節では (m, n) モデルの定義と角尾らの予想を示す。

図 1 に (m, n) モデルを示す ($m \geq 2$, $n \geq 1$)。データ線の bit 幅は全て m である。x_i と u_i はそれぞれ関数 g_i の入出力を表し ($i = 0, 1, 2, \cdots, n-1$)、$X = x_0 \,||\, x_1 \,||\, \cdots \,||\, x_{n-1}$ は (m, n) モデルの入力を表す (|| はデータの連結を表す)。(m, n) モデルの出力 y は u_i の XOR 和 ($\oplus$) である。g_i はバランス関数と呼ばれ、その出力 u_i は次式を満たす。

$$\bigoplus_{i=0}^{2^m-1} u_i = 0. \tag{1}$$

次に u_i と y に関する統計量を定義する。初めに入力 $x_i = 0, 1, 2, \cdots, 2^m - 1$ に対する出力 u_i の出現回数の分布 (頻度分布) を $f_i(u)$ とし ($\sum f_i(u) = 2^m$)、$X = 0, 1, 2, \cdots, 2^{mn} - 1$ に対する出力 y の頻度分布を $f_y(y)$ とする ($\sum f_y(y) = 2^{mn}$)。さらに $f_i(u)$ に対して mod2 演算を適用した頻度分布を $f_i^{(2)}(u)$ とし、これを mod2 頻度分布と呼ぶ。つまり $f_i(u)$ が偶数であれば $f_i^{(2)}(u)$ は 0 となり、奇数であれば 1 となる。

次に角尾らの予想を示す。図 1 の (m, n) モデル ($m \geq 2$, $n \geq 1$) において、

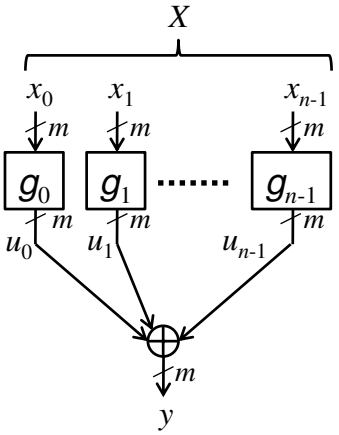


図 1: (m, n) モデル.

[予想 1] $m < 2n$ ならば $f_y(y)$ は任意の y に対して偶数である。
[予想 2] $m = 2n$ ならば $(f_y(0) + f_y(y))$ は任意の y に対して偶数である。

2 角尾らの予想の証明

初めに予想 1 から証明する。畳み込み演算を用いると $f_y(\tau)$ は次式で与えられる。

$f_y(\tau)$

* 東京理科大学理工学部 〒 278-8510 千葉県野田市山崎 2641. Tokyo University of Science, 2641, Yamazaki, Noda, Chiba 278-8510, Japan. yasutaka@rs.noda.tus.ac.jp, kaneko@ee.noda.tus.ac.jp

$$= \sum_{u_0=0}^{2^m-1} \sum_{u_1=0}^{2^m-1} \cdots \sum_{u_{n-2}=0}^{2^m-1} f_0(u_0) f_1(u_1) \cdots f_{n-2}(u_{n-2}) \cdot f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2} \oplus \tau). \quad (2)$$

ここで頻度分布 $f_i(u)$ の特性関数を $\varphi_i(t)$ として、$f_i(u)$ から $\varphi_i(t)$ への変換を H_{2^m} とする。さらに H_{2^m} の逆変換を $\mathrm{H}_{2^m}^{-1}$ とすると、式 (2) は次式で書き直せる。

$$f_y(\tau) = \mathrm{H}_{2^m}^{-1}\big\{\varphi_0(t)\varphi_1(t)\cdots\varphi_{n-2}(t)\varphi_{n-1}(t)\big\}. \quad (3)$$

本稿ではアダマール変換とその逆変換をそれぞれ H_{2^m}, $\mathrm{H}_{2^m}^{-1}$ とする (アダマール変換と畳み込みの関係は節 A に示す)。

次に $f_i^{(2)}(u)$ 及び $f_i(u)$ の性質を考察する。関数 g_i はバランス関数なので、$f_i^{(2)}(u)$ は符号長 2^m, 情報ビット数 $2^m - m - 1$ の拡大ハミング符号の符号語である。従ってその検査行列を $\mathcal{H}_{2^m}$ とすると次式が成り立つ。

$$\mathcal{H}_{2^m}\mathbf{f}_i^{(2)} = \mathbf{0}, \quad \mathbf{f}_i^{(2)} = (f_i^{(2)}(0), f_i^{(2)}(1), \cdots, f_i^{(2)}(2^m-1))^t. \quad (4)$$

$(\cdot)^t$ はベクトル及び行列の転置を表す。式 (4) では各々の要素の加算は XOR である。式 (4) から直ちに次式が導かれる。

$$\mathcal{H}_{2^m}\mathbf{f}_i = (\text{全ての要素が偶数であるベクトル}), \quad \mathbf{f}_i = (f_i(0), f_i(1), \cdots, f_i(2^m-1))^t. \quad (5)$$

尚、式 (5) において、各々の要素の加算は算術加算である。これより、特性関数 $\varphi_i(t) = \mathrm{H}_{2^m} f_i(u)$ の性質として次が導かれる (導出過程は節 B に示す)。

$$\varphi_i(t) \text{ は } 4 \text{ を因数に持つ.} \quad (6)$$

これより、式 (3) において $\mathrm{H}_{2^m}^{-1} = \frac{1}{2^m}\,\mathrm{H}$ に注意すれば、$f_y(\tau)$ は $4^n/2^m = 2^{2n-m}$ を因数に持つことが分かる。従って、$m < 2n$ ならば $f_y(\tau)$ は任意の τ に対して偶数である。

(予想 1 の証明終わり)

次に予想 2 を証明する。尚、$y = 0$ の場合、予想 2 は自明なので証明は省く。畳み込み演算を用いると $f_y(0) + f_y(\tau)$ は次式で与えられる。

$$f_y(0) + f_y(\tau) = \sum_{u_0=0}^{2^m-1} \sum_{u_1=0}^{2^m-1} \cdots \sum_{u_{n-2}=0}^{2^m-1} f_0(u_0) f_1(u_1) \cdots f_{n-2}(u_{n-2}) \cdot \big\{f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2}) + f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2} \oplus \tau)\big\}. \quad (7)$$

これを等価変形すると次式となる。

$$f_y(0) + f_y(\tau) = \sum_{u_0 \in U(\tau)} \sum_{u_1 \in U(\tau)} \cdots \sum_{u_{n-2} \in U(\tau)} \{f_0(u_0) + f_0(u_0 \oplus \tau)\}\{f_1(u_1) + (f_1(u_1 \oplus \tau)\} \cdots \cdot \{f_{n-2}(u_{n-2}) + f_{n-2}(u_{n-2} \oplus \tau)\} \cdot \big\{f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2}) + f_{n-1}(u_0 \oplus u_1 \oplus \cdots \oplus u_{n-2} \oplus \tau)\big\}. \quad (8)$$

ここで集合 $U = \{0, 1, 2, \cdots, 2^m - 1\}$ とすると、τ に依存する集合 $U(\tau)$ は $\{U(\tau), U(\tau) \oplus \tau\} = U$ を満たす。

次に $F_i(u) = (f_i(u) + f_i(u \oplus \tau))$ $(u \neq u \oplus \tau)$ とおき、そのアダマール変換を $\varphi_{F_i}(t) = \mathrm{H}_{2^{m-1}} F_i(u)$ とすると、性質 (6) から次の性質が導かれる。

$$\varphi_{F_i}(t) \text{ は } 4 \text{ を因数に持つ.} \quad (9)$$

$\varphi_{F_i}(t)$ を用いると、$f_y(0) + f_y(\tau)$ は次式で与えられる。

$$f_y(0) + f_y(\tau) = \frac{1}{2^{m-1}} \mathrm{H}_{2^{m-1}} \varphi(t), \quad \varphi(t) = \varphi_{F_0}(t)\varphi_{F_1}(t)\cdots\varphi_{F_{n-1}}(t). \quad (10)$$

尚、上段左辺の第 1 項の引数は 0 に固定されているので、式 (12) により逆変換する際には $\mathbf{x}$=$\mathbf{0}$ に固定する。ここで性質 (9) より $\varphi(t)$ は 4^n を因数に持つ。従って $f_y(0) + f_y(\tau)$ は $4^n/2^{m-1} = 2^{2n-m+1}$ を因数に持つ。これより、$m \leq 2n$ ならば頻度分布 $f_y(0) + f_y(\tau)$ は任意の τ において偶数となる。

(予想 2 の証明終わり)

A　アダマール変換と畳み込み演算の関係

定義域が $\mathrm{GF}(2)^m$ である任意の 2 つの実数関数を $p_i(\mathbf{x})$ $(i = 0, 1)$ とする $(\mathbf{x} \in \mathrm{GF}(2)^m, p_i(\mathbf{x}) \in R)$。さらに $p_i(\mathbf{x})$ をアダマール変換した関数を $P_i(\mathbf{t}) = \mathrm{H}_{2^m}\ p_i(\mathbf{x})$ とする $(\mathbf{t} \in \mathrm{GF}(2)^m, P_i(\mathbf{t}) \in R)$ と、アダマール変換 H_{2^m} は次式で定義される。

$$P_i(\mathbf{t}) = \sum_{\mathbf{x}} p_i(\mathbf{x})(-1)^{\mathbf{x}\cdot\mathbf{t}}. \quad (11)$$

また逆変換 $\mathrm{H}_{2^m}^{-1}$ は次式で定義される。

$$p_i(\mathbf{x}) = \frac{1}{2^m} \sum_{\mathbf{t}} P_i(\mathbf{t})(-1)^{\mathbf{x}\cdot\mathbf{t}}. \quad (12)$$

次に $p_0(\mathbf{x})$ と $p_1(\mathbf{x})$ の畳み込みを $q(\boldsymbol{\tau})$ とすると、$q(\boldsymbol{\tau})$ は次式で与えられる。

$$q(\boldsymbol{\tau}) = p_0(\mathbf{x}) * p_1(\mathbf{x}) = \sum_{\mathbf{x}} p_0(\mathbf{x}) p_1(\mathbf{x} \oplus \boldsymbol{\tau}). \quad (13)$$

さらに $q(\boldsymbol{\tau})$ のアダマール変換を $Q(\mathbf{t})$ とすれば、$Q(\mathbf{t})$ は次式で与えられる。

$$\begin{aligned} Q(\mathbf{t}) &= \sum_{\boldsymbol{\tau}} q(\boldsymbol{\tau})(-1)^{\boldsymbol{\tau}\cdot\mathbf{t}} \\ &= \sum_{\boldsymbol{\tau}}\sum_{\mathbf{x}} p_0(\mathbf{x})p_1(\mathbf{x}\oplus\boldsymbol{\tau})(-1)^{\boldsymbol{\tau}\cdot\mathbf{t}}. \end{aligned} \tag{14}$$

式 (14) において $\boldsymbol{\sigma} = \mathbf{x} \oplus \boldsymbol{\tau}$ として整理すると次式となる。

$$\begin{aligned} Q(\mathbf{t}) &= \sum_{\boldsymbol{\sigma}}\sum_{\mathbf{x}} p_0(\mathbf{x})p_1(\boldsymbol{\sigma})(-1)^{(\mathbf{x}\oplus\boldsymbol{\sigma})\cdot\mathbf{t}} \\ &= \sum_{\boldsymbol{\sigma}}\sum_{\mathbf{x}} p_0(\mathbf{x})p_1(\boldsymbol{\sigma})(-1)^{\mathbf{x}\cdot\mathbf{t}\oplus\boldsymbol{\sigma}\cdot\mathbf{t}}. \end{aligned} \tag{15}$$

$(-1)^0 = 1$ なので式 (15) において $(-1)^{\mathbf{x}\cdot\mathbf{t}\oplus\boldsymbol{\sigma}\cdot\mathbf{t}} = (-1)^{\mathbf{x}\cdot\mathbf{t}} \cdot (-1)^{\boldsymbol{\sigma}\cdot\mathbf{t}}$ と変形できる。これを適用すると式 (15) は次式となる。

$$\begin{aligned} Q(\mathbf{t}) &= \sum_{\boldsymbol{\sigma}}\sum_{\mathbf{x}} p_0(\mathbf{x})(-1)^{\mathbf{x}\cdot\mathbf{t}} \cdot p_1(\boldsymbol{\sigma})(-1)^{\boldsymbol{\sigma}\cdot\mathbf{t}} \\ &= \sum_{\mathbf{x}} p_0(\mathbf{x})(-1)^{\mathbf{x}\cdot\mathbf{t}} \cdot \sum_{\boldsymbol{\sigma}} p_1(\boldsymbol{\sigma})(-1)^{\boldsymbol{\sigma}\cdot\mathbf{t}} \\ &= P_0(\mathbf{t}) \cdot P_1(\mathbf{t}). \end{aligned} \tag{16}$$

故に式 (12) を用いて式 (16) を逆変換することにより、式 (13) に示された畳み込み $q(\boldsymbol{\tau})$ が得られる。$i = 0, 1, 2, \cdots$ と増えた場合も同様である。

B 性質 (6) の導出

性質 (6) 導出の理解を助けるために、ここではアダマール変換を行列として定義し、変換前後の関数はベクトルとする。アダマール行列 $\mathbf{H}$ とその逆行列 $\mathbf{H}^{-1}$ は次式で定義される。

$$\begin{aligned} &\mathbf{H}_1 = 1, \\ &\mathbf{H}_{2^m} = \begin{bmatrix} \mathbf{H}_{2^{m-1}} & \mathbf{H}_{2^{m-1}} \\ \mathbf{H}_{2^{m-1}} & -\mathbf{H}_{2^{m-1}} \end{bmatrix}, \\ &\mathbf{H}_{2^m}^{-1} = \frac{1}{2^m}\mathbf{H}_{2^m}, \quad (1 \le m \in N). \end{aligned} \tag{17}$$

ここで $\mathbf{H}$ の下付き添え字は行列の次数を表す。$\mathbf{H}_{2^m}$ の第 1 行目の要素は全て 1 である。その他の行は 1 の要素数が 2^{m-1} であり、-1 の要素数が 2^{m-1} である。また $\mathbf{H}_{2^m}$ の任意の行は符号長 2^m である拡大ハミング符号の検査行列 $\mathcal{H}_{2^m}$ の行の線形和となっている (但し、$\mathbf{H}_{2^m}$ の要素-1 は 0 で置き換える)。従って $\boldsymbol{\varphi} = \mathbf{H}_{2^m}\ \mathbf{f}_i$ とすると、ベクトル $\boldsymbol{\varphi}$ の第 0 要素 φ_0 は 2^m である ($\because \sum_{u=0}^{2^m-1} f_i(u) = 2^m$. $f_i(u)$ は $\mathbf{f}_i$ の要素)。第 1 要素 φ_1 は次式となる。

$$\varphi_1 = \left(\sum_{u=0}^{2^{m-1}-2} f_i(2u)\right) - \left(\sum_{u=0}^{2^{m-1}-2} f_i(2u+1)\right). \tag{18}$$

ここで右辺の第 1 項と第 2 項は共に拡大ハミング符号の符号語の要素の総和であるので式 (5) よりそれらの値は偶数となる。これより式 (18) は次式で書き直せる。

$$\begin{aligned} \varphi_1 &= \left(2^{m-1} + 2a\right) - \left(2^{m-1} - 2a\right) \\ &= 4a, \quad (a \in Z). \end{aligned} \tag{19}$$

従ってベクトル $\boldsymbol{\varphi}$ の第 1 要素 φ_1 は 4 を因数に持つ。第 2 要素以降についても式 (19) と同様なので、4 を因数に持つ。これより、性質 (6) が導かれる。

参考文献

[1] 角尾幸保, 辻原悦子, 久保博靖, 茂真紀, 川端剛嗣, “一般化 Feistel 構造の飽和特性,” IEICE 論文誌, vol. J93-A, no. 4, pp. 269–276, April, 2010