研究報告

共通鍵ブロック暗号の
線形攻撃耐性評価報告書

平成24年3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

2003 年に制定された CRYPTREC 推奨暗号リストに載っている 128 ビットブロック暗号 (AES, Camellia,CIPHERUNICORN-A,Hierocrypt-3,SC2000) について,192/256 ビット鍵の場合の線形攻撃に対する計算量的安全性に関し、線形特性確率の上界評価を行った。これら暗号は、2003 年の評価で、ブロック長で定まる 2^{-128} 以下の線形特性確率の上界が得られており、通常の線形攻撃に対する安全性が確認されているが、ここでは、関連鍵攻撃まで想定した安全性の第一次調査として、データ攪拌部全ラウンドに対し、S-box サイズの丸め線形パス探索を行い、その特性確率の上界を表 1 に示す。表 2 には、特性確率の上界が鍵長 $|K|$ に対し、$2^{-|K|}$ を下回るラウンド数 R を示す。　以下、各暗号の評価結果をまとめる。

AES,Camellia 及び Hierocrypt-3　データ攪拌部は、秘密鍵のビット数相当の安全性を有している。鍵スケジュール部が理想的であれば、関連鍵攻撃の条件において、線形攻撃に対する耐性があると考えられる。なお、Camellia の評価は、FL 関数無しで行った値である。表 2 において,AES の*印は、上界では無く、真の最大線形特性確率の評価である事を示す。

CIPHERUNICORN-A　データ攪拌部は、全ての鍵長において同じであり、線形特性確率は、鍵長に依存しない。ラウンド関数の構造が複雑であり、簡易な構造に変形し、評価した。従来と異なり、定数乗算及び A3 関数に関し、bit 単位の接続可能性に極力配慮した再評価を行った。しかし、拡大鍵入力の独立性を考慮した評価結果は、従来と同じである。192/256 ビット鍵の場合、この評価では $2^{-192}/2^{-256}$ より大きな上界の値である。

SC2000　S-box として、4,5,6 ビット幅のもの 3 種類が混在する事、及びビットスライス構造を持つ事の為丸め線形パス評価では、大幅に緩い上界しか得られない。表中の () 内の値は、[8] の繰り返しパスを全ラウンドに、適用した値であり、存在する線形パスの確率である。256 ビット鍵の場合、2^{-256} より大きいパスとなっている。

暗号 \ 鍵長	線形特性確率の上界		
	128bits	192bits	256bits
AES	2^{-330}	2^{-450}	2^{-480}
Camellia	2^{-228}	2^{-324}	同左
CIPHERUNICORN-A	2^{-171}	同左	同左
Hierocrypt-3	2^{-450}	2^{-480}	2^{-600}
SC2000	(2^{-176})	(2^{-204})	同左

表 1: 線形特性確率の上界

暗号＼鍵長	R/全ラウンド数		
	128bits	192bits	256bits
AES	4*/10	7/12	8/14
Camellia	11/18	15/24	21/24
CIPHERUNICORN-A	12/16	–	–
Hierocrypt-3	2/6	4/7	4/8
SC2000	(15/19)	(21/22)	–

表 2: 攻撃計算量が鍵全数探索を上回るラウンド数 R

目 次

第1章　はじめに

電子政府推奨暗号リスト（以下 CRYPTREC リスト [1]) は、各府省が情報システムの構築に当たり利用が推奨される暗号リストとして、CRYPTREC により 2003 年に策定された。それ以来 10 年が経とうとし、計算機環境や、暗号の解析・攻撃技術の進展及び、新たな暗号技術の開発がなされており、CRYPTREC に於いては、2013 年にリスト改訂を予定している。

ここでは、2003 年に制定された CRYPTREC 推奨暗号リストに載っている 128 ビットブロック暗号 (AES[3],Camellia[4],CIPHERUNICORN-A[5], Hierocrypt-3[6],SC2000[7]) について,192/256 ビット鍵の場合の線形攻撃に対する計算量的安全性に関し、改めて線形特性確率の上界評価を行う。これら暗号は、2003 年の評価で、ブロック長で定まる 2^{-128} 以下の線形特性確率の上界が得られており、通常の線形攻撃に対する安全性が確認されている [2]。

現時点においては、実際的な脅威では無いが、AES が関連鍵攻撃で理論的解読可能 [9] との報告もある。ここでは、関連鍵攻撃まで想定した安全性の第一次調査として、データ攪拌部全ラウンドに対し、S-box サイズの丸め線形パス探索を行い、その特性確率の上界を求める。

以下に本報告書の構成について述べる。第 2 章では、丸め線形確率評価法をまとめ第 3 章~ 7 章で、それぞれの暗号に固有の評価過程をまとめる。

第2章　線形攻撃

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。

2.1　線形確率

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。

$$LP(\Gamma_x, \Gamma_y) = (2 \cdot \frac{\#\{x \in \{0,1\}^n \mid x \bullet \Gamma_x = y \bullet \Gamma_y\}}{2^n} - 1)^2 \tag{2.1}$$

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。

$$LP_{max} = \max_{\Gamma x, \Gamma y \neq 0} LP(\Gamma x, \Gamma y) \tag{2.2}$$

2.2　最大線形特性確率

暗号化関数に対しても、式 (2.2) 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。

$$LCP_{max} = \max_{\substack{\Gamma_{x_0}, \Gamma_{x_1}, \cdots, \Gamma_{x_R} \\ \Gamma_{x_i} \neq 0}} \prod_{i=0}^{R-1} LP(\Gamma_{x_i}, \Gamma_{x_{i+1}}) \tag{2.3}$$

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。

2.3　丸め線形解析

実際に最大線形特性確率を求めることも計算量的に困難である場合、丸め線形解析を用いて、最大線形特性確率の上界である最大丸め線形特性確率 LCP_{Tmax} を求める。丸め線形解析とは、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。S-box 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。

$$\Gamma = (\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3), \quad \Gamma_i \in \mathrm{GF}(2). \tag{2.4}$$

マスクの伝播は分岐においては図 2.1 のいずれかで表され、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。データが分岐する箇所においてマスクは、XOR 和となる。ここで丸めマスク Γ の XOR 演算 ($\Gamma' \bar{\oplus} \Gamma''$) の規則は表 2.1 で定められる。排他的論理和においては図 2.2 で表される。データが排他的論理和される場所では、マスクは同じ値が分岐する。

また S-box においては、入出力マスクが active であれば、その S-box を active S-box と呼ぶ。以降、ラウンド r の処理で生じる active S-box の数を $as^{(r)}$ と書き、その合計を $AS(=\Sigma as^{(r)})$ と表す。

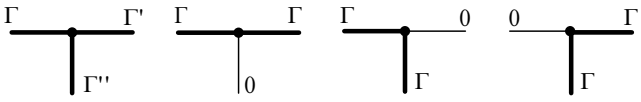

図 2.1: 分岐における丸めマスクの伝播

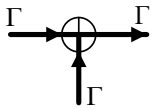

図 2.2: 排他的論理和における丸めマスクの伝播

表 2.1: 丸めマスクの XOR 演算規則

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

2.4 丸め線形パス探索

暗号関数の丸め線形解析においては、S-box 等の非線形関数に関し、線形確率を求め、その最大確率 lp_{max} で、丸め線形パスの入出力マスクが接続できると考え評価を行う。S-box のビット幅単位の丸めを行うとし、丸め線形パスにおいて、active な S-box 数を数え上げ、その数を AS とする。最大丸め線形特性確率 LCP_{Tmax} は、AS の最小値を AS_{min} とするならば

$$LCP_{Tmax} = lp_{max}^{AS_{min}} \tag{2.5}$$

で与えられる。従って、丸め線形マスクの伝搬状況 $\Gamma_{Tx_0} \rightarrow \Gamma_{Tx_1} \rightarrow \Gamma_{Tx_2} \rightarrow \cdots \rightarrow \Gamma_{Tx_R}$ に対し、伝播のコストを AS として、それを最小化するパスを探索する事が LCP_{Tmax} を求める事になる。これは、最短経路探索問題であり、Viterbi 探索で求められる。

この様な、探索で得られる、最大丸め線形特性確率 LCP_{Tmax} は、最大線形特性確率 LCP_{max} の上界を与える物であり、必ずしもその確率を持つ線形パスの存在を保証するものでは無い。緩い上界となる原因の一つが、丸めマスクの XOR 演算規則の冗長性である。例えば、

$$\begin{aligned} z &= x \oplus y \\ w &= x \oplus y \end{aligned} \tag{2.6}$$

の線形関係となっていて、$x = 1$,$y = 1$ の時、丸めマスクの XOR 和で評価すると、$(z, w) = (0,0), (0,1), (1,0)\ or (1,1)$ の 4 通りの値となるが、明らかに $(x, w) = (0,0)\ or (1,1)$ の二通りの値しか取らない。より tight な上界を与える為には、線形関数も、可能であれば一つの構成部品として捉え、丸めパスの接続の可否を表として用意し、それを利用した探索を行う必要がある。

第3章　AES

3.1　構成部品の特性

AES のデータランダム化部の構成部品は、非線形関数として、S-box、線形関数として MixColumuns、Shiftrows、AddRoundKey である S-box は 8 ビット入出力であり、8 ビット丸め線形解析を行う。

3.1.1　S-box

S-box の最大線形確率は、$LP_{max} = 2^{-6}$ であり、LP_{max} を与える入出力マスク (Γ_x, Γ_y) は 1275 組存在する。非零の入力マスクに対し、$LP_{max} = 2^{-6}$ を与える出力マスクは、5 通りづつ存在する。その一部を表 3.1 に示す。

表 3.1: Sbox 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

(0x1,0x48)	(0x1,0x50)	(0x1,0x88)	(0x1,0x90)	(0x1,0xc0)
(0x2,0x38)	(0x2,0x40)	(0x2,0x78)	(0x2,0x8f)	(0x2,0xcf)
(0x3,0x1a)	(0x3,0x24)	(0x3,0x44)	(0x3,0x60)	(0x3,0x7a)
(0x4,0x1c)	(0x4,0x20)	(0x4,0x3c)	(0x4,0x95)	(0x4,0xb5)
(0x5,0xf)	(0x5,0x67)	(0x5,0x68)	(0x5,0xb6)	(0x5,0xb9)
(0x6,0x19)	(0x6,0x74)	(0x6,0xa3)	(0x6,0xba)	(0x6,0xd7)
(0x7,0xd)	(0x7,0x12)	(0x7,0x62)	(0x7,0x6f)	(0x7,0x70)
(0x8,0xf)	(0x8,0x35)	(0x8,0x52)	(0x8,0x5d)	(0x8,0x67)
(0x9,0xe)	(0x9,0x42)	(0x9,0x4c)	(0x9,0x98)	(0x9,0xda)
(0xa,0x5b)	(0xa,0x66)	(0xa,0x8e)	(0xa,0xb3)	(0xa,0xd5)
(0xb,0x4d)	(0xb,0x6b)	(0xb,0x9e)	(0xb,0xd3)	(0xb,0xf5)
(0xc,0x38)	(0xc,0x8f)	(0xc,0xb7)	(0xc,0xc6)	(0xc,0xfe)
(0xf,0xbb)	(0xf,0xf1)	(0xf,0xfa)		

3.1.2　MixColumns

MixColumns で用いられる行列は、最大分岐特性を持ち、その分岐数は 5 である。また、その逆行列である invMixColumns の行列もまた分岐数は 5 である。従って、入出力マスクが非零の場合、最低でも 5 バイトの非零マスクが存在する。

3.1.3 Shiftrows,AddRoundKey

Shiftrowsは、バイト単位の転置であり、8ビット丸め線形マスクも同じ転置を受ける。AddRoundKeyは、ラウンド鍵の排他的論理和であり、線形解析においては、その影響を考える必要は無い。

3.2 AESの丸め線形解析

3.2.1 解析方法

最小のActive S-box数を与える、丸め線形パスは、Viterbiアルゴリズムで探索できる。各段の丸め線形マスクを状態と考え、Shiftrowsの効果を状態遷移を表すトレリス線図として表現し、MixColumunsが最小分岐数5の行列である事を用いて、次段でactiveとなる最小S-box数を見積もれば良い。Viterbiアルゴリズムの遷移に当たっての増加コストが、このMixcolumnsで決定する最小S-box数である。

Viterbiアルゴリズムで用いる状態及び状態遷移を図3.1を用いて説明する。各ラウンド関数の8bit-truncate入出力マスクを$\Gamma x_{(i,t)}$,$\Gamma x_{(i,t+1)}$と表す。$\Gamma x_{(i,t)}$,$\Gamma x_{(i,t+1)}$は1ビットである。
状態遷移の始めは1段目の状態変数である次式

$$\begin{aligned} st(0) = (&\Gamma x_{(0,0)}, \Gamma x_{(1,0)}, \Gamma x_{(2,0)}, \Gamma x_{(3,0)}, \Gamma x_{(4,0)}, \Gamma x_{(5,0)}, \Gamma x_{(6,0)}, \Gamma x_{(7,0)}, \Gamma x_{(8,0)}, \\ &\Gamma x_{(9,0)}, \Gamma x_{(10,0)}, \Gamma x_{(11,0)}, \Gamma x_{(12,0)}, \Gamma x_{(13,0)}, \Gamma x_{(14,0)}, \Gamma x_{(15,0)}) \end{aligned} \tag{3.1}$$

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

$$\begin{aligned} st(1) = (&\Gamma x_{(0,1)}, \Gamma x_{(1,1)}, \Gamma x_{(2,1)}, \Gamma x_{(3,1)}, \Gamma x_{(4,1)}, \Gamma x_{(5,1)}, \Gamma x_{(6,1)}, \Gamma x_{(7,1)}, \Gamma x_{(8,1)}, \\ &\Gamma x_{(9,1)}, \Gamma x_{(10,1)}, \Gamma x_{(11,1)}, \Gamma x_{(12,1)}, \Gamma x_{(13,1)}, \Gamma x_{(14,1)}, \Gamma x_{(15,1)}) \end{aligned} \tag{3.2}$$

とする。$st(0)$から$st(1)$への可能な遷移はMixColumnsの最小分岐数に則って決定される。$t \geq 0$では次式の状態変数をとる。

$$\begin{aligned} st(t) = (&\Gamma x_{(0,t)}, \Gamma x_{(1,t)}, \Gamma x_{(2,t)}, \Gamma x_{(3,t)}, \Gamma x_{(4,t)}, \Gamma x_{(5,t)}, \Gamma x_{(6,t)}, \Gamma x_{(7,t)}, \Gamma x_{(8,t)} \\ &, \Gamma x_{(9,t)}, \Gamma x_{(10,t)}, \Gamma x_{(11,t)}, \Gamma x_{(12,t)}, \Gamma x_{(13,t)}, \Gamma x_{(14,t)}, \Gamma x_{(15,t)}) \end{aligned} \tag{3.3}$$

3.3 解析結果

Viterbiアルゴリズムの探索結果として、10段構成の128bit AESでは$LCP_{Tmax}^{10round} = 2^{-330}$、12段構成の192bitAESでは$LCP_{Tmax}^{12round} = 2^{-450}$、同じく14段構成の256bitAESでは$LCP_{Tmax}^{14round} = 2^{-480}$が得られた。秘密鍵ビット数との関係として、$2^{-128}, 2^{-192}, 2^{-256}$を初めて下回るラウンド数は、それぞれ4,7,8ラウンドであり、仕様の段数より小さい。表3.2にAS数及び最大丸め線形特

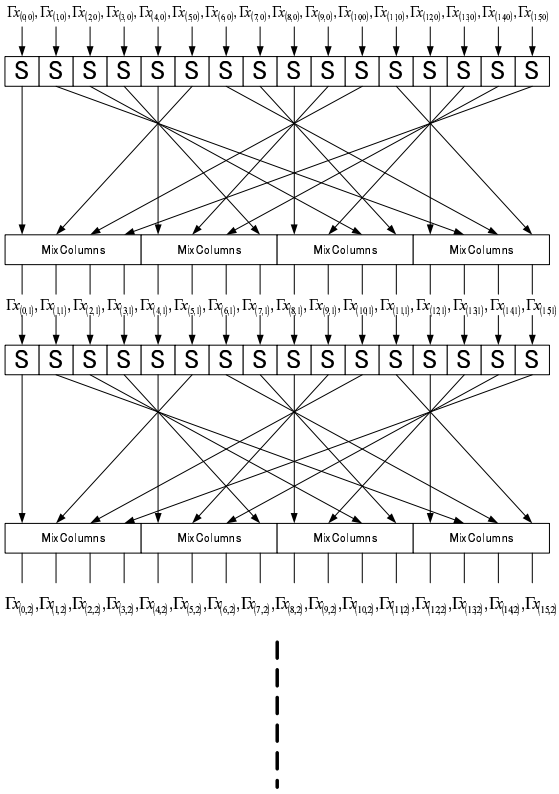

図 3.1: Viterbi 探索における状態変数の取り方

性確率の段数との関係を示す。確率は $log_2(LCP_{Tmax})$ の値で表している。表 3.3,3.4,3.5 に本解析結果を与える丸め線形パスの一例を示す。これは丸め評価による線形確率の上界値であり、必ずしも、真の最大線形特性確率ではない。しかし、最大丸め線形パスが、真の最大線形パスで有るか否かを、確認したところ、4 段までは真の最大線形パスとなっている事を確認した。即ち、表 3.1 の最大線形確率を与えるマスクのみを利用して、4 段まで繋がる線形パスが存在する。5 段目以降は、その最大丸め線形パスでは、真の線形パスは繋がらない。従って、5 段以上について、LCP_{Tmax} は、真の最大線形特性確率ではなくその上界である。

表 3.2: 段数と active Sbox 数及び LCP_{Tmax},LCP_{max} の関係

段数	AS 数	$LCP_{Tmax}[log_2]$	$LCP_{max}[log_2]$
1	1	-6	-6
2	5	-30	-30
3	9	-54	-54
4	25	-150	-150
5	26	-156	—
6	30	-180	—
7	34	-204	—
8	50	-300	—
9	51	-306	—
10	55	-330	—
11	59	-354	—
12	75	-450	—
13	76	-456	—
14	80	-480	—

表 3.3: 10 ラウンドにおける本稿の結果を与える丸めマスクの一例

t	ラウンド関数の丸め入力マスク st(t)
0	st(0)=(0010000110000100)
1	st(1)=(0000000000010000)
2	st(2)=(0000000000001111)
3	st(3)=(1111111111111111)
4	st(4)=(0010000110000100)
5	st(5)=(0000000000010000)
6	st(6)=(0000000000001111)
7	st(7)=(1111111111111111)
8	st(8)=(0010000110000100)
9	st(9)=(0000000000010000)

表 3.4: 12 ラウンドにおける本稿の結果を与える丸めマスクの一例

t	ラウンド関数の丸め入力マスク st(t)
0	st(0)=(0000000000010000)
1	st(1)=(0000000000001111)
2	st(2)=(1111111111111111)
3	st(3)=(0010000110000100)
4	st(4)=(0000000000010000)
5	st(5)=(0000000000001111)
6	st(6)=(1111111111111111)
7	st(7)=(0010000110000100)
8	st(8)=(0000000000010000)
9	st(9)=(0000000000001111)
10	st(10)=(1111111111111111)
11	st(11)=(0010000110000100)

表 3.5: 14 ラウンドにおける本稿の結果を与える丸めマスクの一例

t	ラウンド関数の丸め入力マスク st(t)
0	st(0)=(0010000110000100)
1	st(1)=(0000000000010000)
2	st(2)=(0000000000001111)
3	st(3)=(1111111111111111)
4	st(4)=(0010000110000100)
5	st(5)=(0000000000010000)
6	st(6)=(0000000000001111)
7	st(7)=(1111111111111111)
8	st(8)=(0010000110000100)
9	st(9)=(0000000000010000)
10	st(10)=(0000000000001111)
11	st(11)=(1111111111111111)
12	st(12)=(0010000110000100)
13	st(13)=(0000000000010000)

3.4 まとめ

ここでは、AES の 8bit 丸め線形解析で行い、Viterbi アルゴリズムを利用して丸め線形パスを探索し、最大線形特性確率の上界を求めた。その結果、鍵長に対応した $2^{-128}, 2^{-192}, 2^{-256}$ を初めて下回るラウンド数は、それぞれ 4,7,8 ラウンドであり、仕様のラウンド数より小さい。よって AES は、鍵スケジュール部が理想であれば、線形攻撃に対して十分な耐性を持つ。

第4章　Camellia

4.1　構成部品の特性

Camellia のデータランダム化部の構成部品の、非線形関数は S_1,S_2,S_3,S_4 の 4 種類の S-box である。ラウンド関数は SP 構造であり、この S-box の後ろに行列 P が位置する。Camellia は、このラウンド関数を用いた Feistel 構造を持つ。これ以外に 6 段毎に鍵依存の線形関数 FL 及び FL^{-1} 関数が挿入されるがこれらは線形関数であり、ここでは、FL 及び FL^{-1} 関数を取り除いた Camellia を評価した。丸め線形解析においては、線形関数の丸め線形特性の取り扱いによっては、本来の最大線形特性確率に比べ、大幅に緩い上界を与える可能性があるからである。S-box のビット幅は 8-bit であり、ここでは 8 ビット丸め線形解析を行う。

4.1.1　S-box

Camellia の S-box の最大線形確率を計算したところ、S_1,S_2,S_3,S_4 のいずれにおいても $LP_{max} = 2^{-6}$ である。

4.1.2　P 関数

Camellia のラウンド関数に於いては、S-box 層通過後の 8 バイトデータ $\boldsymbol{X} = (\boldsymbol{x_8}, \boldsymbol{x_7}, \cdots, \boldsymbol{x_1})$、$(x_i \in GF(2^8))$ が、[1]行列 $\mathbf{P}$ で線形変換され、ラウンド関数の出力の 8 バイトデータ $\boldsymbol{Y} = (\boldsymbol{y_8}, \boldsymbol{y_7}, \cdots, \boldsymbol{y_1})$、$(y_i \in GF(2^8))$ は、

$$\boldsymbol{Y^T} = \mathbf{P}\ \boldsymbol{X^T} \tag{4.1}$$

となる。ここで ^{T} は転置を表す。行列 $\mathbf{P}$ は、次式である。

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{4.2}$$

入出力マスクを、それぞれ $\boldsymbol{\Gamma_X}$、$\boldsymbol{\Gamma_Y}$ とすれば、接続可能な入出力マスクは

$$\boldsymbol{\Gamma_Y}\mathbf{P} = \boldsymbol{\Gamma_X} \tag{4.3}$$

[1]Camellia の仕様書では、横ベクトルは、第 1 バイトを先頭に配置し、縦ベクトルは、第 1 バイトを最下部に表記してある。ここの表記では、式 (4.1) が正しい表現となるように、通常の表記 $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_8})$ とは異なるバイト順とした。

を満たす物である。この入出力マスクを、丸めたものを $\bar{\boldsymbol{\Gamma}}_{\boldsymbol{X}} = Tr(\boldsymbol{\Gamma}_{\boldsymbol{X}})$ 等と、上に¯を付けて表すと、丸め線形パス $\bar{\boldsymbol{\Gamma}}_{\boldsymbol{X}} \rightarrow \bar{\boldsymbol{\Gamma}}_{\boldsymbol{Y}}$ が実際の線形パス $\boldsymbol{\Gamma}_{\boldsymbol{X}} \rightarrow \boldsymbol{\Gamma}_{\boldsymbol{Y}}$ として接続可能か否かは次節の手法で判断が出来る。Camellia の P 関数は、発見的手法で求められたものであり、単に、丸めマスクの XOR 演算規則や、最小分岐数の考えのみを用いたのでは、最大線形特性確率の緩い上界となってしまう。本報告書の最良丸め線形パス探索に於いては、全ての丸め線形パス $\bar{\boldsymbol{\Gamma}}_{\boldsymbol{X}} \rightarrow \bar{\boldsymbol{\Gamma}}_{\boldsymbol{Y}}$ に対し、次節の方法で、接続の可否を調べておき、それをテーブルとして持つ事により最良丸めパスの探索を行っている。

4.2 線形変換部の丸め差分解析を行う際の虚パスの排除法

丸め線形マスクの XOR 演算規則では繋がっているが、真の線形パスとしては、繋がり得ない丸め線形パスをここでは虚パスと呼ぶ。丸め線形マスクの接続の問題は、線形パスと差分パスの双対性の関係より出力マスクを入力差分に、入力マスクを出力差分に置き換えて、パスの接続を考えれば良い。ここでは直感的に理解しやすい差分パスの接続問題として説明する。

図 4.1 の線形変換部を考える。

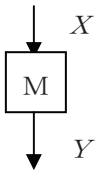

図 4.1: 線形変換部

M バイト入力 $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_M})$ が行列 $\mathbf{M}$ で変換され、N バイト出力 $\boldsymbol{Y} = (\boldsymbol{y_1}, \boldsymbol{y_2}, \cdots, \boldsymbol{y_N})$ となる。

$$\boldsymbol{Y^T} = \mathbf{M}\boldsymbol{X^T} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1M} \\ m_{21} & m_{22} & \cdots & m_{2M} \\ \vdots & \vdots & \cdots & \vdots \\ m_{N1} & p_{N2} & \cdots & m_{NM} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \tag{4.4}$$

ここで $x_i, y_j, m_{ij} \in \mathrm{GF}(2^l)$ であり、l はバイトサイズである。差分 $\Delta\boldsymbol{X}$、$\Delta\boldsymbol{Y}$ に対し、$\Delta\boldsymbol{Y^T} = \mathbf{M}\Delta\boldsymbol{X^T}$ である。以下煩雑さを避ける為、$\Delta\boldsymbol{X}$、$\Delta\boldsymbol{Y}$ を単に $\boldsymbol{X}$、$\boldsymbol{Y}$ と表記する[2]。

入力丸め差分は

$$\bar{\boldsymbol{X}} = Tr(\boldsymbol{X}) = (\bar{x_1}, \bar{x_2}, \cdots, \bar{x_N}) \text{ 但し、} \bar{x_i} = \begin{cases} 0 & (x_i = 0) \\ 1 & (x_i \neq 0) \end{cases} \tag{4.5}$$

[2]線形パスの議論においては、入出力マスクを Γ_x、Γ_y としたとき、線形変換部で、接続可能なマスクは $\Gamma_x^T = \mathbf{M^T}\Gamma_y^T$ となる。従って、この議論を、入力 $\boldsymbol{X}$ ⇒ 出力マスクΓ_y、出力 $\boldsymbol{Y}$ ⇒ 入力マスクΓ_x、行列 $\mathbf{M}$ ⇒ 行列 $\mathbf{M^T}$ と読み替えて適用すればよい。

とする。出力丸め差分 $\bar{\boldsymbol{Y}}$ も、同様な表記とする。丸め差分 $\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}$ が与えられた時、パス $\bar{\boldsymbol{X}} \rightarrow \bar{\boldsymbol{Y}}$ がビット単位に接続可能かどうかは以下の方法で調べることができる。

ここで、幾つかのベクトル、行列を定義する。

- $\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}$ において、1 となっている箇所数を、それぞれ m, n とする。0 となっている箇所数は、それぞれ $M-m$、$N-n$ である。
- $\boldsymbol{X}$ から、$\bar{x_i}=1$ に対応する要素のみを選んだ m 次元ベクトルを $\boldsymbol{X_1}$ とする。
- Y から、$\bar{y_j}=0$ に対応する要素のみを選んだ $N-n$ 次元ベクトルを $\boldsymbol{Y_0}$、$\bar{y_j}=1$ に対応する要素のみを選んだ n 次元ベクトルを $\boldsymbol{Y_1}$、それを連結したベクトルを $\boldsymbol{Y'=(Y_0||Y_1)}$ とする。
- $\bar{x_i}=0$ に対応する列を、行列 $\mathbf{M}$ から削除し、$\boldsymbol{Y}$ から $\boldsymbol{Y'}$ を作成する時の、要素入れ替えに対応した行入れ替えを行った $N \times m$ 行列を次式とする。

$$\mathbf{M}' = \begin{bmatrix} \mathbf{M_0} \\ \mathbf{M_1} \end{bmatrix} \tag{4.6}$$

このとき、式 (4.4 は、$(N-n) \times m$ 行列 $\mathbf{M_0}$、$n \times m$ 行列 $\mathbf{M_1}$ を使い、次の二式となる。($\boldsymbol{Y_0}$ に注意。)

$$0 = \mathbf{M_0} \boldsymbol{X_1^T} \tag{4.7}$$

$$\boldsymbol{Y_1^T} = \mathbf{M_1} \boldsymbol{X_1^T} \tag{4.8}$$

$\boldsymbol{X_1}$ の各要素が、$\boldsymbol{X}$ の $x_i \neq 0$ 成分を表し、$\boldsymbol{Y_1}$ の各要素が $\boldsymbol{Y}$ の $y_j \neq 0$ 成分を表していることに注意すると、パス $\bar{\boldsymbol{X}} \rightarrow \bar{\boldsymbol{Y}}$ がビット単位に接続する事は、各要素が非零の $GF(2^l)$ 上のあるベクトル $\boldsymbol{X_1}, \boldsymbol{Y_1}$ に対し、式 (4.7)(4.8) が成立することに等しい。

まず、式 (4.7) に着目し、行列 $\mathbf{M_0}$ を基本操作により既約台形正準型に変形し、その後、適切に列入替えを行うならば、等価な式として、

$$\mathbf{M_0} \boldsymbol{X_1^T} = 0 \qquad \Rightarrow \qquad \begin{bmatrix} \mathbf{I_k} & \mathbf{P_0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \boldsymbol{X_{11}^T} \\ \boldsymbol{X_{12}^T} \end{pmatrix} = 0 \tag{4.9}$$

ここで、$rank(\mathbf{M_0}) = k$、最後の列入替 (既約台形正準型の行列の左半分に単位行列を集める操作) に対応して、ベクトル $\boldsymbol{X_1}$ の要素を並び替えたベクトルが、$\boldsymbol{(X_{11}||X_{12})}$ である。($\mathbf{P_0}$) は、$k \times (m-k)$ 行列、$\mathbf{I_k}$ は k 次単位行列、$\boldsymbol{X_{11}, X_{12}}$ は、それぞれ $k, (m-k)$ 次元ベクトルである。従って、式 (4.7) を満たすベクトル $\boldsymbol{X_1}$ は、パリティ検査行列 $\mathbf{H} = (\mathbf{I_k}||\mathbf{P_0})$ を持つ $(m, m-k)$ 符号の符号語であり情報点が $\boldsymbol{X_{12}}$、検査点が $\boldsymbol{X_{11}}$ である。

検査条件 1($\bar{x_i}=1$ の検査) : 行列 $\mathbf{P}_0$ の中に全零の $GF(2^l)$ の行がある場合、それに対応する検査点 $\boldsymbol{X_{11}}$ の要素が $x_i = 0$ である事になり、非零の $GF(2^l)$ のベクトル $\boldsymbol{X_1}$ に対し、式 (4.7) が成立しないことを意味する。従って、この様な場合、パス $\bar{\boldsymbol{X}} \rightarrow \bar{\boldsymbol{Y}}$ がビット単位に接続することはない。

次に、式 (4.8) を含めて議論する。式 (4.7) の条件は、式 (4.9) の条件と等価であるので、式 (4.9) における $\boldsymbol{X_1}$ の要素の並び替えに対して、$\mathbf{M_1}$ の列を入れ替えた行列を、$\mathbf{M_1'}$ とするならば、式 (4.8), (4.9) の連立方程式は以下となる。

$$\begin{bmatrix} \mathbf{I}_k & \mathbf{P}_0 \\ \mathbf{M_1'} \end{bmatrix} \begin{pmatrix} \boldsymbol{X_{11}^T} \\ \boldsymbol{X_{12}^T} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{Y_1^T} \end{pmatrix} \tag{4.10}$$

なお、ここで、式 (4.9) の下半分の $N-n-k$ 行分の式は $0=0$ の恒等式であり式 (4.10) からは除外してある。式 (4.7) を満足する $\boldsymbol{X_1}$ の中で式 (4.8) を満足する物を探す事に注意するならば、式

(4.10) の左辺の行列において、上半分の k 行を下半分に加算する基本行操作を行なっても連立方程式は等価に保たれる。右辺側の上半分は 0 であり、それを下半分の行に加算しても、右辺の値は変化しないからである。そのような、基本行操作を行えば、式 (4.10) の下半分の先頭 k 列は 0 と出来る。すなわち、等価な式として、

$$\begin{bmatrix} \mathbf{I_k} & \mathbf{P_0} \\ \mathbf{0} & \mathbf{P_1} \end{bmatrix} \begin{pmatrix} \boldsymbol{X}_{\boldsymbol{11}}^{\boldsymbol{T}} \\ \boldsymbol{X}_{\boldsymbol{12}}^{\boldsymbol{T}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{Y}_{\boldsymbol{1}}^{\boldsymbol{T}} \end{pmatrix} \tag{4.11}$$

が得られる。

検査条件 2($\bar{y_j} = 1$ の検査) ：行列 $\mathbf{P_1}$ の中に全零の行がある場合、それに対応する $\boldsymbol{Y_1}$ の要素が $y_j = \mathbf{0}$ である事を意味し、各要素が非零の $GF(2^l)$ のベクトル $\boldsymbol{X_l}$ に対し、式 (4.8) が成立しない事を意味する。従って、この様な場合、パス $\bar{\boldsymbol{X}} \to \bar{\boldsymbol{Y}}$ がビット単位に接続する事はない。

この二つの検査条件を通過した丸め差分パス $\bar{\boldsymbol{X}} \to \bar{\boldsymbol{Y}}$ であれば、$N < 2^l$ の条件下において、ビット単位に接続する実際のパス $\boldsymbol{X} \to \boldsymbol{Y}$ が存在する事が、次節の考察で言える。

4.3 検査条件 1、2 を通過した丸めパスが実パスを持つ理由

式 (4.11) を変形し、次式

$$\begin{bmatrix} \mathbf{P_0} \\ \mathbf{P_1} \end{bmatrix} \boldsymbol{X}_{\boldsymbol{12}}^{\boldsymbol{T}} = \begin{pmatrix} \boldsymbol{X}_{\boldsymbol{11}}^{\boldsymbol{T}} \\ \boldsymbol{Y}_{\boldsymbol{1}}^{\boldsymbol{T}} \end{pmatrix} \tag{4.12}$$

が得られる。この式の解として、各要素が非零のベクトル $\boldsymbol{X_{12}}, \boldsymbol{X_{11}}, \boldsymbol{Y_1}$ が存在することが、丸めパス $\bar{\boldsymbol{X}} \to \bar{\boldsymbol{Y}}$ がビット単位に接続する条件である。行列のサイズを確認し、問題を見やすく整理する。$\mathbf{P_0}$ は $k \times (m-k)$ 行列、$\mathbf{P_1}$ は $n \times (m-k)$ 行列である。$(n-k)$ 次元ベクトル $\boldsymbol{X_{12}} = \boldsymbol{Z} = \begin{pmatrix} z_1 & z_2 & \dots & z_{n-k} \end{pmatrix}$、$(n+k)$ 次元ベクトル $(\boldsymbol{X_{12}} \parallel \boldsymbol{Y_1}) = \boldsymbol{W} = \begin{pmatrix} w_1 & w_2 & \dots & w_{n+k} \end{pmatrix}$ と置く。確認すべき問題は、次式が全て非零の z_i、w_j に対して解を持つ事である。

$$\mathbf{Q}\boldsymbol{Z} = \boldsymbol{W} \tag{4.13}$$

但し、行列 $\mathbf{Q}$ は次式である。

$$\mathbf{Q} = [q_{ij}] = \begin{bmatrix} \mathbf{P_0} \\ \mathbf{P_1} \end{bmatrix} \tag{4.14}$$

ここで、ベクトルの要素 w_j の非零性が問題であり、これは a を非零とし、a w_j の非零性を検査しても良い。また、z_i の非零性に関しては、その並び順は問題では無い。従って、式 (4.14) の行列 $\mathbf{Q}$ に対し、行の非零の定数倍及び、列の入れ替えを行っても確認すべき問題は変化しない。よって、一般性を失うこと無く $(n+k) \times (m-k)$ 行列 $\mathbf{Q}$ は

$$\mathbf{Q} = \begin{bmatrix} 1 & * & * & * & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & * & * & * & \dots & * \\ 0 & 1 & * & * & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 1 & * & * & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 & \dots & * \end{bmatrix} \tag{4.15}$$

の形に変形できる。第1列に1を持つ行の数を$l(1)$、非零の要素が第2列から始まる行の数を$l(2)$、$\cdots$とする。$\mathbf{P_0}, \mathbf{P_1}$には、全零行が無いので、各行の非零の先頭要素は、1となっている。この$\mathbf{Q}$に対し、式(4.14)の先頭の$l(1)$行を、要素に展開すれば

$$\begin{array}{ccccccccc} z_1 & + & q_{12}z_2 & + \ldots & + & q_{1(m-k)}z_{m-k} & = & w_1 \\ z_1 & + & \ldots & + \ldots & + & \ldots & = & \ldots \\ z_1 & + & q_{l(1)2}z_2 & + \ldots & + & q_{l(1)(m-k)}z_{m-k} & = & w_{l(1)} \end{array} \tag{4.16}$$

である。この式において、$w_1, \ldots, w_{l(1)}$に要求される性質は、非零性である。その為には、非零の$z_2, \ldots, z_{m-k}$を任意に選び、次の連立不等式が満足される様に、非零のz_1を定めれば良い。

$$\begin{array}{ccccccc} -z_1 & \neq & q_{12}z_2 & + \ldots & + & q_{1(m-k)}z_{m-k} \\ -z_1 & \neq & \ldots & + \ldots & + & \ldots \\ -z_1 & \neq & q_{l(1)2}z_2 & + \ldots & + & q_{l(1)(m-k)}z_{m-k} \end{array} \tag{4.17}$$

この連立不等式の右辺の値は、高々$l(1)$種類である。$GF(2^l)$の非零元は$2^l - 1$種類ある。従って、$2^l > l(1)$であれば、この連立不等式を満足するz_1を選ぶ事が出来る[3]。行列$\mathbf{Q}$の$l(1)+1$行目から$l(1)+l(2)$行目までの式を非零のz_2を選ぶ連立不等式と考えれば、非零の$z_3, \ldots, z_{m-k}$を任意に選ぶ事により[4]、非零の$w_{l(1)+1}, \ldots, w_{l(1)+l(2)}$が定まる。以下同様の繰り返しである。連立不等式に要求される条件が一番厳しい$l(1) = n + k$を想定しても、$2^l > N$であれば、$2^l > N \geq n + k$であり、式(4.13)は、全ての要素z_i, w_jが非零の解を持つ事ができる。即ち、丸めパス$\bar{\boldsymbol{X}} \rightarrow \bar{\boldsymbol{Y}}$は、ビット単位に接続する実パス$\boldsymbol{X} \rightarrow \boldsymbol{Y}$を持つ。

4.4 解析結果

ここでは128bit Camelliaと192/256bit Camelliaのデータ攪拌部について、8ビット丸め線形パスを用いて、最大丸め線形特性確率$LCP_{T_{max}}$を評価する。これは、データ攪拌部においてactive S-boxの総数が最小となる丸め線形パスを導出することである。最良パス探索は、Viterbiアルゴリズムで行った。探索アルゴリズム実行前の準備として、ラウンド関数における$\mathbf{P}$行列において、$0x00$から$0xFF$までの256種類の丸め入力マスクΓ_Xに対し、ビット単位に接続可能な丸め出力マスクΓ_Yを第4.2節の方法で求めてテーブル作成し、Viterbiアルゴリズムにおけるマスク遷移可能条件として準備した。Viterbiアルゴリズムにおける線形マスク探索においては、ラウンド関数に於いて、出力マスクを選び、最大線形確率で接続される入力マスクを選ぶ作業が行われる関係上、丸めパスのXOR演算を避ける為に、暗号系を等価変形し、ラウンド関数の構造をSP型からPS型に変形して探索を行っている。

最良丸めパスの導出結果を図4.2及び図4.3に示す。図の数値はtruncate線形マスクであり、第1バイトから順に並べたものの16進表示である。F関数の下の、「+」の後の数値はactive S-boxの増加数を表す。今回の（SP型からPS型への）等価変形により、始端にP^{-1}関数、終端にP関数が付く事になるが、これらは線形関数であり線形確率に影響を生じないため図においては省略している。

[3] $2^l \leq l(1)$の場合、右辺の値によっては、連立不等式を満足するz_1が選べない事がある。例えば、$GF(2^2)$において、z_1に対する連立不等式の数が3以上になると、それを満足する非零のz_1が選べないことがある。その場合は、非零の$z_2, \ldots, z_{m-k}$の選び方に制約が発生する。この報告書では、8ビット丸めパスを考えているので、連立不等式の数が255以上にならなければ、この様な心配はない

[4] 非零のz_1を定める連立不等式(4.17)において、z_2に要求される性質は、非零の任意の値であった事に注意

active S-box の個数を総計すると、128bit Camellia では 38 個、192/256bit Camellia では 54 個である。これらの結果と Camellia の S-box の最大線形確率は 2^{-6} という事実から Camellia の $LCP_{T_{max}}$ は次式で与えられる。

$$LCP_{T_{max}} = (2^{-6})^{38} \qquad 128bit \text{ 鍵の場合} \tag{4.18}$$

$$LCP_{T_{max}} = (2^{-6})^{54} \qquad 192/256bit \text{ 鍵の場合} \tag{4.19}$$

式 (4.18), 式 (4.19) のどちらの場合も $LCP_{T_{max}} < 2^{-\text{データブロック長}}$ なので線型近似式を用いた乱数識別攻撃に対して安全といえる。また、式 (4.18), 式 (4.19) のどちらの場合も $LCP_{T_{max}} < 2^{-\text{秘密鍵長}}$ なので鍵スケジュールが理想であれば、関連鍵攻撃まで許す線形攻撃に対して秘密鍵長相当の安全性を有するといえる。

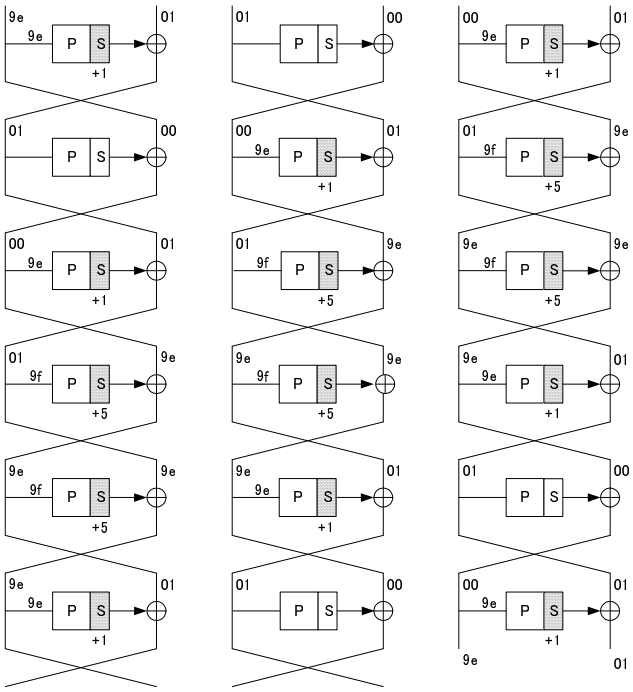

図 4.2: $LCP_{T_{max}}$ を与える線形パス例 (Key:128bit)

ラウンド毎の最小累積 active S-box 数を示せば、表 4.1 である。これより、128 ビット鍵の場合 11 段で、192bit 鍵、256bibit 鍵の場合、それぞれ 15 段、21 段で、$2^{-|K|}$ 以下の最大線形特性確率が保証される。

4.5 まとめ

Camellia の線形攻撃耐性評価の一環として、データランダム化部の最大丸め線形特性確率 $LCP_{T_{max}}$ を Viterbi 探索により導出した。結果として $LCP_{T_{max}}$ は 128bit 鍵では 2^{-228}、192/256bit 鍵では 2^{-324} となり、どちらの場合も鍵スケジュールが理想であれば、関連鍵攻撃までの攻撃条件を考えても、線形攻撃に対して耐性がある。

ラウンド	active S-box 数
1	0
2	1
3	2
4	6
5	9
6	11
7	13
8	14
9	18
10	20
11	22
12	25
13	26
14	30
15	32
16	34
17	36
18	38
19	42
20	44
21	46
22	48
23	50
24	54

表 4.1: ラウンド毎の最小累積 active S-box 数

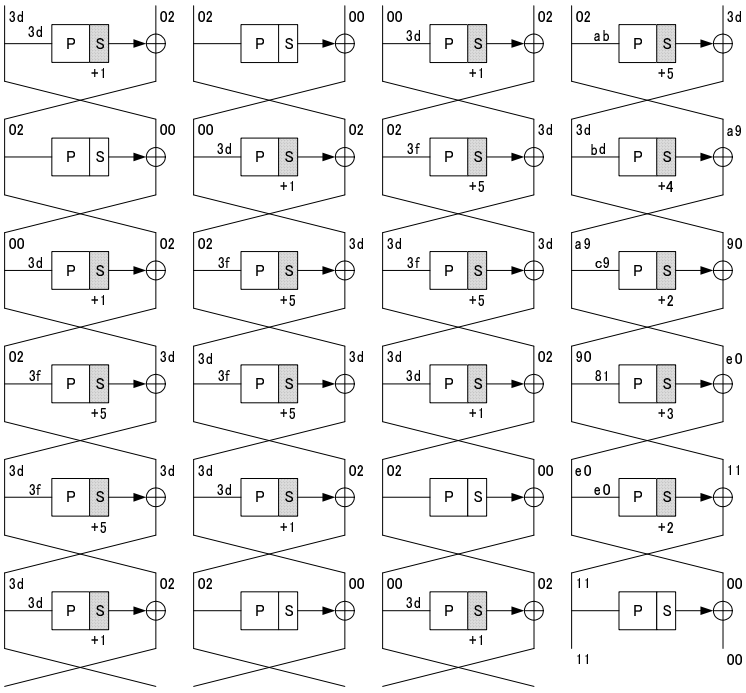

図 4.3: $LCP_{T_{max}}$ を与える線形パス例 (Key:192/256bit)

第5章　CIPHERUNICORN-A

5.1　データランダム化部

CIPHERUNICORN-A のデータ攪拌部は、Feistel 構造の 16 段構成であり、初期/終期処理として、拡大鍵の加算/減算を行なっている。詳細は、CIPHERUNICORN-A の仕様書 [5] を参照して頂きたい。ここでは、データランダム化部の最大線形特性確率の上界を求める議論に係わる部分を説明する。

5.1.1　F 関数

F 関数の構造を図 5.1 に示す。F 関数は、64 ビット入出力の関数であり、一時鍵生成部（図上半分）と本流部（図下半分) からなる二重構造を有する。64bit 入力データ $X_l||X_r$ が入力であり、拡大鍵 FK_a^i,FK_b^i が算術加算され、本流部入力、拡大鍵 SK_a^i,SK_b^i が加算され一時鍵生成部入力となる。F 関数の構成部品は、T_i $(i = 0, 1, 2, 3)$、定数乗算、$A3$ 関数、一時鍵生成部の出力による T_i の選択である。

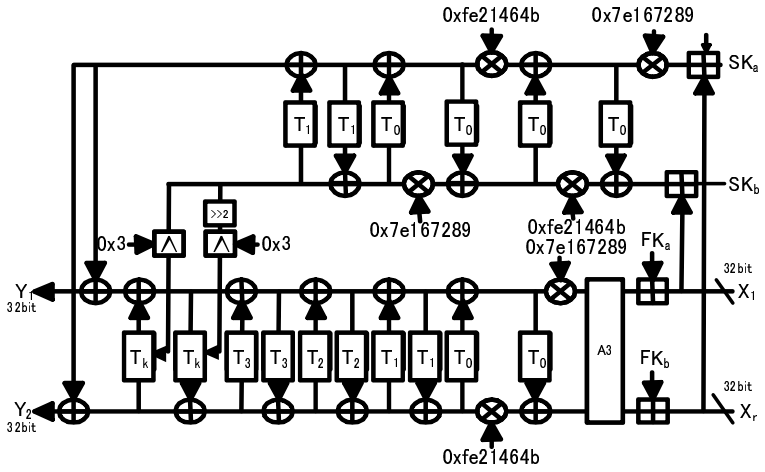

図 5.1: F 関数

5.1.2 A3 関数

A3 関数は、左巡回シフトと排他的論理和からなる線形関数であり、64 ビット入力 X を、64 ビット出力 Y に変換する。

$$Y = (X <<< const0) \oplus (X <<< const1) \oplus (X <<< const2) \tag{5.1}$$

$$const0 \ = \ 0 \ , \ const1 \ = \ 23 \ , \ const2 \ = \ 41 \tag{5.2}$$

5.1.3 定数乗算

図 5.1 の中で $\bigotimes$ で表される $\mod 2^{32}$ の算術乗算であり、定数は 2 種類ある。

$$Y = X \otimes const\, n \ (n = 0, 1) \tag{5.3}$$

$$const0 = 0x7e167289 \ , \ const1 = 0xfe21464b \tag{5.4}$$

$$\tag{5.5}$$

5.1.4 T_i 関数

T_i 関数を図 5.2 に示す。4 つの S-box$S_0, S_1.S_2, S_3$ の並列回路であり、添え字 i で選択されたバイト位置の 8 ビットを入力とし、32 ビットデータを出力する。

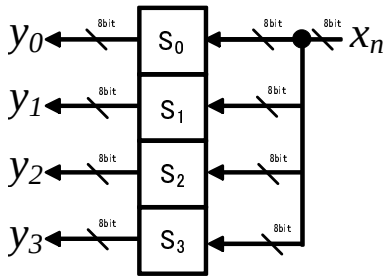

図 5.2: T 関数

5.2 Feistel 構造の最大線形特性確率の上界

Feistel 型暗号[1]においては、ラウンド関数が 2 段連続で passive なパス[2]、差分パスにおいては、入力差分 $\Delta X = 0$ は、自明パスである。非自明な線形パスの線形特性確率の最大値が、最大線形特性確率である。

ラウンド関数 F において、零入力マスク $\Gamma_x = 0$、非零出力マスク Γ_y の線形パスが繋がり、その（特性）確率の最大値を、$P0 = \max_{\Gamma_y \neq 0} LP(0 \rightarrow \Gamma_y)$ とすると、データランダム化部において、図 5.3 の線形パスが繋がる。これを 2 段繰り返し型パスといい、連続する 2 段にの中に、1 段

[1]DES 型暗号。一般化 Feistel ではない
[2]線形パスにおいては、出力マスク $\Gamma_y = 0$

分 passive な F 関数が存在する。データランダム化部の段数を R 段とするならば、全体の線形特性確率は、$P_0^{R/2}$ となる。

ラウンド関数の線形（特性）確率の最大値を、$P1 = \max_{\Gamma_y \neq 0, \Gamma_x} LP(\Gamma_x \rightarrow \Gamma_y)$ とすると、図 5.4 の線形パスが繋がる可能性がある。このパスにおいて、3 段当たり 1 段分 passive な F 関数が存在する。この形のパスに関し、全体の最大線形特性確率の上界は、$P_1^{2R/3}$ となる。4 段当たり 1 段 passive であれば、上界は、$P_1^{3R/4}$、5 段当たり 1 段 passive であれば、上界は、$P_1^{4R/5}$、$\cdots$ 、となる。

以上を総合して、R 段 Feistel 型暗号の最大線形特性確率 LCP_{max} の上界は、

$$LCP_{max} \leq \max(P_0^{R/2}, P_1^{2R/3}) \tag{5.6}$$

である。CIPHERUNICORN-A のラウンド数 16 を使えば

$$LCP_{max} \leq \max(P_0^{8}, P_1^{10}) \tag{5.7}$$

となる。ここでは、この式で、最大線形特性確率の上界を評価する。

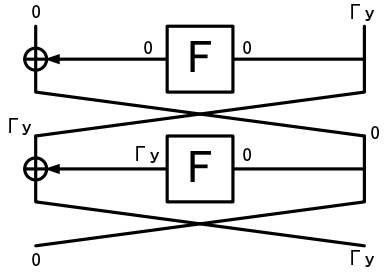

図 5.3: 入力マスクが 0 の場合

5.3 mF'' 関数

CIPHERUNICORN-A のラウンド関数 F は、複雑な構造を持つので、自己評価書や CRYPTREC の評価報告 [10] では、mF 関数に置き換えて、評価を行っている。そこでは、算術加算が排他的論理和に、定数乗算が最上位バイトにマスクを集める排他的論理和に A3 関数が、任意のマスク通過を許す線形関数に、一時鍵で選択される T_k 関数が、攻撃者に都合の良いマスク伝播を確率 1 で許す関数に近似されている。文献 [11] では、T_n 関数が active になる場所で場合分けした上で、実質的に定数乗算の近似を回避した関数 mF' を使って評価している[3]、

ここでは、より詳細に評価するために、新しい近似関数として図 5.5 に示す mF'' 関数を用いる。従来の近似手法との違いは、A3 関数と定数乗算において、bit 単位のパス接続可能性に極力配慮した線形丸めパスに対し、拡大鍵入力の独立性を考慮して評価した事である。

5.4 部品関数の特性

ここでは、8 ビット丸め線形パスの線形確率として、部品関数の最大線形確率をまとめる。

[3]最大差分特性確率の上界の再評価として mF' が、考察されたと思われる。

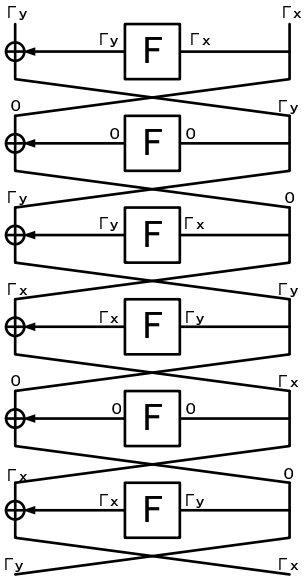

図 5.4: 入力マスクが非 0 の場合

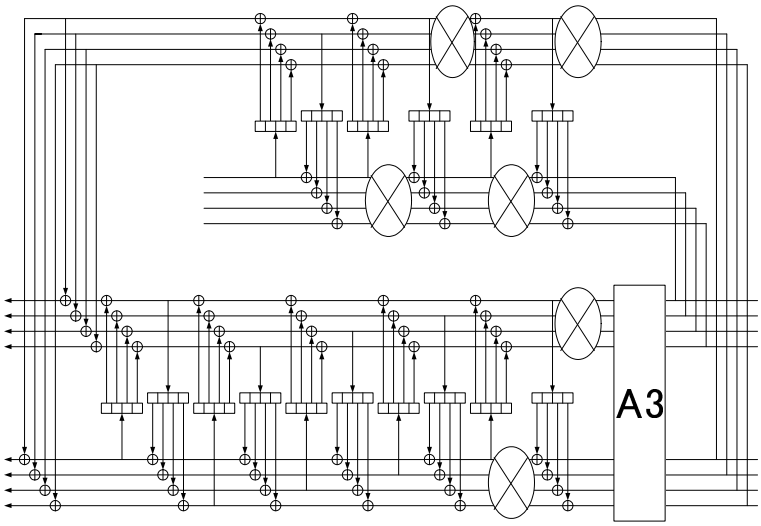

図 5.5: mF” 関数

5.4.1 T_n 関数

T_n 関数は、4 つの S-box(S_0, S_1, S_2, S_3) で構成される 1 バイト入力 4 バイト出力の関数である。どの S-box がアクティブになるかによって、最大線形確率は異なる。T_n 関数入力マスクの零非とアクティブになる S-box の組み合わせに対し、最大線形確率を表 5.1 に示す。

表 5.1: T_n 関数の最大線形確率

連結状態	入力マスク 0 の場合	入力マスク非 0 の場合
S_0	-	$2^{-6.000}$
S_1	-	$2^{-6.000}$
S_2	-	$2^{-6.000}$
S_3	-	$2^{-6.000}$
$S_0 \| \| S_1$	$2^{-3.825}$	$2^{-3.081}$
$S_0 \| \| S_2$	$2^{-3.660}$	$2^{-3.081}$
$S_0 \| \| S_3$	$2^{-3.504}$	$2^{-3.081}$
$S_1 \| \| S_2$	$2^{-3.504}$	$2^{-3.081}$
$S_1 \| \| S_3$	$2^{-3.825}$	$2^{-3.081}$
$S_2 \| \| S_3$	$2^{-3.660}$	$2^{-3.215}$
$S_0 \| \| S_1 \| \| S_2$	$2^{-3.215}$	$2^{-2.599}$
$S_0 \| \| S_1 \| \| S_3$	$2^{-3.215}$	$2^{-2.599}$
$S_0 \| \| S_2 \| \| S_3$	$2^{-3.215}$	$2^{-2.712}$
$S_1 \| \| S_2 \| \| S_3$	$2^{-3.081}$	$2^{-2.712}$
$S_0 \| \| S_1 \| \| S_2 \| \| S_3$	$2^{-2.712}$	$2^{-2.385}$

5.4.2 A3 関数

A3 関数は、線形関数であり式 (5.2) をビット単位に、行列とベクトルを使い表せば、次式である。式中の罫線は、丸めパスを意識した、8 ビット区切りである。

$$
\begin{pmatrix}
y_{63}\\ y_{62}\\ y_{61}\\ y_{60}\\ y_{59}\\ y_{58}\\ y_{57}\\ \underline{y_{56}}\\
y_{55}\\ y_{54}\\ y_{53}\\ y_{52}\\ y_{51}\\ y_{50}\\ y_{49}\\ \underline{y_{48}}\\
y_{47}\\ y_{46}\\ y_{45}\\ y_{44}\\ y_{43}\\ y_{42}\\ y_{41}\\ \underline{y_{40}}\\
y_{39}\\ y_{38}\\ y_{37}\\ y_{36}\\ y_{35}\\ y_{34}\\ y_{33}\\ \underline{y_{32}}\\
y_{31}\\ y_{30}\\ y_{29}\\ y_{28}\\ y_{27}\\ y_{26}\\ y_{25}\\ \underline{y_{24}}\\
y_{23}\\ y_{22}\\ y_{21}\\ y_{20}\\ y_{19}\\ y_{18}\\ y_{17}\\ \underline{y_{16}}\\
y_{15}\\ y_{14}\\ y_{13}\\ y_{12}\\ y_{11}\\ y_{10}\\ y_{9}\\ \underline{y_{8}}\\
y_{7}\\ y_{6}\\ y_{5}\\ y_{4}\\ y_{3}\\ y_{2}\\ y_{1}\\ y_{0}
\end{pmatrix}
=
\left[\begin{array}{cccccccc}
10000000 & 00000000 & 00000001 & 00000000 & 00000000 & 01000000 & 00000000 & 00000000\\
01000000 & 00000000 & 00000000 & 10000000 & 00000000 & 00100000 & 00000000 & 00000000\\
00100000 & 00000000 & 00000000 & 01000000 & 00000000 & 00010000 & 00000000 & 00000000\\
00010000 & 00000000 & 00000000 & 00100000 & 00000000 & 00001000 & 00000000 & 00000000\\
00001000 & 00000000 & 00000000 & 00010000 & 00000000 & 00000100 & 00000000 & 00000000\\
00000100 & 00000000 & 00000000 & 00001000 & 00000000 & 00000010 & 00000000 & 00000000\\
00000010 & 00000000 & 00000000 & 00000100 & 00000000 & 00000001 & 00000000 & 00000000\\
00000001 & 00000000 & 00000000 & 00000010 & 00000000 & 00000000 & 10000000 & 00000000\\
\hline
00000000 & 10000000 & 00000000 & 00000001 & 00000000 & 00000000 & 01000000 & 00000000\\
00000000 & 01000000 & 00000000 & 00000000 & 10000000 & 00000000 & 00100000 & 00000000\\
00000000 & 00100000 & 00000000 & 00000000 & 01000000 & 00000000 & 00010000 & 00000000\\
00000000 & 00010000 & 00000000 & 00000000 & 00100000 & 00000000 & 00001000 & 00000000\\
00000000 & 00001000 & 00000000 & 00000000 & 00010000 & 00000000 & 00000100 & 00000000\\
00000000 & 00000100 & 00000000 & 00000000 & 00001000 & 00000000 & 00000010 & 00000000\\
00000000 & 00000010 & 00000000 & 00000000 & 00000100 & 00000000 & 00000001 & 00000000\\
00000000 & 00000001 & 00000000 & 00000000 & 00000010 & 00000000 & 00000000 & 10000000\\
\hline
00000000 & 00000000 & 10000000 & 00000000 & 00000001 & 00000000 & 00000000 & 01000000\\
00000000 & 00000000 & 01000000 & 00000000 & 00000000 & 10000000 & 00000000 & 00100000\\
00000000 & 00000000 & 00100000 & 00000000 & 00000000 & 01000000 & 00000000 & 00010000\\
00000000 & 00000000 & 00010000 & 00000000 & 00000000 & 00100000 & 00000000 & 00001000\\
00000000 & 00000000 & 00001000 & 00000000 & 00000000 & 00010000 & 00000000 & 00000100\\
00000000 & 00000000 & 00000100 & 00000000 & 00000000 & 00001000 & 00000000 & 00000010\\
00000000 & 00000000 & 00000010 & 00000000 & 00000000 & 00000100 & 00000000 & 00000001\\
10000000 & 00000000 & 00000001 & 00000000 & 00000000 & 00000010 & 00000000 & 00000000\\
\hline
01000000 & 00000000 & 00000000 & 10000000 & 00000000 & 00000001 & 00000000 & 00000000\\
00100000 & 00000000 & 00000000 & 01000000 & 00000000 & 00000000 & 10000000 & 00000000\\
00010000 & 00000000 & 00000000 & 00100000 & 00000000 & 00000000 & 01000000 & 00000000\\
00001000 & 00000000 & 00000000 & 00010000 & 00000000 & 00000000 & 00100000 & 00000000\\
00000100 & 00000000 & 00000000 & 00001000 & 00000000 & 00000000 & 00010000 & 00000000\\
00000010 & 00000000 & 00000000 & 00000100 & 00000000 & 00000000 & 00001000 & 00000000\\
00000001 & 00000000 & 00000000 & 00000010 & 00000000 & 00000000 & 00000100 & 00000000\\
00000000 & 10000000 & 00000000 & 00000001 & 00000000 & 00000000 & 00000010 & 00000000\\
\hline
00000000 & 01000000 & 00000000 & 00000000 & 10000000 & 00000000 & 00000001 & 00000000\\
00000000 & 00100000 & 00000000 & 00000000 & 01000000 & 00000000 & 00000000 & 10000000\\
00000000 & 00010000 & 00000000 & 00000000 & 00100000 & 00000000 & 00000000 & 01000000\\
00000000 & 00001000 & 00000000 & 00000000 & 00010000 & 00000000 & 00000000 & 00100000\\
00000000 & 00000100 & 00000000 & 00000000 & 00001000 & 00000000 & 00000000 & 00010000\\
00000000 & 00000010 & 00000000 & 00000000 & 00000100 & 00000000 & 00000000 & 00001000\\
00000000 & 00000001 & 00000000 & 00000000 & 00000010 & 00000000 & 00000000 & 00000100\\
00000000 & 00000000 & 10000000 & 00000000 & 00000001 & 00000000 & 00000000 & 00000010\\
\hline
00000000 & 00000000 & 01000000 & 00000000 & 00000000 & 10000000 & 00000000 & 00000001\\
10000000 & 00000000 & 00100000 & 00000000 & 00000000 & 01000000 & 00000000 & 00000000\\
01000000 & 00000000 & 00010000 & 00000000 & 00000000 & 00100000 & 00000000 & 00000000\\
00100000 & 00000000 & 00001000 & 00000000 & 00000000 & 00010000 & 00000000 & 00000000\\
00010000 & 00000000 & 00000100 & 00000000 & 00000000 & 00001000 & 00000000 & 00000000\\
00001000 & 00000000 & 00000010 & 00000000 & 00000000 & 00000100 & 00000000 & 00000000\\
00000100 & 00000000 & 00000001 & 00000000 & 00000000 & 00000010 & 00000000 & 00000000\\
00000010 & 00000000 & 00000000 & 10000000 & 00000000 & 00000001 & 00000000 & 00000000\\
\hline
00000001 & 00000000 & 00000000 & 01000000 & 00000000 & 00000000 & 10000000 & 00000000\\
00000000 & 10000000 & 00000000 & 00100000 & 00000000 & 00000000 & 01000000 & 00000000\\
00000000 & 01000000 & 00000000 & 00010000 & 00000000 & 00000000 & 00100000 & 00000000\\
00000000 & 00100000 & 00000000 & 00001000 & 00000000 & 00000000 & 00010000 & 00000000\\
00000000 & 00010000 & 00000000 & 00000100 & 00000000 & 00000000 & 00001000 & 00000000\\
00000000 & 00001000 & 00000000 & 00000010 & 00000000 & 00000000 & 00000100 & 00000000\\
00000000 & 00000100 & 00000000 & 00000001 & 00000000 & 00000000 & 00000010 & 00000000\\
00000000 & 00000010 & 00000000 & 00000000 & 10000000 & 00000000 & 00000001 & 00000000\\
\hline
00000000 & 00000001 & 00000000 & 00000000 & 01000000 & 00000000 & 00000000 & 10000000\\
00000000 & 00000000 & 10000000 & 00000000 & 00100000 & 00000000 & 00000000 & 01000000\\
00000000 & 00000000 & 01000000 & 00000000 & 00010000 & 00000000 & 00000000 & 00100000\\
00000000 & 00000000 & 00100000 & 00000000 & 00001000 & 00000000 & 00000000 & 00010000\\
00000000 & 00000000 & 00010000 & 00000000 & 00000100 & 00000000 & 00000000 & 00001000\\
00000000 & 00000000 & 00001000 & 00000000 & 00000010 & 00000000 & 00000000 & 00000100\\
00000000 & 00000000 & 00000100 & 00000000 & 00000001 & 00000000 & 00000000 & 00000010\\
00000000 & 00000000 & 00000010 & 00000000 & 00000000 & 10000000 & 00000000 & 00000001
\end{array}\right]
\begin{pmatrix}
x_{63}\\ x_{62}\\ x_{61}\\ x_{60}\\ x_{59}\\ x_{58}\\ x_{57}\\ \underline{x_{56}}\\
x_{55}\\ x_{54}\\ x_{53}\\ x_{52}\\ x_{51}\\ x_{50}\\ x_{49}\\ \underline{x_{48}}\\
x_{47}\\ x_{46}\\ x_{45}\\ x_{44}\\ x_{43}\\ x_{42}\\ x_{41}\\ \underline{x_{40}}\\
x_{39}\\ x_{38}\\ x_{37}\\ x_{36}\\ x_{35}\\ x_{34}\\ x_{33}\\ \underline{x_{32}}\\
x_{31}\\ x_{30}\\ x_{29}\\ x_{28}\\ x_{27}\\ x_{26}\\ x_{25}\\ \underline{x_{24}}\\
x_{23}\\ x_{22}\\ x_{21}\\ x_{20}\\ x_{19}\\ x_{18}\\ x_{17}\\ \underline{x_{16}}\\
x_{15}\\ x_{14}\\ x_{13}\\ x_{12}\\ x_{11}\\ x_{10}\\ x_{9}\\ \underline{x_{8}}\\
x_{7}\\ x_{6}\\ x_{5}\\ x_{4}\\ x_{3}\\ x_{2}\\ x_{1}\\ x_{0}
\end{pmatrix}
\tag{5.8}
$$

A3 関数は、線形関数であり、丸め線形パス $\bar{\Gamma_x} \to \bar{\Gamma_y}$ は、確率 1 で繋がるか、又は繋がらないかの何れかである。真の線形パスとして繋がり得ない丸め線形パス（虚パス) を排除する。手法は、第 4.2 に準じた方法で行う。A3 関数として与えられた行列は $GF(2)$ 上の行列であり、連立方程式の処理は、$GF(2)$ で行われる。その為、**検査条件 1(**$\bar{x_i} = 1$ **の検査)**や**検査条件 2(**$\bar{y_j} = 1$ **の検査)**において、行列の処理で、確認出来るのは、マスク Γ_x や Γ_y の対応するビットが連立方程式の解として 0 に固定されるか否かである。丸めビットサイズを l として、丸め入力マスク $\bar{\Gamma_x} = (\bar{x_1}, \ldots, \bar{x_M})$、$\bar{\Gamma_y} = (\bar{y_1}, \ldots, \bar{x_N})$ 、各丸めマスク $\bar{x_i}$ 等に対応する l 次元マスクとして $x_i = (x_{i1}, x_{i2}, \ldots, x_{il})$ 等とする。この時、検査条件 1 は、$\bar{x_i} = 1$ に対応する全マスクビット $(x_{i1}, x_{i2}, \ldots, x_{il})$ が 0 に固定される時、パスが接続不可と判断する。検査条件 2 も同様である。即ち、丸めマスクに対応するビット区切り内の全マスクビットの検査条件が不合格の場合、パスの接続が不可と判断する。

検査条件 1 及び 2 で除外されたパス $\bar{\boldsymbol{X}} \to \bar{\boldsymbol{Y}}$ が、ビット単位に接続する事は無いが、拡大体上の線形変換として $\mathbf{M}$ が与えられた場合と異なり、除外されずに残ったパスが、必ずビット単位に接続するパスを持つ事は、まだ証明されていない。

5.4.3 定数乗算

定数乗算 $Y = const \times X$ は、2^{32} を法とする定数乗算であり、入力 X の各ビットの値が、それより下位の出力ビットに影響を与える事はない。従って、入力マスク Γ_x において、1 の立っている最上位ビット位置を i_{max}、出力マスク Γ_y において、1 の立っている最上位ビット位置を o_{max} としたとき、線形確率 LP は、$i_{max} > o_{max}$ の時 0 となる。また、CIPHERUNICORN-A の定数は、2^{32} と互いに素であり、逆元 $const^{-1}$ が存在し、$X = const^{-1} \times Y$ である。この式においても X と Y の対応関係は同じであり前と同じマスクに対しては、同じ線形確率となる。従って $i_{max} < o_{max}$ においても、$LP = 0$ となる。これより、算術加算の場合と同じく、$i_{max} = o_{max}$ 以外は、$LP = 0$ という maxbit 条件が成立する。明らかに、丸めマスクにおいても maxbit 条件が満たされなければ、$LP = 0$ となる。

この考察より、CIPHERUNICORN-A の定数乗算において、8 ビット丸め入力マスク $\bar{\Gamma}_x$ と丸め出力マスク $\bar{\Gamma}_y$ について 0000 から 1111 まで、全ての組み合わせにおいて、0 で無い線形確率を持つのは、表 5.2 で、●で示した組み合わせのみである。今回の評価においては、定数乗算で接続可能なパスについては、確率 1 と評価した。

5.5 解析結果

以上の準備の上で、8 ビット丸め評価で、Viterbi 探索を行い、mF” 関数の丸め線形特性確率の最大値を求めた。なお、一時鍵生成部の出力によって選択される T_i 関数は、攻撃者に取って都合の良い入力バイトを選ぶとした。また、独立鍵条件の考察は、文献 [11] と同じく、必要に応じ定数乗算の後方に、拡大鍵を移動して判断している。

得られた丸め線形特性確率の最大値（最大線形特性確率の上界）は、以下である。パスとしては、文献 [11] とは、異なるものが得られているが、特性確率は同一である。

- 入力マスクが 0 のとき $P_0 = 2^{-21.369}$
- 入力マスク非 0 のとき $P_1 = 2^{-21.036}$

これより、式 (5.7) で評価すれば、入力マスクが 0 の時のパスで、最大線形特性確率の上界は評価され、

$$LP_{max} \leq LP_{Tmax} = P_0^8 = (2^{-21.369})^8 = 2^{-170.95} \tag{5.9}$$

である。逆に、LCP_{Tmax} が 2^{-128} を初めて下回るラウンド数を、式 (5.7) から評価すれば、16 段中 12 段である。

この時の、mF” 関数における丸めパスを図 5.6 に示す。ラウンド鍵 FK,SK が一様にランダムであるとすると、等価変形して鍵移動をする事でアクティブとなっている非線形関数の入力は独立であると考えられる。この図において、入力マスク 0 で、T 関数内の 4 つの S-box($S_0||S_1||S_2||S_3$) がアクティブになっている物が 1 つ、同じく入力マスクが非零で 4 つの S-box($S_0||S_1||S_2||S_3$) がアクティブになっている物が 7 個あるので、表 5.1 より $P_0 = 2^{-2.385} \times 2^{-2.712 \times 7} = 2^{-21.369}$ と計算される。

表 5.2: 定数乗算における接続可能な線形パス

Γ_x \ Γ_y	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	-	-	●	●	-	-	-	-	-	-	-	-	-	-	-	-
0011	-	-	●	●	-	-	-	-	-	-	-	-	-	-	-	-
0100	-	-	-	-	●	●	●	●	-	-	-	-	-	-	-	-
0101	-	-	-	-	●	●	●	●	-	-	-	-	-	-	-	-
0110	-	-	-	-	●	●	●	●	-	-	-	-	-	-	-	-
0111	-	-	-	-	●	●	●	●	-	-	-	-	-	-	-	-
1000	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1001	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1010	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1011	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1100	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1101	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1110	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●
1111	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	●

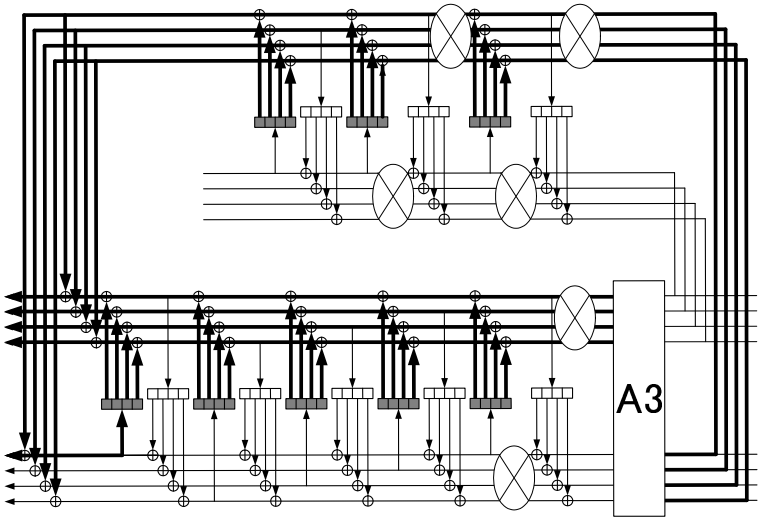

図 5.6: 入力マスクが零の場合の線形特性確率が最大となる経路

5.6 まとめ

ここでは、簡略化したラウンド関数 mF" 関数を用い、共通鍵ブロック暗号 CIPHERUNICORN-A の線形攻撃耐性を検討した。評価は 8bit 丸め線形解析で行い、データランダム化部の最大丸め線形特性確率 $LCP_{T_{max}} = 2^{-170.95}$ を得ている。この確率が初めて 2^{-128} を下回るのは 16 段中 12 段である。CIPHERUNICORN-A は、鍵長によらず、同一ラウンド数なので、この数字は、簡略化したラウンド関数であっても、128bit 鍵相当の強度がある事を示している。192/256bit 鍵の場合、この評価では $2^{-192}/2^{-256}$ より大きな線形特性確率の上界となっている。

第6章　Hierocrypt-3

6.1　部品関数の特性

Hierocrypt-3は、東芝より提案された128ビットブロック暗号であり、鍵長は、128/192/256ビットの3種類をサポートしている [6]。そのデータランダム化部では入れ子型SPN 構造を採用している。構成部品の非線形関数は、8 ビット入出力の S-box　1 種類を用いている。二重の入れ子構造は 2 重構造であり、その下位構造は、4 つの S-box を持つ SP 構造である。その P 層に当たる物が、mds_L 行列と呼ばれる拡散行列であり、SPS の構造で 32 ビット入出力の非線形関数を構成している。提案者は、これを、XS-box (EXtended S-box) と呼んでいる。入れ子の上位構造では、この XS-box と拡散行列である MDS_H 行列で SP 構造を構成している。XS-box の繰り返し回数を段数として、Hierocrypt-3 のデータランダム化部の段数は、128 ビット鍵の場合 6 段、192 ビットで 7 段、256 ビットで 8 段となっている。詳細については提案書 [6] を参照されたい。

6.1.1　S-box

Hierocrypt-3 の S-box も他の暗号と同じく、拡大体 $GF(2^8)$ 上のべき乗関数と、アフィン変換の合成関数であり、線形攻撃、差分攻撃に対し最強の S-box となっている。その最大線形確率は、$LP_{max} = 2^{-6}$ である。

6.1.2　拡散行列 mds_L

行列 mds_L は、$GF(2^8)$ 上の 4×4 行列であり、分岐数 5 を持つ。ここでは、8 ビット丸め線形特性で評価を行うので、丸め入力マスク $\bar{\Gamma}_x$、丸め出力マスク $\bar{\Gamma}_x$ 共に 4 ビットベクトルである。接続可能な丸め入出力マスクを、第 4.2 節の手法で、求めた結果を表 6.1 に示す。表において 4 ビットの丸め入出力マスクは、16 進数で表示されている。表中の○は接続可能である事を表し、無印は接続不可能であることを示す。mds_L は線形関数であり、接続可能な場合、線形確率 1 で繋がる（真の）入出力マスクがあるという意味になる。

6.1.3　拡散行列 MDS_H

MDS_H 行列は、16 バイトデータを線形変換する $GF(2^8)$ 上の 16×16 行列である。バイト単位の XOR 演算のみで実装できる事を意図して、その行列の要素は 0 又は 1 の 2 種類の値に限定されている。この行列に対し、8 ビット丸め入力マスク $\bar{\Gamma}_x$、及び出力マスク $\bar{\Gamma}_x$ は、16 ビットベクトルとなる。接続可能な丸め入出力マスクの組み合わせを、第 4.2 節の手法で求めた。MDS_H 行列は、$GF(2^4)$ の 4×4 巡回型 MDS 行列を元に構成されており、4 行及び 4 列単位の巡回構造を持つ。それを使って、第 4.2 節の手法で調査する際の丸め入出力マスクの組み合わせの種類数を削減出来る。

表 6.1: mds_L 関数の線形特性

		入力マスク															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
出力マスク	0	○															
	1																○
	2																○
	3								○				○		○	○	○
	4																○
	5								○				○		○	○	○
	6								○				○		○	○	○
	7				○		○	○	○		○	○	○	○	○	○	○
	8																○
	9								○				○		○	○	○
	A								○				○		○	○	○
	B				○		○	○	○		○	○	○	○	○	○	○
	C								○				○		○	○	○
	D				○		○	○	○		○	○	○	○	○	○	○
	E				○		○	○	○		○	○	○	○	○	○	○
	F		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

MDS_H 行列の入出力の特性の一部を、表 6.2 に示す。表中の丸め入出力マスク 16 進数で表している。

6.2 解析結果

部品関数の線形特性の解析結果を使い Viterbi 探索により、データランダム化部の最大丸め線形特性確率 LCP_{Tmax} を求めた。S-box は 1 種類であり、active S-box 数 AS を最小にするパスを求めている。段数対最小 AS 数、及び LCP_{Tmax} を表 6.3 に記す。LCP_{Tmax} が最大線形特性確率 LCP_{max} の上界である。表には、文献 [12] の最大線形特性確率の上界も参考として記す。

表 6.3 より、段数 6/7/8 段において、最大線形特性確率の上界は鍵長 $|K|$ に対し $2^{-|K|}$ を下回っている。以下に、各段数に対し最良丸めパスを示す。図 6.1，図 6.2，図 6.3 において、太線はパスが active であることを示している。

6.3 まとめ

Hierocrypt-3 のデータ攪拌部について、線形攻撃に対する耐性評価を最大線形特性確率の上界を、最大丸め線形特性確率 LCP_{Tmax} を Viterbi 探索により調査した。LCP_{Tmax} は、128 ビット鍵に対し 2^{-450}、192 ビット鍵に対し 2^{-480}、256 ビット鍵に対し 2^{-600} であり、十分小さい。鍵スケジュールが理想であれば、関連鍵攻撃までの攻撃条件を考えても、線形攻撃に対し耐性を持つ。

表 6.2: MDS_H 行列の入出力特性（抜粋）

丸め出力マスク	接続可能な丸め入力マスク
0001	6EEB
00F0	5534，55FE，773D，77F7，77FF，AAFF，BB9F，BBEA，BBFF，CCDF，$\cdots$
0F00	5345，5FE5，73D7，7F77，7FF7，AFFA，B9FB，BEAB，BFFB，CDFC，$\cdots$
388C	0070，0578，0B74，21F7，52F3，7374，9271，B3F6，C0F2，D2F3，$\cdots$
55EA	0100，0B00，1125，218C，31A9，41B3，5196，613F，711A，812B，910E，$\cdots$
5EA5	1000，10E9，114E，3010，50D7，702E，83BF，9010，A2A8，B000，$\cdots$
A55E	0010，00B0，027E，1030，2153，3173，4132，5112，6071，7051，$\cdots$
F000	3455，3D77，69EE，75DD，7CFF，7DFF，9FBB，B6FF，BFDD，BFFF，$\cdots$

表 6.3: 最大線形特性確率の上限

鍵長	段数	最少 AS 数	$LCP_{T_{max}}[log_2]$	文献 [12]
-	1	5	-30	-
-	2	25	-150	-150
-	3	30	-180	-180
-	4	50	-300	-300
-	5	55	-330	-
128 ビット	6	75	-450	-
192 ビット	7	80	-480	-
256 ビット	8	100	-600	-

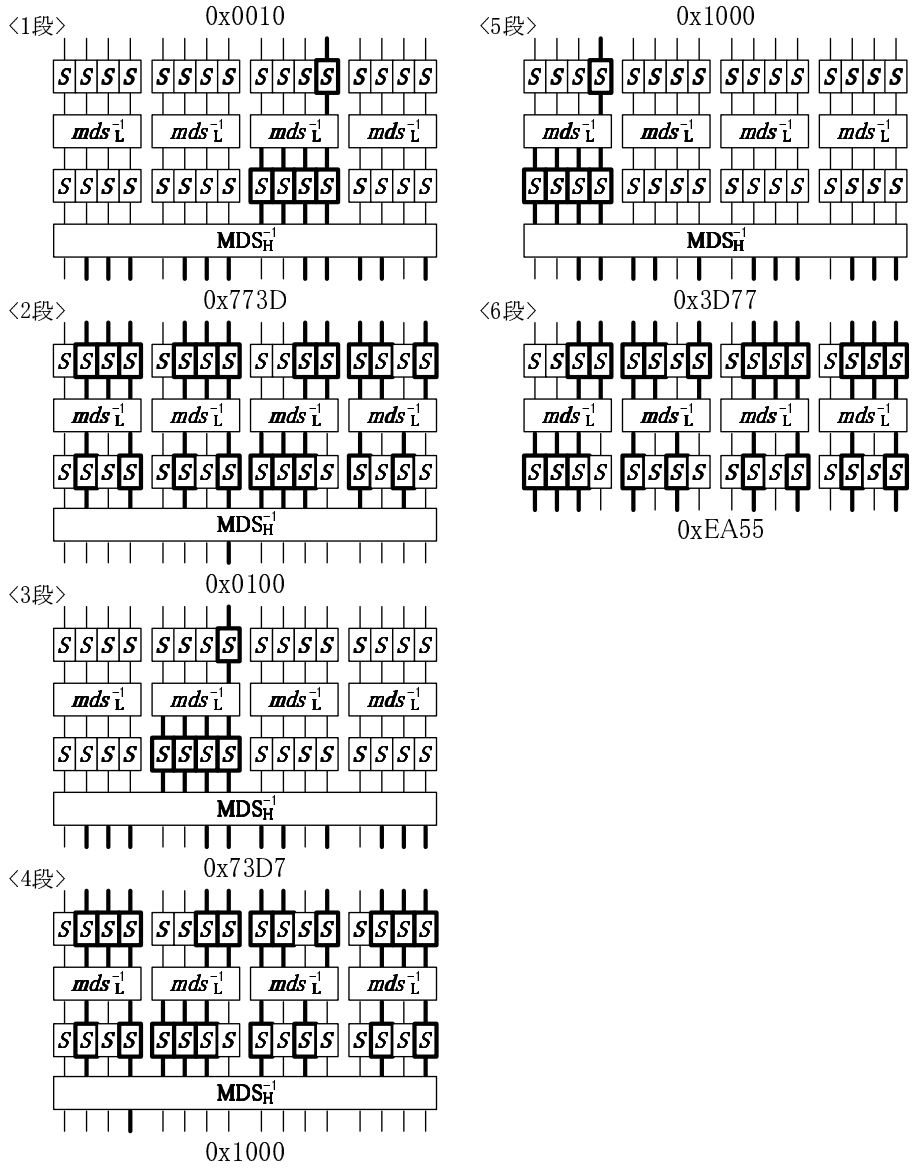
〈1段〉
0x0010
〈2段〉
0x773D
〈3段〉
0x0100
〈4段〉
0x73D7
0x1000
〈5段〉
0x1000
〈6段〉
0x3D77
0xEA55

図 6.1: 6 段の最良パス

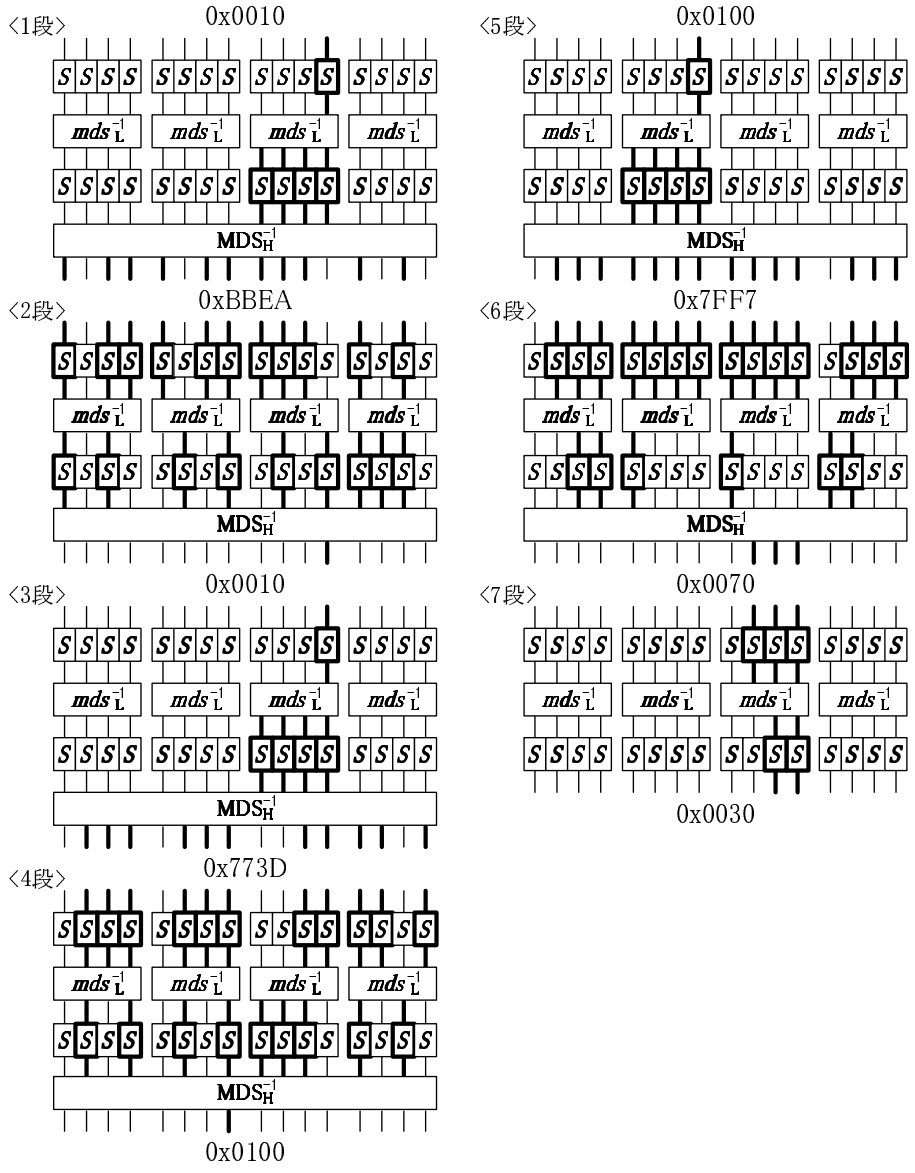
〈1段〉
0x0010
〈2段〉
0xBBEA
〈3段〉
0x0010
〈4段〉
0x773D
0x0100
〈5段〉
0x0100
〈6段〉
0x7FF7
〈7段〉
0x0070
0x0030
S
mds$^{-1}_{L}$
MDS$^{-1}_{H}$

図 6.2: 7 段の最良パス

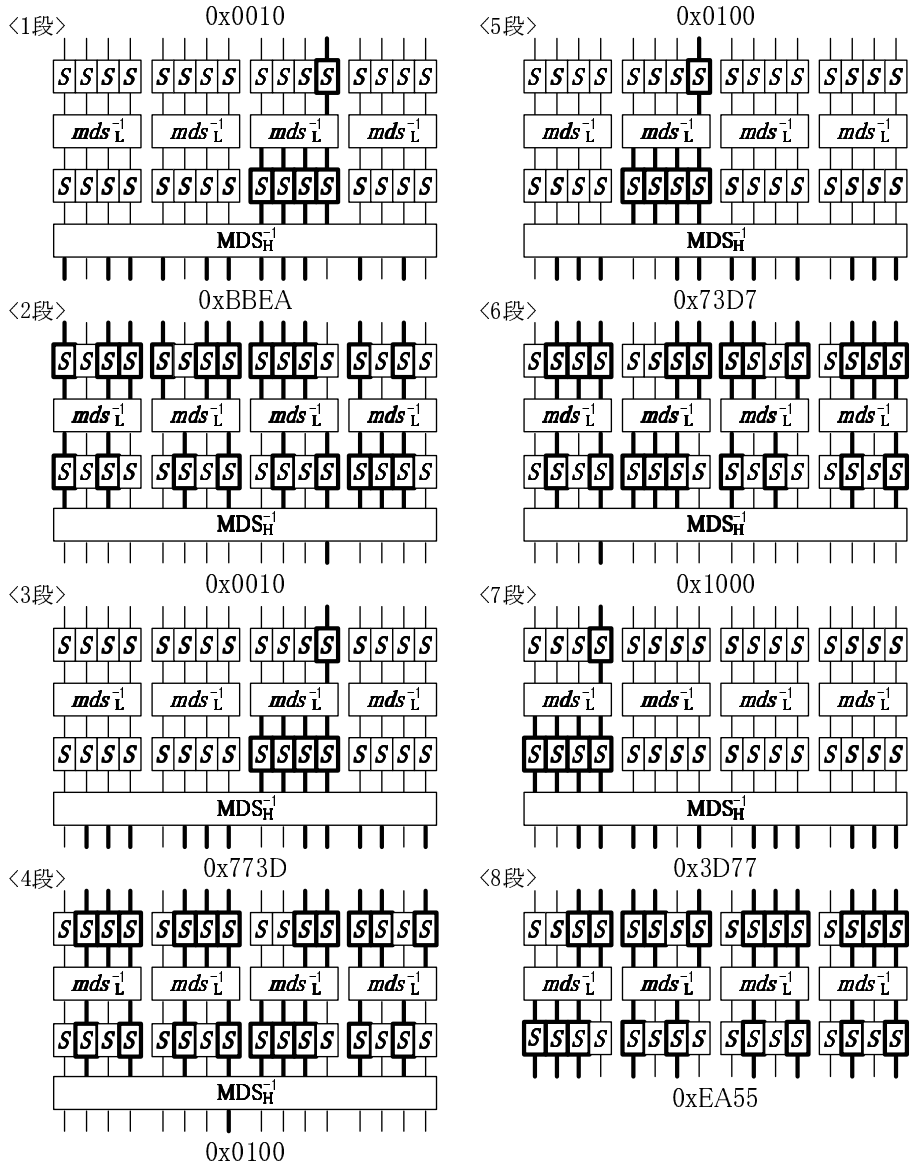
〈1段〉
0x0010
〈2段〉
0xBBEA
〈3段〉
0x0010
〈4段〉
0x773D
0x0100
〈5段〉
0x0100
〈6段〉
0x73D7
〈7段〉
0x1000
〈8段〉
0x3D77
0xEA55
S
mds$^{-1}_{L}$
MDS$^{-1}_{H}$

図 6.3: 8 段の最良パス

第7章　SC2000

7.1　データ攪拌部

SC2000 のデータ攪拌部は、Feistel と SPN の混合構造で構成されている。データ攪拌部の関数は、(32bit × 4) の入出力である I 関数、B 関数、R 関数の 3 種類である。以下、概略を記すが、詳細は、仕様書 [7] を参照されたい。I 関数は、拡大鍵の排他的論理和である。B 関数は、SPN 型で、R 関数は、Feistel 型である。データ攪拌部は、$I - B - R - R$ の繰り返しである。線形攻撃耐性に関わるのは、B 関数と R 関数である。

R 関数のラウンド関数は、64bit 入出力の全単射関数 F であり、パラメータ $mask$ により、2 種類の関数 R_3,R_5 に分けられる。引き続く R 関数 2 段の間に 2 つの 64 ビットデータの左右入れ替えをする SWAP が存在する。SWAP を × と表せば、B、R 関数の接続のみを表示すれば、128 ビット鍵の場合、

$$B - R_5 \times R_5 - B - R_3 \times R_3 - B - R_5 \times R_5 - B - R_3 \times R_3 - B - R_5 \times R_5 - B - R_3 \times R_3 - B \quad (7.1)$$

の 19 段である。192/256 ビット鍵の場合は、

$$B - R_5 \times R_5 - B - R_3 \times R_3 - B - R_5 \times R_5 - B - R_3 \times R_3 - B - R_5 \times R_5 - B - R_3 \times R_3 - B - R_5 \times R_5 - B \quad (7.2)$$

の 22 段である。R_5 と R_3 を区別して考えれば、$B - R_5 \times R_5 - B - R_3 \times R_3 -$ の 6 段周期となる。

7.1.1　F 関数

F 関数の中では、32 ビットづつ 2 系統の処理が、S 関数、M 関数で行われ、最後に L 関数で 2 つの 32 ビットが混ぜ合わされる。M は、$GF(2)$ 上の 32×32 行列 $\mathbf{M}$ による線形変換である。L 関数は、32 ビット 2 ワードを処理し、$mask$ 依存の線形変換となっている。S 関数では、32bit の入力を (6,5,5,5,5,6)bit の 6 系統に分割し、6 ビット変数は、6bit テーブル S_6 で、5bit 変数は　5 ビットテーブル S_5 で変換される。

7.1.2　B 関数

B 関数は 32 ビット × 4 本の入力データ (a, b, c, d) を、ビットスライス構造で処理する。a, b, c, d の 4 ワードにおける同一ビット位置のデータを 4bit 毎にまとめ、32 個の 4 ビット S ボックス S_4 で処理し、その出力を各ワードの同一ビット位置に戻して B 関数の出力とする。

7.2 線形特性の評価

SC2000 には、ビットサイズの異なる S_4, S_5, S_6 の 3 種類の S-box が存在する。他の暗号と同じように、S-box の入力ビット幅単位の丸め線形解析を行うと、B 関数内にビットスライス構造があるため、大幅に緩い線形特性確率の上界しか得られない。ここでは、文献 [8] に示されている 6 段繰り返し型パスを、全段に適用し、初段と最終段に、変更を加え、データランダム化部全段の最大線形特性確率の目安とする。Feistel 型暗号では、2 段又は 3 段繰り返しパスの最大丸め線形特性確率が、最大線形特性確率の上界を与える事が知られているが、SC2000 は、SPN と Feistel の混合型であり、そのような上界の導出法は知られていない。しかし、多くの場合、最良線形パスは、繰り返し型又はその修正型であり、SC2000 の場合も、繰り返し構造が、最大線形特性パスを与える事を、ここでは期待する。

7.2.1 部品の特性

ここで、取り扱う 6 段繰り返しパスを理解する為に、部品関数の特性を幾つか紹介する。

S-box

SC2000 の S-boxS_4、S_5、S_6 の線形確率を調査し、以下を得た。

- S_4 の線形確率は、0,2^{-2},2^{-4} の何れかである。
- S_5 の線形確率は、0,2^{-4} の何れかである。
- S_6 の線形確率は 0,2^{-4},$2^{-4.83}$,2^{-6},2^{-8} の何れかである。
- 任意の非零入力マスク $\Gamma_x \neq 0$ は、その S-box の最大線形確率に等しい確率のパスを持つ。

S 関数

S 関数に入力される 32bit の入力マスクを S-box サイズに応じ (6,5,5,5,5,6)bit の 6 系統に分割し $\Gamma x = (\boldsymbol{x_0}, \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}, \boldsymbol{x_5})$ とする。x_i が非零であれば対応する S-box は、active となる。

B 関数

B 関数のマスク 128 ビットを、32 ビット毎に区切り、入力マスク $\Gamma_x = (\boldsymbol{x_0}, \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3})$、出力マスク $\Gamma_y = (\boldsymbol{y_0}, \boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3})$ とする。次の性質が言える。

- 繋がる入出力マスクは $(\boldsymbol{x_0} \vee \boldsymbol{x_1} \vee \boldsymbol{x_2} \vee \boldsymbol{x_3}) = (\boldsymbol{y_0} \vee \boldsymbol{y_1} \vee \boldsymbol{y_2} \vee \boldsymbol{y_3})$ を満たす
- active な S_4 ボックス数は、$wt(\boldsymbol{y_0} \vee \boldsymbol{y_1} \vee \boldsymbol{y_2} \vee \boldsymbol{y_3})$ である。但し $wt(\cdot)$ はハミング重み。

7.2.2 6 段繰り返しパス $B - R_5 \times R_5 - B - R_3 \times R_3$

文献 [8] に示されている 6 段繰り返しパスを説明する。$B - R_5 \times R_5 - B - R_3 \times R_3$ の 6 段に渡るパスであり、入力マスク$\Gamma_x = (0x204000a2, 0x20000022, 0, 0x20400022)$ = 出力マスクΓ_y のパスである。線形特性確率は、以下に述べる確率の積であり、$LCP = 2^{-12}2^{-16}2^{-12}2^{-16} = 2^{-56}$。

1,4 段目 B 関数

1 段目入力マスク:$(0x204000a2, 0x20000022, 0, 0x20400022)$、出力マスク:$(0, 0, 0x204000a2, 0x00400000)$
4 段目入力マスク:$(0x204000a2, 0x00400000, 0, 0x20000022)$、出力マスク:$(0, 0, 0x204000a2, 0x00400000)$
5 個の S_4 ボックスが active であり、その内 1 個の確率は 2^{-4}。線形確率 $LP = 2^{-12}$

2,5 段目 R 関数

R 関数内の F 関数入力マスク=$(0, 0)$、出力マスク=$(0, 0)$。パッシブであり、確率 $LP = 1$

3 段目 R_5 関数

R_5 関数内の F 関数入力マスク=$(0, 0x20400022)$、出力マスク=$(0x204000a2, 0x00400000)$
S_6 ボックス 2 個、S_5 ボックス 1 個 active、M 関数出力のマスクは、$(0, 0x204000a2)$、線形確率 $LP = 2^{-16}$

6 段目 R_3 関数

R_3 関数内の F 関数入力マスク=$(0, 0x20400022)$、出力マスク=$(0x204000a2, 0x20000022)$
2 個の S_6 ボックス、1 個の S_5 ボックスが active。M 関数出力のマスクは、$(0, 0x204000a2)$。線形確率 $LP = 2^{-16}$

改めて、6 段繰り返しパスをまとめれば、次式である。$\overset{\frown}{\ }$ の上の数字は、線形特性確率 LCP の $-log_2(LCP)$ を表す。

$$\overbrace{B}^{12} - \overbrace{R_5}^{0} \times \overbrace{R_5}^{16} - \overbrace{B}^{12} - \overbrace{R_3}^{0} \times \overbrace{R_3}^{16} \tag{7.3}$$

7.3 解析結果

128 ビット鍵の場合の 19 段構造に、第 7.2.2 節の繰り返しパスを適用し、初段及び最終段の自由度の範囲で、最適化すると、$LCP = 2^{-176}$ となる。この値を与える入出力マスクの一例は、$(0x204000a2, 0x200000a2, 0, 0x204000a2) \rightarrow (0, 0x00000080, 0x204000a2, 0x00400080)$ である。パスを以下に示す。$\overset{\frown}{\ }$ が 6 段繰り返しパス、又はそれを元に最適化したパスを表す。

$$\overbrace{B - R_5 \times R_5 - B - R_3 \times R_3}^{54} - \overbrace{B - R_5 \times R_5 - B - R_3 \times R_3}^{56} - \overbrace{B - R_5 \times R_5 - B - R_3 \times R_3}^{56} - \overbrace{B}^{10} \tag{7.4}$$

1 段目の B 関数は、出力マスクが繰り返しパスで与えられ S_4 ボックスが 5 個 active となるが、入力マスクの選び方により先頭の 6 段繰り返しパスの特性確率を、2^{-54} に出来る。19 段目の B 関数及び 18 段目の F 関数は、F 関数の入力マスク、B 関数の入力マスクの最下位 32 ビット及び出力マスクに自由度が出る。最適化結果として、B 関数内の S_4 ボックスの出力マスクを変更し最終段の B 関数の確率は、2^{-10} である。このパスにおいて線形特性確率が、初めて 2^{-128} を下回るのは、19 段中 15 段目である。

同様に、192/256 ビット鍵の 22 段構造では、入出力マスクを最適化すると、例えば $(0x204000a2, 0x200000a2, 0, 0x204000a2) \rightarrow (0, 0x00400000, 0x204000a2, 0x20000022)$ に対し、$LCP = 2^{-204}$ である。

$$\overbrace{B - R_5 \times R_5 - B - R_3 \times R_3}^{54} - \overbrace{B - R_5 \times R_5 - B - R_3 \times R_3}^{56} - \overbrace{B - R_5 \times R_5 - B - R_3 \times R_3}^{56} - \overbrace{B - R_5 \times R_5 - B}^{38} \tag{7.5}$$

このパスの線形特性確率は、2^{-256} を上回っている。2^{-192} を初めて下回るのは、22 段中 21 段目である。

7.4 まとめ

SC2000 では、S-box として、4,5,6 ビット幅のもの 3 種類が混在する事、及びビットスライス構造を持つ事の為丸め線形パス評価では、大幅に緩い上界しか得られない。文献 [8] の 6 段繰り返しパスを全ラウンドに適用すると、128 ビット鍵の場合、線形特性確率 2^{-176} のパスが実在する。このパスにおいて、特性確率 2^{-128} を、初めて下回るのは、19 段中 15 段目である。192/256 ビット鍵の場合、特性確率 2^{-204} のパスが実在する。確率 2^{-192} を初めて下回るのは、22 段中 21 段目である。

参考文献

[1] 電子政府推奨暗号リスト http://www.cryptrec.go.jp/images/cryptrec_01.pdf

[2] CRYPTREC "CRYPTREC 報告書",http://www.ipa.go.jp/report.html

[3] NIST, "Announcing the ADVANCE ENCRYPTION STANDARD(AES)", FIPS 197, Nov. 2001. http://csrc.nist.gov/publications/fips/fips197/fips197.pdf

[4] NTT, 三菱電機 : "128 ビットブロック暗号 Camellia アルゴリズム仕様書", http://www.cryptrec.go.jp/cryptrec_03_spec_cypherlist_files/PDF/06_01jspec.pdf ,2001

[5] NEC:"暗号技術仕様書 CIPHERUNICORN-A",http://www.cryptrec.go.jp/cryptrec_03_spec_cypherlist_files/PDF/07_01jspec.pdf ,2000

[6] 東芝 : "暗号技術仕様書:Hierocrypt-3",http://www.cryptrec.go.jp/cryptrec_03_spec_cypherlist_files/PDF/08_02jspec.pdf ,2002

[7] 富士通研究所、東京理科大学 : "共通鍵ブロック暗号 SC2000 暗号技術仕様書",http://www.cryptrec.go.jp/cryptrec_03_spec_cypherlist_files/PDF/09_01jspec.pdf ,2001

[8] H.Yanami, T.Shimoyama, and O.Dunkelman, "Differential and Linear Cryptanalysis of a Reduced-Round SC2000", FSE 2002, LNCS 2365: 34-48

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

[10] 金子研究室:"共通鍵ブロック暗号 CIPHERUNICORN-A の安全性に関する詳細調査報告書",http://www.cryptrec.go.jp/estimation/rep_ID0027.pdf ,2002

[11] 角尾 幸保, 久保 博靖, 茂 真紀, 洲崎 智保, 宮内 宏, "CIPHERUNICORN-A の差分解読／線形解読に対する安全性について (II)," SCIS 2003, 予稿集, 5D-1 (2003)

[12] 盛合志帆, "Hierocrypt-3 の最大線形/差分確率および最大線形/差分特性確率について", http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/129b_HIERO3.pdf ,2001.