付録Ａ．CIPHERUNICORN-Eの高階差分及び補間攻撃について

平成 12 年 1 月 12 日

1　はじめに

本資料は、電子政府事業に用いる暗号方式に関する応募資料 (暗号技術仕様書 CIPHERUNICORN-E 及び自己評価書 CIPHERUNICORN-E) に基づき詳細評価を行うために作成したものである。アルゴリズムに関する記述は省略したので、関数名の仕様や変数名は、仕様書を参照して頂きたい。

2　CIPHERUNICORN-Eのアルゴリズム

CIPHERUNICORN-E の全体構造は以下のとおりである。(図 1 参照)

ブロック長	64 ビット
秘密鍵長	128 ビット
全体構造	Feistel 型
段数	16 段 (2 段ごとに 64 ビット幅関数 (L 関数) を使用)
鍵スケジューラ	秘密鍵を入力とする Feistel 構造

段関数であるＦ関数は、32 ビット入出力の関数であり、段当たり拡大鍵 128 ビットが供給される。Ｆ関数は本流部と一時鍵生成部からなり、前者はＴ関数１０層の従属接続と一時鍵加算部のＫ関数から構成される。Ｔ関数は４つのＳボックスで構成される。

3　高階差分解読 (Higher Order Differential Cryptanalysis)

3.1　自己評価書による評価

F 関数 (拡大鍵は全て 0 とする) の出力ビット全てについて, 代数次数 6 次までの項を調査した (表 1)。6 次の項数が十分あることから,F 関数あたりの代数次数は少なくとも 6 である。1 段目の F 関数への入力を 0 に固定した場合,3 段で 36 次となり, 解読に必要な平文を用意することは不可能になる。よって,5 段以上あれば解読は困難である。また,6 次の項数も十分であることから, 平文の一部のビットを 0 で固定して全体の代数次数を下げるような解読の適用も困難であると、自己評価書には記載されている。

表 1: F 関数ブール多項式項数

次数	平均	最大値	最小値	期待値
1 次	17	24	11	16
2 次	247	264	225	248
3 次	2478	2540	2408	2480
4 次	17946	18156	17697	17980
5 次	100762	101128	100377	100688
6 次	453003	453677	452218	453096

期待値:次数 d で取りうる項の種類 ${}_{32}C_d$ の 1/2

3.2 形式的代数次数評価

3.2.1 Sbox の代数次数

CIPHERUNICORN-E では、4 種類の S ボックス $S_i(0 \leq i \leq 3)$ が使用される。各 S ボックス入出力をブール多項式表現した。計算の結果を表 2 に示す。表の通り, 多項式の次数は全て 7 次であり, 項数はほぼ期待値である 128 を中心に分布し、全ての次数項が、各出力ビットで偏り無く使用されていると推測される。

表 2: Sbox ブール多項式次数

SBox	0	1	2	3	4	5	6	7
S_0	7	7	7	7	7	7	7	7
S_1	7	7	7	7	7	7	7	7
S_2	7	7	7	7	7	7	7	7
S_3	7	7	7	7	7	7	7	7

以下に一例として S_0 の最下位ビットを示す。

$$
\begin{aligned}
s_{0_0} \quad = \quad & x_1x_2x_3x_4x_5x_6x_7 + x_0x_1x_2x_4x_5x_6x_7 + x_2x_3x_4x_5x_6x_7 + \\
& x_0x_3x_4x_5x_6x_7 + x_1x_2x_3x_5x_6x_7 + x_0x_2x_3x_5x_6x_7 + x_0x_1x_2x_5x_6x_7 + \\
& x_1x_2x_3x_4x_6x_7 + x_0x_2x_3x_4x_5x_7 + x_0x_1x_3x_4x_5x_7 + x_0x_1x_2x_4x_5x_7 + \\
& x_0x_1x_2x_3x_4x_7 + x_1x_2x_3x_4x_5x_6 + x_0x_2x_3x_4x_5x_6 + x_0x_1x_3x_4x_5x_6 + \\
& x_0x_1x_2x_3x_5x_6 + x_0x_1x_2x_3x_4x_5 + x_3x_4x_5x_6x_7 + x_2x_4x_5x_6x_7 + \\
& x_1x_4x_5x_6x_7 + x_2x_3x_5x_6x_7 + x_1x_3x_5x_6x_7 + x_0x_3x_5x_6x_7 + \\
& x_1x_2x_4x_6x_7 + x_0x_1x_3x_6x_7 + x_0x_1x_2x_6x_7 + x_0x_3x_4x_5x_7 + \\
& x_0x_1x_4x_5x_7 + x_0x_1x_3x_5x_7 + x_0x_1x_2x_5x_7 + x_0x_2x_3x_4x_7 + \\
& x_0x_2x_4x_5x_6 + x_1x_2x_3x_5x_6 + x_0x_2x_3x_5x_6 + x_0x_1x_2x_5x_6 + \\
& x_0x_1x_2x_4x_6 + x_0x_1x_2x_3x_6 + x_1x_2x_3x_4x_5 + x_0x_1x_2x_3x_4 + x_1x_5x_6x_7 + \\
& x_2x_4x_6x_7 + x_1x_4x_6x_7 + x_1x_3x_6x_7 + x_0x_3x_6x_7 + x_1x_2x_6x_7 + \\
& x_0x_2x_6x_7 + x_2x_4x_5x_7 + x_2x_3x_5x_7 + x_1x_3x_5x_7 + x_0x_3x_5x_7 + \\
& x_1x_2x_5x_7 + x_0x_2x_5x_7 + x_2x_3x_4x_7 + x_0x_3x_4x_7 + x_0x_2x_4x_7 + \\
& x_0x_1x_4x_7 + x_0x_1x_2x_7 + x_3x_4x_5x_6 + x_2x_4x_5x_6 + x_1x_4x_5x_6 + \\
& x_0x_4x_5x_6 + x_1x_2x_5x_6 + x_0x_2x_3x_6 + x_0x_1x_2x_6 + x_1x_2x_4x_5 + \\
& x_0x_2x_4x_5 + x_1x_2x_3x_5 + x_0x_1x_3x_5 + x_0x_1x_2x_5 + x_0x_2x_3x_4 + \\
& x_0x_1x_3x_4 + x_3x_6x_7 + x_0x_6x_7 + x_3x_5x_7 + x_2x_5x_7 + x_1x_5x_7 + x_0x_5x_7 + \\
& x_3x_4x_7 + x_2x_4x_7 + x_1x_4x_7 + x_2x_3x_7 + x_0x_1x_7 + x_3x_5x_6 + x_2x_5x_6 + \\
& x_1x_5x_6 + x_3x_4x_6 + x_0x_3x_6 + x_1x_2x_6 + x_0x_1x_6 + x_3x_4x_5 + x_1x_4x_5 + \\
& x_1x_3x_5 + x_0x_2x_5 + x_0x_1x_5 + x_2x_3x_4 + x_1x_2x_4 + x_0x_2x_4 + x_1x_2x_3 + \\
& x_0x_2x_3 + x_4x_7 + x_3x_7 + x_0x_7 + x_5x_6 + x_3x_6 + x_2x_6 + x_1x_6 + x_2x_5 + \\
& x_1x_5 + x_2x_4 + x_1x_3 + x_0x_3 + x_1x_2 + x_7 + x_5 + x_3 + 1
\end{aligned}
$$

3.2.2 L 関数

L 関数は 128 ビット入力 64 ビット出力の関数で入力が $X = X_L||X_R$(64 ビット) 及び拡大鍵 $LK = LK_L||LK_R$(64 ビット) である。また, 出力は,$Z = Z_L||Z_R$(64 ビット) である。

L 関数では以下の演算が行われる。

$$Z_L = X_L \oplus (X_R \wedge LK_R) \oplus (X_L \wedge LK_L \wedge LK_R)$$

$$Z_R = X_R \oplus (X_L \wedge LK_L) \oplus (X_R \wedge LK_L \wedge LK_R)$$

となる。各ビットごとの入出力に注目してみると, 鍵によって X_L,X_R,$X_L \oplus X_R$ の 3 通りであり、入力に対する次数は 1 次である。

3.2.3 T 関数

F 関数内に使用されている T 関数は、32 ビット入出力の関数であり、4 種類ある。各T関数には、4種類のSボックス $S_i(0 \leq i \leq 3)$ が使用されている。32 ビット入力中の定められた 8 ビットが、これらSボックスに入り、他の 24 ビットと排他的論理和される構造を持つ。T 関数の次数はSボックスの次数で定まり、全て 7 次である。

3.2.4 K 関数

K 関数は入力番号 n と入力データ $X = X_0||X_1||X_2||X_3$(32 ビット) 及び一時鍵 wk を入力とし出力データ Z(32 ビット) とする関数である。それぞれ,

$$K(0, wk) = (X_0 \oplus wk)||X_1||X_2||X_3$$

$$K(1, wk) = X_0||(X_1 \oplus wk)||X_2||X_3$$

$$K(2, wk) = X_0||X_1||(X_2 \oplus wk)||X_3$$

$$K(3, wk) = X_0||X_1||X_2||(X_3 \oplus wk)$$

となっており, 全て 1 次である。

3.2.5 F 関数

F 関数の主流部に着目すると, 最初に拡大鍵が算術和され, その後 T 関数を 4 回, 再び拡大鍵を算術和し, T 関数を 4 回使用し最後に K 関数 T 関数の組み合わせを 2 回行っている。最初に T 関数を 4 回通るまでの次数の追跡を図 2 に示す。次数の追跡においては、単純な代数次数だけでなく、係わりを持つ入力ビット数による制限（次数≦入力ビット数）も併用した。

F 関数としてみた場合の次数を図 3 に示す。入力から 5 層目の T 関数出力において,32 次に達すると考えられる。

F 関数は全単射でない 32bit 入出力関数であるので,32 次となりうる。後述の実験結果のように、F関数 32 階差分は、鍵に依存する非零値であり確かに、32 次であることがわかった。

従って、L 関数の影響を無視して、平文左半分を変数にとる 32 階差分攻撃を考えても、

図 4 の様に 3 段目通過時点で暗号全体の形式的代数次数は, それぞれ 32 次以上となり, 攻撃に必要な平文が平文の全数を超えてしまい, 高階差分攻撃が適用できない。

3.3 SBox の高階差分特性

S_3〜S_0 の SBox について高階差分値を求めた。S_3〜S_0 の 4 つの全ての SBox について、１〜8 階差分値を計算し、定数となる bit を探索した。その結果、5 階差分、6 階差分の一部について、ある bit が定数となった。その結果を表 3、表 4 に示す。

表 3: SBox の 5 階差分値

変数	S_0	S_1	S_2	S_3
x_1,x_2,x_3,x_4,x_5	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxx1xx
x_1,x_2,x_3,x_5,x_7	xxxxxxxx	x1xxxxxx	xxxxxxxx	xxxxxxxx
x_0,x_3,x_4,x_5,x_7	xxxxxxxx	x1xxxxxx	xxxxxxxx	xxxxxxxx
x_1,x_3,x_4,x_5,x_7	xxxxxxxx	x1xxxxxx	xxxxxxxx	xxxxxxx1
x_0,x_1,x_3,x_6,x_7	xxxxxxx1	xxxxxxxx	xxxxxxxx	xxxxxxxx
x_0,x_1,x_4,x_6,x_7	xxxxxx1x	xxxxxxxx	xxxxx1xx	xxxxxxxx
x_0,x_1,x_5,x_6,x_7	xxxxxxxx	xxxxxxxx	xx1xx0xx	xxxxxxxx
x_3,x_4,x_5,x_6,x_7	xxxxx1xx	xxxxxxxx	xxxxxxxx	xxxxxxxx

また、7 階差分値は全ての bit が入力値によらない定数値となり、さらに 8 階差分は all zero となった。この結果は SBox が全単射関数であることに起因する。その様子を表 5、表 6 に示す。表中の x は、固定値の選び方により高階差分の値が変化する bit を表し、1 または 0 は固定値によらない値、すなわち定数となるということを示している。

3.4 F 関数の 32 階差分

F 関数の出力ビット全てについて,32 階差分を計算した。結果, 全てのビットについて鍵によって値の違う差分が出力されていることがわかった。

このことから F 関数は鍵によって変わるが,32 次とみなせると言える。

3.5 SQUARE 型攻撃評価

４種類のＳボックスは、全て全単射関数である。従って、F 関数の最右端１バイトのみを変数とし、256 通りの入力を考えれば、４回の T 関数通過後図 2 の $T(3)$ 出力４バイトは、各々256 通り全ての値をとり、８階差分が０である。しかし、５層目の出力においては、本文性質５より８階差分が０となるが、256 通りのパターンが出現している訳ではなく、これ以上の層を通過後の８階差分は確定できない。2.2 節の形式的代数次数解析より１回分多い T 関数通過後まで、８階差分攻撃が可能となるが、依然として、F 関数全体の高階差分特性で、多段の F 関数を攻撃できるものは、見つかっておらず、L 関数を無視したとしても 3 段の F 関数で、高階差分攻撃は適用できない。

4 補間攻撃 (Interpolation Attack)

4.1 自己評価書による評価

補間攻撃は, 暗号文のある n ビットを, 平文に関する $GF(q)$ 上の多項式で表現したとき, その多項式 (鍵の多項式) の, 未知の係数の数が N であれば,N 組の暗号文と平文から, 高い確率でこの多

表 4: SBox の 6 階差分値

変数	S_0	S_1	S_2	S_3
x_0,x_1,x_2,x_3,x_4,x_5	xxxxxxx1	11x1xxx0	0xx1xxxx	1xxxx011
x_0,x_1,x_2,x_3,x_4,x_6	xxxxxxxx	xxxxxxxx	01xxxxxx	011x1xxx
x_0,x_1,x_2,x_3,x_5,x_6	xxxx0xx1	xxxxxxxx	xxxxxxxx	11xx11xx
x_0,x_1,x_2,x_4,x_5,x_6	1xxx1xxx	xxx1xxx1	xxxxxxxx	1xxxx1xx
x_0,x_1,x_3,x_4,x_5,x_6	xxxx0xx1	x1xxxxxx	x01xxxxx	111xx1x1
x_0,x_2,x_3,x_4,x_5,x_6	0xxx0xx1	00x1xxxx	x111xxxx	xxxxxxxx
x_1,x_2,x_3,x_4,x_5,x_6	1xxx1xxx	10xxxxx0	xxxxxxxx	0x1x0010
x_0,x_1,x_2,x_3,x_4,x_7	xxxxx1x1	xx001xxx	1xxxxxx0	1xxxxxxx
x_0,x_1,x_2,x_3,x_5,x_7	xxxxxxxx	x011x1xx	xxxxxxxx	1xxxx1xx
x_0,x_1,x_2,x_4,x_5,x_7	xxxxx1x1	xx11x1x0	1xxxxxxx	1xxxx1xx
x_0,x_2,x_3,x_4,x_5,x_7	xxxxx1x1	101000xx	xxxxxxxx	xxx1xx10
x_1,x_2,x_3,x_4,x_5,x_7	xxxxx1xx	00xx1xx0	xxxxxxxx	0xx1x000
x_0,x_1,x_2,x_3,x_6,x_7	xx0xx1x0	xx01xxxx	xxxxx10x	10xx1xxx
x_0,x_1,x_2,x_4,x_6,x_7	x01xxx0x	xx10xxxx	1xxxx00x	1xxxxxxx
x_0,x_1,x_3,x_4,x_6,x_7	xxxxx000	xxxxxxxx	x1xx00xx	110xxxxx
x_0,x_2,x_3,x_4,x_6,x_7	x1xxx110	xx011xxx	x1xx0xx0	xxxxxxxx
x_1,x_2,x_3,x_4,x_6,x_7	x1xxx11x	xxxx1xxx	xxxxxxxx	1x0x1xxx
x_0,x_1,x_2,x_5,x_6,x_7	xx1x1xxx	xx11x1xx	xx0xx01x	xxxxxxxx
x_0,x_1,x_3,x_5,x_6,x_7	xxx011x0	x1xxx01x	xxxxxxxx	10xxx1xx
x_0,x_2,x_3,x_5,x_6,x_7	xxx111x1	x100x0xx	xxxxxxxx	xxxx1xxx
x_1,x_2,x_3,x_5,x_6,x_7	xxxx10xx	xxxxxxxx	xxxxxxxx	0xxx10xx
x_0,x_1,x_4,x_5,x_6,x_7	xxxx1x0x	xxxxx1xx	xxxxxxxx	0xxxx1xx
x_0,x_2,x_4,x_5,x_6,x_7	11xx1x1x	xx00x1xx	xx1xxxxx	xxxxxxxx
x_0,x_3,x_4,x_5,x_6,x_7	xxx11001	x0xxx1xx	x01x0xxx	xxxxxxxx
x_1,x_2,x_4,x_5,x_6,x_7	10xx1x0x	xxxxxxxx	xxxxxxxx	1xxxx1xx
x_0,x_3,x_4,x_5,x_6,x_7	xxxxxxxx	x0xxx1xx	xxxxxxxx	xx11xxx1
x_1,x_3,x_4,x_5,x_6,x_7	xxxxxxxx	xxxxxxxx	xxxxxxxx	1x00x1x0
x_2,x_3,x_4,x_5,x_6,x_7	00xx101x	01xx1xxx	xxxxxxxx	xx101x11

項式を復元できることを利用して解読するものである。この多項式が復元されれば, 任意の暗号文に対する平文が計算できるため,$N < q$ ならば意味のある攻撃となる。また,N は小さいほど攻撃は容易になる。ここで,q は素数または素数のべきとする。しかし, 暗号器全体に対して上記を保証することは困難であることから, 広い意味での補間攻撃に対する強度を保証することは困難であると思われる。過去の解読報告から考えると,CIPHERUNICORN-E で使用している 4 種の SBox を $GF(2^8)$ 上の多項式で表現したとき, その多項式の項数が十分に大きい値ならば, 暗号全体として補間攻撃に対して十分に安全であると期待できる。SBox を $GF(2^8)$ 上の多項式で表現したときの項数調査結果を表 7 に示す。SBox の生成に使用した既約多項式について,$GF(2^8)$ 上の多項式で表現した結果, 項の数は 252 以上であった。最大項数は 255 なので, 項の数 252 は十分大きい値であり, 補間攻撃に対して十分安全であると期待できる。

4.1.1 $GF(2^8)$ における Sbox の多項式表現

SBox を $GF(2^8)$ 上の多項式で表現した時の項数調査結果を表 8 に示す。項の数は 8 次の全ての原始多項式に対しても 252 以上であった。CIPHERUNICOEN-E の自己評価書の項数評価を裏付ける結果であり、補間攻撃に対する安全性が期待できる。

表 5: 換字テーブルの 7 階差分値

変数	S_0	S_1	S_2	S_3
x_0x_1,x_2,x_3,x_4,x_5,x_6	01110110	00101110	00001111	00010000
x_0x_1,x_2,x_3,x_4,x_5,x_7	11111010	00000010	01101110	01101000
x_0x_1,x_2,x_3,x_4,x_6,x_7	10011000	11000111	00110000	00010111
x_0x_1,x_2,x_3,x_5,x_6,x_7	11000010	10001001	11011001	00110011
x_0x_1,x_2,x_4,x_5,x_6,x_7	00010101	11001010	01011001	01111011
x_0x_1,x_3,x_4,x_5,x_6,x_7	11100000	10111001	10010011	00001010
x_0x_2,x_3,x_4,x_5,x_6,x_7	00100000	00000011	10000110	11000100
x_1x_2,x_3,x_4,x_5,x_6,x_7	00110001	00110110	11111111	01000000

表 6: 換字テーブルの 8 階差分値

変数	S_0	S_1	S_2	S_3
x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7	00000000	00000000	00000000	00000000

5 まとめ

CIPHERUNICORN-E について高階差分攻撃や補間攻撃の点から評価を行った。Sボックス等の構成部品の特性を基に、高階差分探索の努力は行ったが結果として, 2段でも適切な高階差分が発見できなかった。また、F関数に入る拡大鍵ビット数が 128 ビットである事、次数が３２次であり、その項数も、（６次までの探索に基づく予想ではあるが）十分多い事から１段消去型の攻撃も難しい。以上のように、3 段以上の段数においては高階差分攻撃及び補間攻撃の適用可能性は、見いだせなかった。

参考文献

[Nec] 日本電気, ”暗号技術仕様書 CIPHERUNICORN-E ／自己評価書 CIPHERUNICORN-E”, IPA 提出資料

表 7: SBox の $GF(2^8)$ 上の多項式項数

既約多項式	S_0	S_1	S_2	S_3
$0x11d$	$254^\star$	254	254	254
$0x14d$	254	254	$252^\star$	255
$0x165$	255	$254^\star$	254	253
$0x171$	255	253	255	$255^\star$

$\star$ それぞれの SBox を構成する時に使用した既約多項式。

表 8: S-Box の $GF(2^8)$ 上の多項式項数

原始多項式	項数			
	S_0	S_1	S_2	S_3
0x11d	254	254	254	254
0x12b	252	255	254	255
0x12d	255	253	252	253
0x14d	254	254	252	255
0x15f	255	255	253	255
0x163	252	254	254	255
0x165	255	254	254	253
0x169	253	255	254	254
0x171	255	253	255	255
0x187	254	255	253	255
0x18d	255	254	255	252
0x1a9	255	253	254	254
0x1c3	254	254	253	253
0x1cf	252	255	255	255
0x1e7	245	255	255	255
0x1f5	253	254	255	252

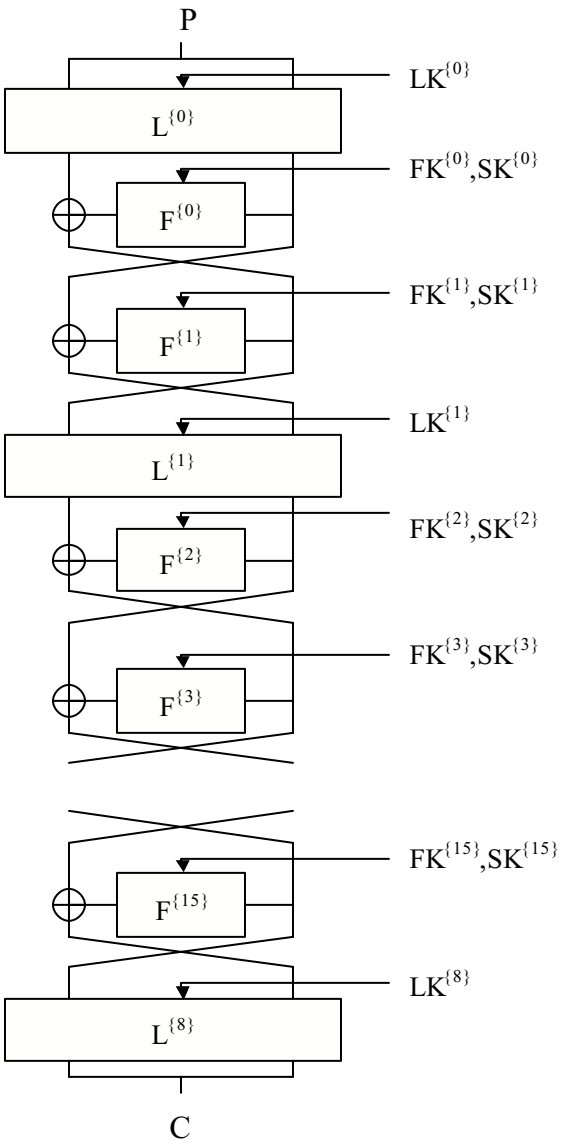

図 1: CIPHERUNICORN-E の構造

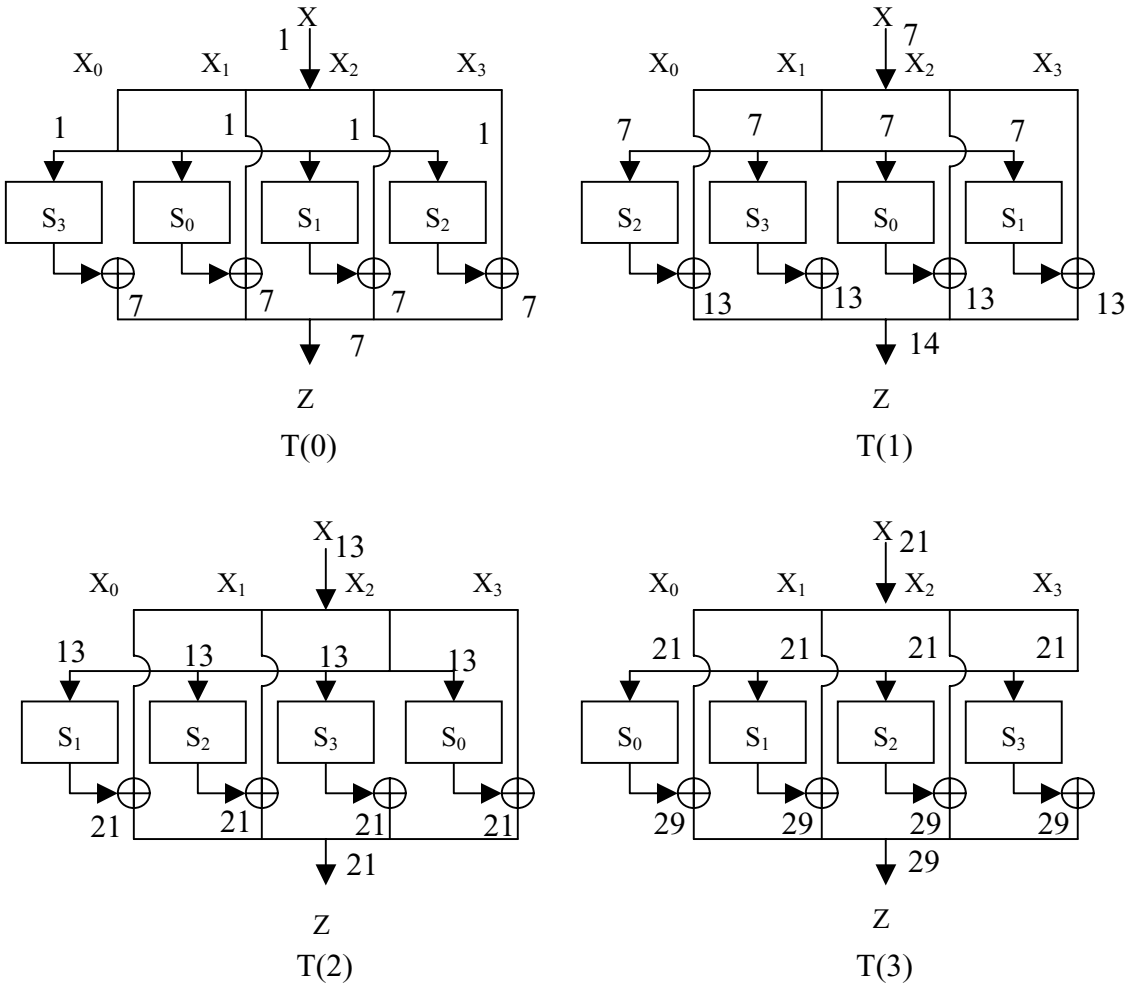

図 2: F 関数内の最初の 4 つの T 関数の形式的代数次数

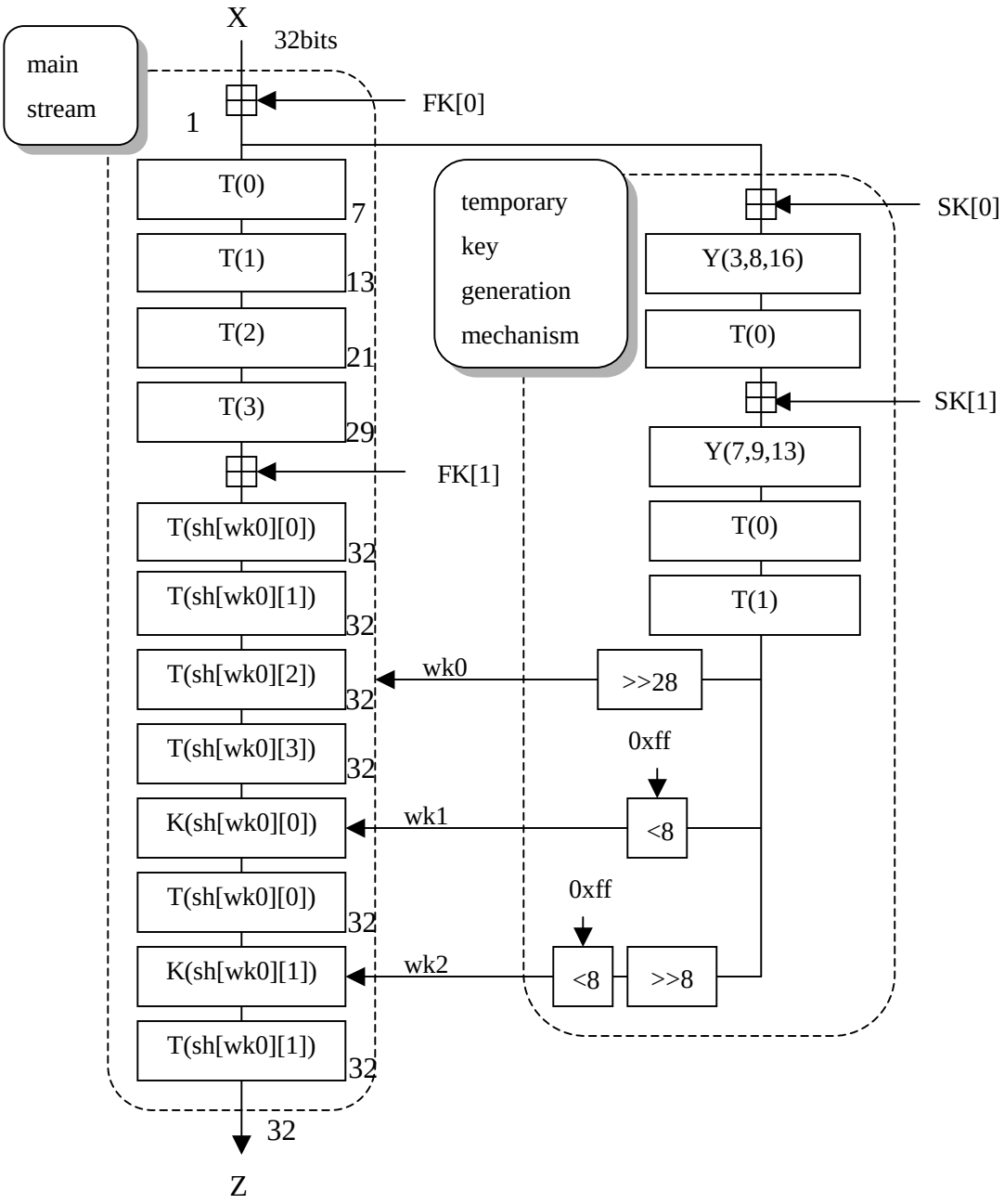

図 3: F 関数の形式的代数次数

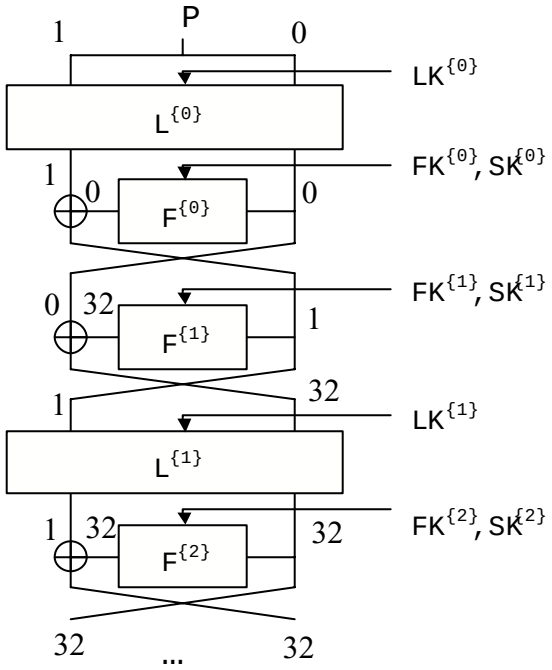

図 4: 形式的代数次数の解析