# CIPHERUNICORN-A に使われる関数の高階差分攻撃耐性評価

**概 要**

この文書は、電子政府において利用可能な暗号技術のリスト作成のための詳細評価の一環として、CIPHERUNICORN-A 暗号の高階差分攻撃耐性をその技術仕様書 [2]、及び自己評価書 [3] をもとに評価したものである。

## 1 CIPHERUNICOEN-A の構造

CIPHERUNICORN-A はデータブロック長 128bit、鍵長 128、192、256bit のいずれかを利用できる DES 型暗号で、AES 互換暗号として位置づけられる。CIPHERUNICORN-A データ撹拌部は DES 暗号などと同じ Feistel 構造を基本とし段数は 16 段構成である。初期/終期処理では、拡大鍵の加算/減算のみを行なう。また、鍵スケジューラは、与えられた秘密鍵から 72 個の拡大鍵を生成する。

全体構造の概略を図 1〜6 に示す。詳細は参考文献 [2] を参照頂きたい。

## 2 各関数の形式的な代数次数の評価

### 2.1 A3 関数

64bit の入力データを指定したシフト数だけ巡回シフトした後、3 つのデータの排他的論理和を求め出力する。

$$Y = (X <<< const_0) \oplus (X <<< const_1) \oplus (X <<< const_2)$$

$$const_0 = 0, const_1 = 23, const_2 = 41$$

巡回シフトと排他的論理和によって構成されるこの関数はでは次数の上昇はなく 1 次であるが、3 つのシフトの排他的論理輪であるため各出力 bit の項数が最大で 3 倍に増える可能性が考えられる。64bit の上位 32bit に対して下位 32bit の全ての bit が影響し、また下位の 32bit に上位 32bit の全ての bit が影響することや、シフト回数が byte 単位ではないことなどから、出力後の各 bit の項は (次数は上昇しないが)64bit 幅で効果的に攪拌されるということが言える。

### 2.2 定数乗算関数

32bit 入力データに対して定数を乗算し、出力結果とする。

$$Y = X \otimes const_n, n = 0, 1$$

$$const_0 = 0\mathrm{x}7e167289, const_1 = 0\mathrm{x}fe21464b$$

$const_0$ を 2 進数で表すと、

$$const_0 = 01111110000101100111001010001001$$

であり、下位 0〜3bit 目は次数が上昇がなく 1 次であるが、4bit 目は 2 次、5bit 目は 3 次、6bit 目は 4 次、7bit 目は 6 次、8bit 目以降は $n$bit 目で最大 $n$ 次と次数が上昇する。(表 2 上)

同様に $const_1$ を 2 進数で表すと

$$const_1 = 11111110001000010100011001001011$$

であり、下位 0〜1bit 目は次数が上昇がなく 1 次であるが、2bit 目は 2 次、3bit 目は 3 次、4bit 目は 4 次、5bit 目で 5 次、$n$bit 目で最大 $n$ 次と次数が上昇する。(表 1 下)

表 1: 定数乗算関数の代数次数

| $Y(const_0)$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 出力 bit | $y_{31}$ | ⋯ | $y_n$ | ⋯ | $y_8$ | $y_7$ | $y_6$ | $y_5$ | $y_4$ | $y_3$ | $y_2$ | $y_1$ | $y_0$ |
| 次数 | 31 | ⋯ | n | ⋯ | 8 | 6 | 4 | 3 | 2 | 1 | 1 | 1 | 1 |

| $Y(const_1)$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 出力 bit | $y_{31}$ | ⋯ | $y_n$ | ⋯ | $y_8$ | $y_7$ | $y_6$ | $y_5$ | $y_4$ | $y_3$ | $y_2$ | $y_1$ | $y_0$ |
| 次数 | 31 | ⋯ | n | ⋯ | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 1 |

### 2.3 Tn 関数 [図 1]

32bit 入出力。入力を 1byte ごとに区切り、入力番号に対応する 1byte を換字テーブルの入力値とする。換字テーブルは 8bit 入力、8bit 出力であり、CIPHERUNICORN-E で使用した 4 種類を使用する。

$$Y = S_0(X_n) \parallel S_1(X_n) \parallel S_2(X_n) \parallel S_3(X_n)$$

Tn 関数の次数は Tn 関数の内部で使用される換字テーブルの次数より 7 次である。
(換字テーブルに関する調査は 3 節参照)

### 2.4 F 関数 [図 2]

F 関数は、本流と 1 次鍵生成部から構成される。本流では入力データ 64bit と拡大鍵を加算し、A3 関数、定数乗算、$T_n$ 関数を実行する。1 次鍵生成部では、拡大鍵と入力データ 64bit を加算し、乗算定数、Tn 関数を通過後、1 次鍵を取り出す。1 次鍵を本流のデータ攪拌に使用し、出力データ 64bit を生成する。本流、1 次鍵生成部、両方において Tn 関数の 3 段の通過により形式的次数は bit 幅より大きくなるため、64 次となる。

### 2.5 MT 関数 [図 3]

入力データ 64bit の上位 32bit ($X_0$) と定数 const との乗算後、$T_0$ 関数の入力とする。$T_0$ 関数の出力を入力データの下位 32bit($X_1$) と排他的論理和し、出力とする。

$$
\begin{aligned}
Y &= Y_0 \parallel Y_1 \\
Y_0 &= (X_0 \otimes const) \\
Y_1 &= X_1 \oplus T_0(Y_0) \\
const &= \text{0x01010101} \\
&= 00000001000000010000000100000001(binary)
\end{aligned}
$$

出力データ Y の上位 32bit の次数は定数 $const$ を乗算するため、定数乗算後の出力 $Y_0$ は、下位 0〜8bit 目までは 1 次であり、9bit 目が 2 次、10bit 目が 3 次、11bit 目が 4 次、12bit 目が 6 次、13bit 目が 8 次、14bit 目が 10 次、$\cdots$ 、と次数が上昇する。下位 32bit$Y_1$ は定数乗算後 $T_0$ 関数を通過するため $Y_0$ に T 関数の 7 次を乗算した分の次数が上昇することになるとなる。各 bit の次数を表 2 に示す。

表 2: MT 関数出力の代数次数

| $Y_0$ | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 出力 bit | $y_{15}$ | $y_{14}$ | $y_{13}$ | $y_{12}$ | $y_{11}$ | $y_{10}$ | $y_9$ | $y_8$ | $y_7$ | $y_6$ | $y_5$ | $y_4$ | $y_3$ | $y_2$ | $y_1$ | $y_0$ |
| 次数 | 12 | 10 | 8 | 6 | 4 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 出力 bit | $y_{31}$ | | | $\cdots$ | | | $y_n$ | | | | $\cdots$ | | | $y_{18}$ | $y_{17}$ | $y_{16}$ |
| 次数 | 31 | | | $\cdots$ | | | n | | | | $\cdots$ | | | 18 | 16 | 14 |

| $Y_1$ | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 出力 bit | $y_{15}$ | $y_{14}$ | $y_{13}$ | $y_{12}$ | $y_{11}$ | $y_{10}$ | $y_9$ | $y_8$ | $y_7$ | $y_6$ | $y_5$ | $y_4$ | $y_3$ | $y_2$ | $y_1$ | $y_0$ |
| 次数 | 32 | 32 | 32 | 32 | 28 | 21 | 14 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 出力 bit | $y_{31}$ | | | $\cdots$ | | | $y_n$ | | | | $\cdots$ | | | $y_{18}$ | $y_{17}$ | $y_{16}$ |
| 次数 | 32 | | | $\cdots$ | | | 32 | | | | $\cdots$ | | | 32 | 32 | 32 |

### 2.6　各関数の形式的な代数次数の評価のまとめ

F 関数は、A3 関数で各 byte にまたがって攪拌された後、T 関数 (換字テーブル) を 9 回〜10 回通過するため 1 段でも 64 次に近い十分大きな次数と推定され、16 段で構成される CIPHERUNICORN-A は十分な代数的強度が期待される。

## 3　換字テーブルについての諸評価

CIPHERUNICOEN-A で使用される換字テーブルは 8bit 入力、8bit 出力であり、CIPHERUNICORN-E で使用した 4 種類と同じものが使用される。この換字テーブル ($S_0$〜$S_7$) について諸評価を行なった。

### 3.1　最大差分確率 $DP$

換字テーブルの最大差分確率は、$S_0$〜$S_3$ のいずれについても $2^{-6}$ であった。また、任意の入力差分 $\Delta x_0$ に対しても最大差分確率は $2^{-6}$ であり、逆に任意の出力差分 $\Delta y_0$ に対しても最大差分確率は $2^{-6}$ であった。このことは、最大差分確率を与える入力差分と出力差分の組み合わせに偏りがなく、差分攻撃に対する耐性が高いということを示している。

### 3.2　最大線形確率 $LP$

換字テーブルの最大線形確率は、$S_0$〜$S_3$ のいずれについても $2^{-6}$ であった。差分確率の場合と同様に、任意の入力マスク $\Gamma x_0$ に対しての最大差分確率も $2^{-6}$ であり、逆に任意の出力マスク $\Gamma y_0$ に対しても最大差分確率は $2^{-6}$ であった。このことは、最大線形確率を与える入力マスクと出力マスクの組み合わせに偏りがなく、線形攻撃に対する耐性が高いということを示している。

### 3.3　代数次数及び項数

換字テーブル $S_3$〜$S_0$ の入出力 8bit について、出力の各 bit に対するブール多項式を求め、その代数次数及び項数を求めた。その結果、代数次数は換字テーブルの全 4 種類の全ての bit について 7 次であり。その項数は、108〜141 項であり、平均的な項数が現れており、代数的攻撃に対しても高い耐性が期待される。表 3 にその結果を示す。

表 3: 換字テーブルのブール多項式の代数次数と項数

| $S_0$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 出力 bit | $S_{0_7}$ | $S_{0_6}$ | $S_{0_5}$ | $S_{0_4}$ | $S_{0_3}$ | $S_{0_2}$ | $S_{0_1}$ | $S_{0_0}$ |
| 次数 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 項数 | 116 | 126 | 127 | 108 | 123 | 121 | 126 | 136 |

| $S_0$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 出力 bit | $S_{1_7}$ | $S_{1_6}$ | $S_{1_5}$ | $S_{1_4}$ | $S_{1_3}$ | $S_{1_2}$ | $S_{1_1}$ | $S_{1_0}$ |
| 次数 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 項数 | 119 | 131 | 128 | 138 | 131 | 114 | 132 | 118 |

| $S_2$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 出力 bit | $S_{2_7}$ | $S_{2_6}$ | $S_{2_5}$ | $S_{2_4}$ | $S_{2_3}$ | $S_{2_2}$ | $S_{2_1}$ | $S_{2_0}$ |
| 次数 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 項数 | 124 | 114 | 139 | 118 | 121 | 138 | 141 | 125 |

| $S_3$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 出力 bit | $S_{3_7}$ | $S_{3_6}$ | $S_{3_5}$ | $S_{3_4}$ | $S_{3_3}$ | $S_{3_2}$ | $S_{3_1}$ | $S_{3_0}$ |
| 次数 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 項数 | 125 | 118 | 138 | 134 | 118 | 129 | 121 | 143 |

### 3.4 高階差分特性

$S_3$〜$S_0$ の換字テーブルについて高階差分値を求めた。$S_3$〜$S_0$ の 4 つの全ての換字テーブルについて、１〜8 階差分値を計算し、常に定数となる出力ビットを探索した。その結果、5 階差分、6 階差分の一部について、あるビットが定数となった。その結果を表 4、表 5 に示す。各行の左端に示されたビットを変数とする高階差分を計算したとき、表中 x は変化する出力ビット、0 または 1 は定数となる出力ビットを表す。、7 階差分値は全ての bit が入力値によらない定数値であり、さらに 8 階差分は all zero となる。これは、代数次数が 7 である事を表している。結果を表 6、表 7 に示す。

表 4: 換字テーブルの 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 |

### 3.5 補間攻撃

補間攻撃は、暗号文のある n ビットブロックを $GF(2^n)$ の元と見なし、平文に関する $GF(2^n)$ 上の多項式で表現したとき、その多項式 (鍵を係数とする多項式) の、未知係数の数が N であれば、N 組の平文・暗号文から、高い確率でこの多項式で復元でき,N+1 組あれば、その復元が正しいか否か高い確率で判別できることを利用して解読するものである。

換字テーブルを $GF(2^8)$ 上の多項式で表現した時の項数を調べた結果を表 8 に示す。既約多項式について、$GF(2^8)$ 上の多項式で表現した結果、項の数はどの原始多項式に対しても 252 以上であった。CIPHERUNICOEN-A の自己評価書 [3] では、最大項数は 255 としているが、252 は十分に大きい値であり、補間攻撃に対して安全であると期待できる。

### 3.6 換字テーブルについての諸評価のまとめ

換字テーブルの最大差分確率、最大線形確率は共に単射の 8bit 入出力関数としては最良の値 $2^{-6}$ であり、代数次数も 7 次と、8bit 入出力の関数としては最良であり、項数も 108 以上であり、この観点からは、最良の 8bit 入出力換字テーブルであると言える。

# 4　F 関数に対する高階差分

F 関数は 64 ビット入出力であるが、計算量の関係で、について 32 階差分、31 階差分特性の調査を行なった。

## 4.1　31 階差分

F 関数の入力 $X_l$(32bit) を all zero とし、$X_r$(32bit) 側で 全ての変数のとり方 (bit oriented で 32 通り) に対して 31 階差分を求めた。はじめに鍵スケジューラには all zero の鍵を入力し、１段目 F 関数出力の３１階差分を求めた。鍵を変えて同じく３１階差分を求めたが、固定値となるものは、見いだせなかった。

## 4.2　32 階差分

同様に F 関数の入力 $X_l$(32bit) を all zero とし、$X_r$(32bit) 側で変数を取り 32 階差分を求めた。結果として、特徴は見いだせなかった。

## 4.3　F 関数に対する高階差分のまとめ

31 階差分、32 階差分により得られた値は、byte 単位でみても、$\neq 0$ でいずれも異なる値であり、法則性を見つけるにはいたらなかった。F 関数のはじめの A3 関数により byte 単位でも攪拌されており、64bit の入出力である F 関数に単純に 32 階差分を適用しても解読に有利な情報は得られないと言える。

# 5　まとめ

CIPHERUNICORN-A で使用される各関数についての強度評価の結果、各関数に特に目立った弱点は発見できなかった。特に F 関数は、データ攪拌部に一時鍵を足しこむ形になっている上、F 関数自身が換字テーブルを使った 6 段 (一時鍵生成部)、10 段 (データ攪拌部) の Feistel 構造になっており、非常に強い関数であると考える。

今回は F 関数内部や鍵スケジューラで使われる関数を個別に調査するにとどまったが、アルゴリズム全体を評価するにあたり、F 関数全体とそれを数段組み合わせたもの、アルゴリズム全体についてなど、もう少し詳しい調査が必要である。また、CIPHERUNICORN-A の難点といわれている、実装した際の暗号化（復号）の速度についても調査が必要であろう。

# 参考文献

[1] Yukiyasu Tsunoo,Hiroyasu Kubo,Hiroshi Miyauchi,Katsuhiro Nakamura 「A New 128-bit Block Cipher CIPHERUNICORN-A」*SCIS2000-A18*, 2000.

[2] 日本電気株式会社「暗号技術仕様書 CIPHERUNICORN-A」*IPA* 提出資料, 2000.

[3] 日本電気株式会社「自己評価書 CIPHERUNICORN-A」*IPA* 提出資料, 2000.

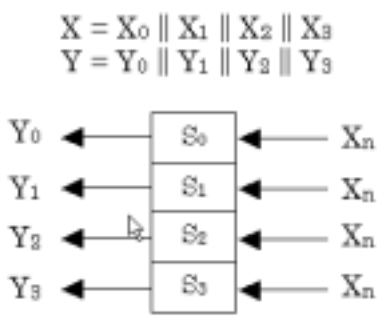


図 1: $T_n$ 関数

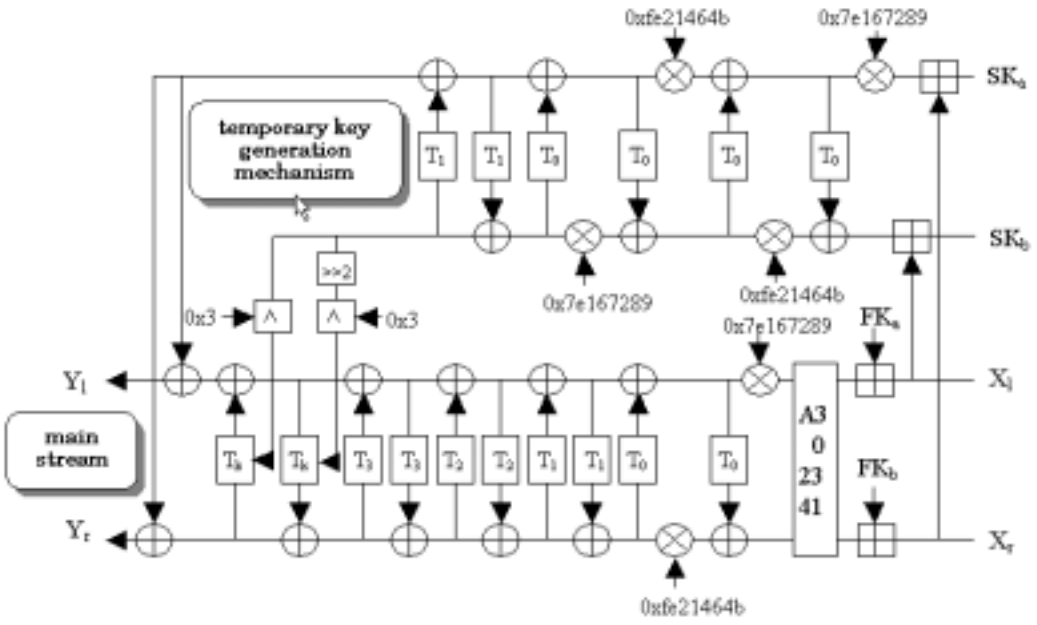


図 2: F 関数

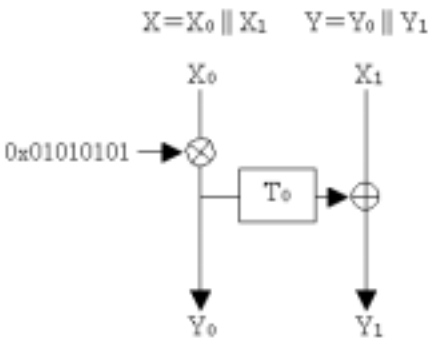


図 3: MT 関数

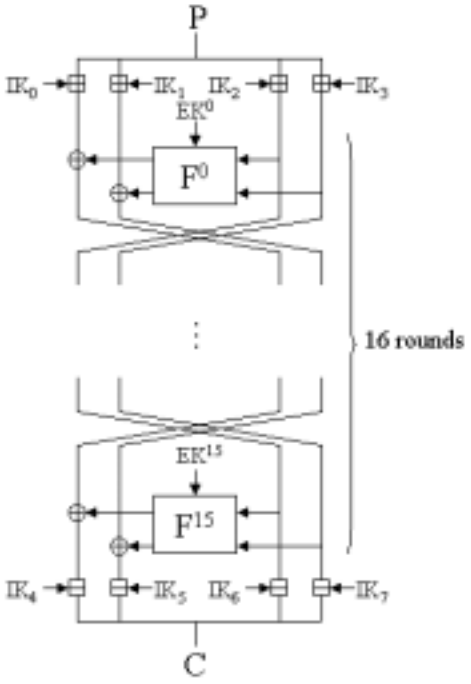


図 4: データ攪拌部 (暗号化)

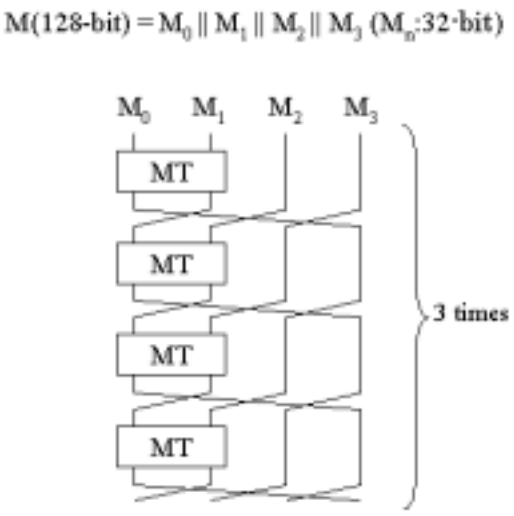


図 5: 鍵スケジューラダミーループ (秘密鍵 128bit)

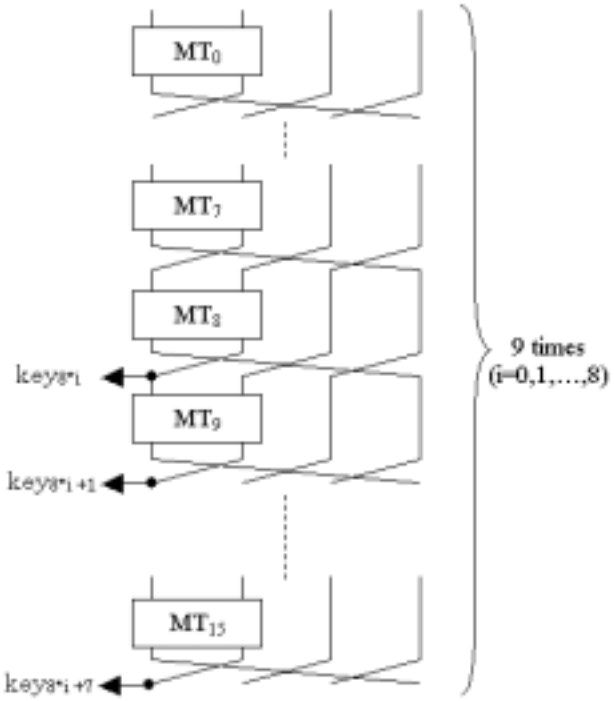


図 6: 鍵スケジューラ拡大鍵取り出し (秘密鍵 128bit)

表 5: 換字テーブルの 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 |

表 6: 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 7: 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 |

表 8: 換字テーブルの $GF(2^8)$ 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 |