付録Ｃ．FEAL－NX

１．　はじめに

これは, 電子政府事業に用いる暗号方式に関する応募資料（FEAL-NX 仕様書及び FEAL-NX 自己評価書[FEAL]）に基づき，高階差分攻撃及び補間攻撃の観点から詳細評価するために作成したものである。アルゴリズムに関する詳細は省略したので、仕様書を参照して頂きたい。

２．　ＦＥＡＬ－ＮＸのアルゴリズム

FEAL（the Fast Date Encipherment Algorithm）は、NTT によって開発された 64 ビット共通鍵暗号方式で、3 つのオプション（鍵の長さ、回転数、鍵パリティ）を持つ。鍵の長さは、64 ビットか、128 ビットであり、回転数(N)は、データランダム化のための内部回転数（即ち段数、以下段数と呼ぶ）を指定し、鍵パリティオプションは、鍵ブロックにおけるパリティビットを使うか、使わないかを選択する。暗号募集要件に照らし合わせ、ここでは FEAL 暗号の鍵パリティビットなし 128 ビット鍵 N 段版を FEAL-NX と呼ぶ。

FEAL－NX のデータランダム化部の構造を図 1 に示す。基本構造は、N 段（32 以上の偶数）Feistel 型である。64 ビット平文入力 $P(64)$は、前処理として 4 個（64 ビット）の拡大鍵$(K_N, K_{N+1}, K_{N+2}, K_{N+3})$が加算される。これをゲートブランチと呼ぶ。その後、左右半分づつ$L_0(32)$、$R_0(32)$に分割され、段関数Ｆによる Feistel 型の処理をＮ段繰り返し、後処理として、ゲートブランチが適用され暗号文 $C(64)$となる。

F 関数Ｙ＝Ｆ（Ｘ；K）を図 2 に示す。32 ビット入出力で、拡大鍵 Kは 16 ビットであり、図１の Ki(I=1,..N)として供給される。Ｆ関数は排他的論理和⊕と S 関数によって構成される。

S 関数は、8 ビット 2 入力 8 ビット出力であり、S_0と S_1の２種類がある。それは、算術加算とローテンションで構成され２つの８ビット入力x、yに対し８ビット出力zを出力する次式の関数である。

$$z = S_0(x, y) = Rot2((x + y) \bmod 256) \qquad (1)$$

$$z = S_1(x, y) = Rot2((x + y + 1) \bmod 256) \qquad (2)$$

３．　S 関数の評価

Ｓ関数の入出力x、y、zの第ｎビットを、それぞれx_n、y_n、z_n（ＬＳＢは第０ビット）とするならば、単位に表現するならば、次式である。

$$z_{(n+2) \bmod 8} = x_n \oplus y_n \oplus d_n \qquad (3)$$

ここで、d_nは算術和のキャリを表し、次式である

$$d_{n+1} = x_n y_n \oplus d_n (x_n \oplus y_n) \qquad n = 0,1,...,6$$

$$d_0 = \begin{cases} 0 & S_0\text{関数} \\ 1 & S_1\text{関数} \end{cases} \qquad (4)$$

上式に基づき S_0及び S_1のブール代数展開式を求めれば、次数、項数は表１、表２である。両関数共に最大次数８次であるが、出力ビット位置によるばらつきが大きい事がわかる。

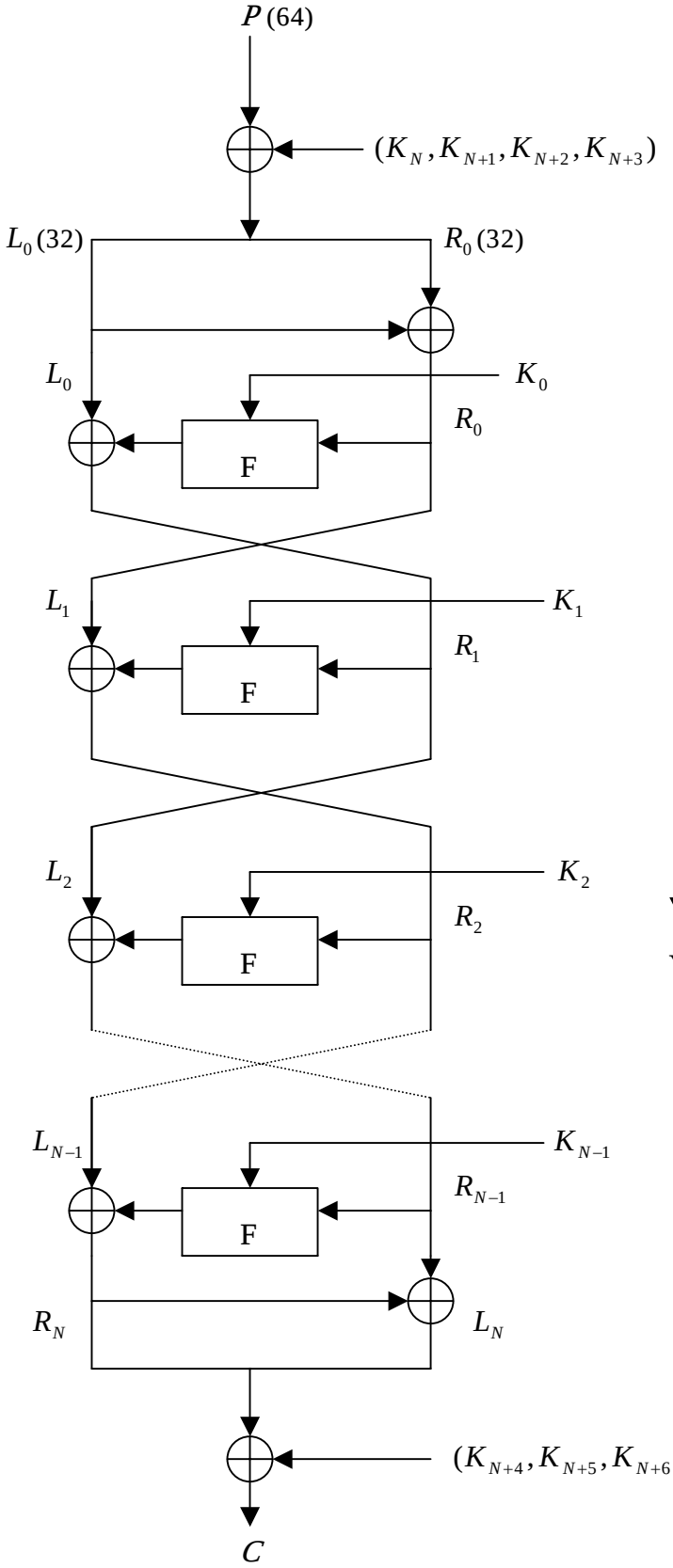

図 1　データランダム化部

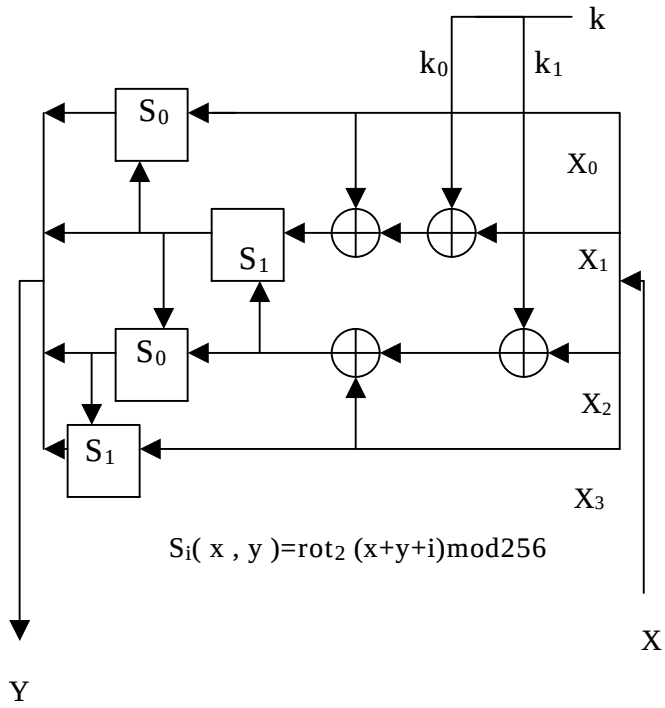

図 2　F 関数

S_0： $z = rot_2 (x+y) \bmod 256$

（表 1）

	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0
1 次項	2	2	2	2	2	2	2	2
2 次項	1	1	1	1	1	0	1	1
3 次項	2	2	2	2	0	0	2	2
4 次項	4	4	4	0	0	0	4	4
5 次項	8	8	0	0	0	0	8	8
6 次項	16	0	0	0	0	0	16	16
7 次項	0	0	0	0	0	0	32	32
8 次項	0	0	0	0	0	0	64	0
全項数	33	17	9	5	3	2	129	65

S_1： $z = rot_2 (x+y+1) \bmod 256$

（表 2）

	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0
1 次項	2	2	2	2	4	3	2	2
2 次項	1	1	1	5	1	0	1	1
3 次項	2	2	10	2	0	0	2	2
4 次項	4	20	4	0	0	0	4	4
5 次項	40	8	0	0	0	0	8	8
6 次項	16	0	0	0	0	0	16	80
7 次項	0	0	0	0	0	0	160	32
8 次項	0	0	0	0	0	0	64	0
全項数	65	33	17	9	5	3	255	129

４． F 関数の評価

４．１ F 関数の形式的代数次数評価

F 関数の入出力をバイト単位に区切り入力を(X_0,X_1,X_2,X_3)、出力を(Y_0,Y_1,Y_2,Y_3)とする。各バイトはＳ関数で処理され出力に出てくる。Ｓ関数のブール展開式（３）（４）を使い、Ｆ関数出力の次数を求めれば、各出力ビットに対し、

$$Y_0=(12,11,10, 9, 8, 7,14,13)$$
$$Y_1=(6, 5, 4, 3, 2, 1, 8, 7) \qquad (5)$$
$$Y_2=(12,11,10, 9, 8, 7,14,13)$$
$$Y_3 =(18,17,16,15,14,13,20,19)$$

となる。これより、Ｆ関数全体としての次数は、20 次である。しかし、ビット毎の次数のばらつきが大きく、１段消去型攻撃の場合、最終段鍵の総当たりよりも効果的な解読法が存在するであろう事が判る。

５． 高階差分攻撃及び補間攻撃の評価

５．１ 形式的代数次数評価

平文 $P=(L_0,R_0)$で、R_0 を fix、L_0 を変数と見て形式的解析をすれば、代数次数の上昇は、図 5 となる。なお、

ゲートブランチ処理は線形であり、平文及び鍵の等価変形を考えれば、以下の議論には影響を与えない。

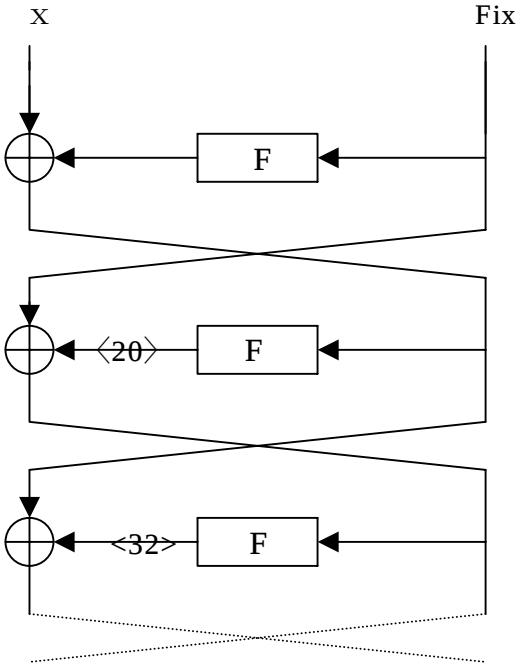

図 5

形式的には、32 階以下の差分で０になることが保証されるのは、3 段目Ｆ関数入力までである。

５．２　1－8 高階差分特性の評価

Ｓ関数は、バイト区切りで２つの変数を入力に持つ。平文をバイト区切りで考え $P = (L_0,R_0) = (X_0,X_1,X_2,X_3,X_4,X_5,X_6,X_7)$とし、左半分４バイト X_0～X_3 の１バイトに着目し、各バイト内の選択ビットを変数に取る形で、８階以下で最も、深い段数まで、出力ビットの高階差分が 0 となる入力変数及び出力ビットの取り方を探した。結果を表 3～6 に示す。表中“ 0 ”は、高階差分値が、平文の fix 値又は鍵によらず 0。“ － ”は、これに依存する値である。ランダムに０，１が発生する可能性を考慮し、複数回の実験を行い、０にならない可能性が、2^{-64} を目安に判断した。この表は、例えば X_1 を変数とした時（表４）で、３段目出力Y_1の２ビット目が５階差分で０となっている。X_1内のある５ビットを選択して、５階差分を取れば、この出力ビットに関し０である事を示している。これを使い、１段消去型攻撃を考えれば、5 階差分で 5 段まで解くことができる。

５．３　31 及び 32 階高階差分特性の評価

平文 $P=(L_0,R_0)=(X_0,X_1,X_2,X_3,X_4,X_5,X_6,X_7)$で、$L_0$ を変数にとり 31 及び 32 階差分の高階差分値を表 7～8 に示す。また、31 階差分の 4 段目の８ビット出力 Y_1 の０は４回確認した。Y_1 がランダムな値であれば、０にならない確率は 2^{-32} である。まだ、理論的解析は終了していないが、31 階差分で４段目の Y_1 が常に０であるならば、この 8 ビットに注目し、１段消去攻撃で、6 段まで解くことができる。

変数X_0:(表3)

	8階差分値					7階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	000000-0	00000000	00000000	00000000	2	000000-0	00000000	00000000	00000000
3	--------	-----0--	--------	--------	3	--------	-----0--	--------	--------
4	--------	--------	--------	--------	4	--------	--------	--------	--------
	6階差分値					5階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	000000-0	00000000	00000000	00000000	2	000000--	00000000	00000000	00000000
3	--------	--------	--------	--------	3	--------	--------	--------	--------
	4階差分値					3階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	000000--	00000000	00000000	--0000--	2	000000--	00000000	00000000	----00--
3	--------	--------	--------	--------	3	--------	--------	--------	--------
	2階差分値					1階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	---000--	00000000	-00000--	--------	2	----00--	----0000	--0000--	--------
3	--------	--------	--------	--------	3	--------	--------	--------	--------

変数X_1:(表4)

	8階差分値					7階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	00000000	00000000	2	00000000	00000000	00000000	00000000
3	--------	-----0--	--------	--------	3	--------	-----0--	--------	--------
4	--------	--------	--------	--------	4	--------	--------	--------	--------
	6階差分値					5階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	00000000	00000000	2	00000000	00000000	00000000	00000000
3	--------	-----0--	--------	--------	3	--------	-----0--	--------	--------
4	--------	--------	--------	--------	4	--------	--------	--------	--------
	4階差分値					3階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	00000000	--0000--	2	00000000	00000000	00000000	----00--
3	--------	--------	--------	--------	3	--------	--------	--------	--------
	2階差分値					1階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	-00000--	00000000	-00000--	----00--	2	--0000--	----0000	--0000--	--------
3	--------	--------	--------	--------	3	--------	--------	--------	--------

変数X_2:(表5)

	8階差分値					7階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	000000-0	-----0--	2	00000000	00000000	000000-0	-----0--
3	--------	--------	--------	--------	3	--------	--------	--------	--------
	6階差分値					5階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	000000-0	-----0--	2	00000000	00000000	000000--	--------
3	--------	--------	--------	--------	3	--------	--------	--------	--------
	4階差分値					3階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	000000--	--------	2	00000000	00000000	000000--	--------
3	--------	--------	--------	--------	3	--------	--------	--------	--------
	2階差分値					1階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3
2	-00000--	00000000	---000--	--------	2	--0000--	----0000	----00--	--------
3	--------	--------	--------	--------	3	--------	--------	--------	--------

変数X_3:(表6)

	8階差分値					7階差分値			
出力段目	Y0	Y1	Y2	Y3	出力段目	Y0	Y1	Y2	Y3

(表7)

	31階差分値			
出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000
4	--------	00000000	--------	--------
5	--------	--------	--------	--------

(注) 4段目の差分値Y1は確率$1-2^{-32}$

(表8)

	32階差分値			
出力段目	Y0	Y1	Y2	Y3
2	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000
4	--------	00000000	--------	--------
5	--------	--------	--------	--------

５．４　ＳＱＵＡＲＥ型攻撃評価

FEAL の F 関数において、全単射性は、F 関数本体、及び、F 関数入力４バイト中の１バイト以上のバイト単位入力選択において発生する。従って、そのようなバイト単位の入力選択のもとで、8,16,24,32 階差分について最も深い段まで高階差分値が０となるものを考察したところ、32 階差分（F 関数の全単射性）を利用する場合が効果的であり、4 段目Ｆ関数入力まで 32 階差分は 0 である。

５．５　線形和攻撃耐性の評価

GF（2^8）上の多項式（現在のところ法多項式 0x163 のみ）を利用した線形和攻撃の強度評価を行った。評価手法は以下の通りである。

① 平文を $p=(x_1,x_2,\cdots,x_i,\cdots)$　$x_i(1\le i\le 8)$ と byte 毎に分割し、任意の 1byte を変数とする。他は 0 に固定する。

② その時得られる n 段目の暗号文 $c=(y_1,y_2,\cdots,y_i,\cdots)$ $y_i(1\le i\le 8)$ の y_i を GF（2^8）上の多項式として表す。

③ これを予め用意しておいた、258 個（128bit）のランダムなマスター鍵に対して行う。

④ 得られた 258 本の GF（2^8）上の多項式について、未知係数個数を見積もる。

⑤ ①～④を全ての x_i , y_i に対し計算し最小未知係数個数を見積もる。

この結果、

1 段数目	2 段数目	3 段数目	4 段数目	5 段数目
1	117	256	256	256

となる。２段目まで、未知係数個数＜256 であり、１段消去型攻撃を考えるならば４段まで攻撃可能である。

５．６　高階差分攻撃耐性の評価

５．１から５．５節の結果をもとに、５．３節の 31 階差分による高階差分攻撃が最適であると考える。N段 FEAL-NX 暗号を攻撃する場合、N段目の後ろにゲートブランチによる後処理が加わっていると考えて評価する。この場合、等価変換を行うならば、N 段目及び N-1 段目には拡大鍵 32 ビットづつ、N-2 段目以前には、段毎に

16 ビットの拡大鍵が挿入される。

１段消去型攻撃では、４段目 Y_1 の 31 階差分＝０の性質を使い、６段目拡大鍵を仮定し、暗号文から遡ることで、拡大鍵（の一部）が推定できる。この攻撃方程式に関係するのは、6 段目 F 関数の１つのＳ関数（図２の上から２番目のＳ関数）の出力であり、関係する鍵は、その入力の 16 ビットとなる。S 関数の最上位ビット入力に係わる鍵ビットは、キャリとして影響する出力ビットを持たず、31 階差分値を計算する時の偶数回（2^{31} 回）の加算でキャンセルされ、攻撃方程式の成否に影響しない。それより関係する鍵は 14 ビットである。5.1 節で言及したように、FEAL-NX のＦ関数では、この鍵 14 ビットの総当たりでは無く、より計算量の少ない攻撃法が存在する。攻撃方程式は 8 ビットの式であるが、これをビット単位の８本の式と考え、S 関数の出力 3 ビット目に相当する位置から順次解いていけばよい。この場合、新たに推定すべき鍵は、方程式毎に 2 ビットの鍵となる。本文３．２節＜鍵の総当たり攻撃＞を、２ビットの鍵に対し行い、これを７回繰り返せば 14 ビットの鍵が求まる事になる。S 関数１回の計算は、F 関数 1/4 回の計算量と見積もれるので、F 関数の計算量を 1 とすると、その計算量は本文(3.8)式より

$$T_{6段} = \frac{7*2^{31+2+1}}{4} \cong 2^{35} \qquad (6)$$

となる。必要平文数は

$$M_{6段} = 2*2^{31} = 2^{32} \qquad (7)$$

となる。２段消去型攻撃を考え、7 段目拡大鍵 32 ビットを総当たりして 6 段攻撃を適用すれば、計算量 $T_{7段}$及び必要平文数$M_{7段}$は

$$T_{7段} = \frac{7*2^{31+34+1}}{4} \cong 2^{67}、\ M_{7段}=34*2^{31} \cong 2^{37} \qquad (8)$$

である。同様に、８段では、計算量及び必要平文数は、

$$T_{8段} = \frac{7*2^{31+2+32+32+1}}{4} \cong 2^{98}、\ M_{8段}=66*2^{31} \cong 2^{38} \qquad (9)$$

さらに、段数を増やす場合、拡大鍵を鍵総当たりする F 関数が 3 段以上に付いては、新たに、段当たり 16 ビット拡大鍵が追加されるので、計算量、必要平文数は

$$T_{9段} = \frac{7*2^{31+2+16+32+32+1}}{4} \cong 2^{114}、\ M_{9段}=82*2^{31} \cong 2^{38} \qquad (10)$$

となる。この方法で、鍵の総当たりより、計算量が多くなる段数は 10 段目であり、9 段目まで攻撃可能であると言える。

6. 結論

FEAL-NX に対する詳細評価として,形式的代数次数解析を含む高階差分解読法及び補間解読法を行った。結果として、有効な 31 階差分を見いだし、これを使うことにより、2^{38}組の平文を用いて、９段 FEAL-NX は 2^{114}回のＦ関数計算量で解ける。これは、４段消去型攻撃である。このように多段の消去が可能となるのは、FEAL-NX の段関数に入る鍵が 16 ビットであることも影響している。以上の考察は、確率１で成立する攻撃方

程式を使うものであり、この 4 段消去攻撃が、そのまま、差分解読や線形解読のような確率的解読法に適用できるわけでは無いが、FEAL-NX は差分攻撃に対し弱い事が知られており、提案暗号の 32 段の安全性には注意深い検討が必要であろう。

[FEAL] NTT,"暗号技術応募書／FEAL-NX 仕様書／FEAL-NX 自己評価書",IPA 提出資料