詳細評価報告書
FEAL-NX 攻撃評価

平成１３年1月１０日

1 概要

1.1 評価結果概要

差分解読により、選択入力差分とそれに対応する暗号文が 2^{64} 個（データ量は 2^{63}）を入手できるという条件のもとで、秘密鍵の総当たり探索よりも少ない計算量で、鍵を求めることができるだろう。

鍵処理の構造から、拡大鍵が 6 段毎に繰り返す秘密鍵が、2^{32} 程度見つかった。ただし、この特性により強度の低下は認められない。

上記 2 点は、実際の運用において、問題はないと考える。

1.2 定義

本報告書では、特に注意書きのない限り FEAL-NX の仕様書([FEAL-NX_s])の記述に従い、以下とする。

(A, B, ……)	：この順序の連結(concatenation)
A⊕B	： ブロック A と B のビット対応の排他的論理和
x･y	：ビット x とビット y の論理積
ビット位置	：ブロックの最左端ビット(MSB)から右方向に 1,2,3,…,と数える
鍵	：暗号化／復号に用いる鍵情報(128 ビット)
Kr	：鍵処理部により生成される拡大鍵

2 既存の解読法に関して

本節は、おもに、既に知られている解読法に対する FEAL-NX の強度に関しての解析報告である。

2.1 差分解読

差分解読に関しては、暗号研究者による評価がなされている。これらは主に FEAL-N([M90])に対する解読法であるが、FEAL-N と FEAL-NX の違いは鍵処理の部分であるため、FEAL-NX に対しても同様と考えて良いであろう。

表 1 差分解読の主な結果

	差分解読法とその応用
4 段	10000([B88])
8 段	10000([GC90])
31 段	2^{63}([BS91])

*: 表中の値は解読に必要な平文暗号文数

FEAL-N と FEAL-NX の違いは鍵長であり、また 1 段あたりの鍵のビット数が 32 と比較的小さいため、32 段目のすべての拡大鍵を仮定し 31 段の差分解読を行うことで鍵の全探索よりも小さな計算量で鍵を求める方式が存在する可能性が考えられる。この方針からデータ量 2^{63}、計算量 2^{99} 以下の解読方式を導いた。31 段の差分解読は E.Biham らによる結果[BS91]を利用している。

以下、本方式のステップを示す。

(1) 解読ステップ

(A) 準備

FEAL-NX の初期鍵、終期鍵を移動し、各関数 f に 32 ビットの鍵が入るような等価変形(付録 B)を行う。

(B) 32 段目の鍵 AK_{31} を仮定する

AK_{31} としては 2^{32} 通りの可能性がある。各 AK_{31} について、以降のステップを行う。本方式の計算量は、概算で、31 段差分解読の 2^{32} 倍程度になると考えられる。

(C) 31 段の FEAL-NX に対して差分解読を行う

AK_{31} を既知であると仮定することにより、FEAL-NX を 31 段とみなすことができる。解読に用いるデータについて、(B)で仮定した AK_{31} を用いて最終段を復号し、31 段の暗号文を計算する。

[BS91]による 31 段に対する差分解読を行う。

AK_{31} の仮定が正しい場合は、[BS91]によると 31 段の差分解読が成功し、AK_{30} は 8 通りが候補として残る。また、AK_{31} の仮定が誤っている場合は、AK_{30} としてどのような鍵が残るか、または、残らないかは不明である。解読が成功したよ

うに見える場合のみ、(AK_{31}, AK_{30})の組を鍵候補として残しておく。

AK_{31} をすべて試すと、候補として残る(AK_{31},AK_{30})の組は、高々2^{35}組である。これらについては、以降のステップでそれぞれ試行する必要がある。

31 段の差分解読で必要なデータ量は[BS91]により、2^{63}である。

(D) 30 段の FEAL-NX に対して差分解読を行う

(AK_{31}, AK_{30})を既知であると仮定すると、30 段とみなすことができる。最大2^{35}通りの(AK_{31}, AK_{30})に対して、30 段の差分解読を行い、AK_{29} を求める。

AK_{29} と AK_{31} は、最左端(MSB)8 ビットと最右端(LSB)8 ビットが同じであるため、一組の(AK_{31}, AK_{30})あたりに残る AK_{29} の候補は 2 通りである。ここで候補として残る(AK_{31}, AK_{30}, AK_{29})の組は最大で、2^{36}組となる。

(E) 29 段の FEAL-NX に対して差分解読を行う

(AK_{31}, AK_{30}, AK_{29})を既知であるとし、最大2^{36}通りの(AK_{31}, AK_{30}, AK_{29})に対して、29 段の差分解読を行う。求める鍵は AK_{28} である。

AK_{28} と AK_{30} は、最左端(MSB)8 ビットと最右端(LSB)8 ビットが同じであるため、一組の(AK_{31}, AK_{30}, AK_{29})あたりに残る AK_{28} の候補は 2 通りである。ここで候補として残る(AK_{31}, AK_{30}, AK_{29}, AK_{28})の組は最大で、2^{37}組となる。

(F) 28 段,…,23 段の FEAL-NX に対して差分解読を行う。

(D)(E)と同様に、28 段から 23 段まで差分解読を続ける。1 段毎に残る AK_r 候補は 2 通りなので、23 段差分解読まで行うと、(AK_{31}, AK_{30}, AK_{29}, AK_{28}, AK_{27}, AK_{26}, AK_{25}, AK_{24}, AK_{23}, AK_{22})の組が 2^{43}通り候補として残る。

(G) 鍵を求める

(F)までで求めた AK 候補を用いると、拡大鍵の情報を以下の形式で得ることができる。

$$K_{31} \oplus K_{29} , K_{30} \oplus K_{28} , K_{29} \oplus K_{27} , K_{28} \oplus K_{26}$$
$$K_{27} \oplus K_{25} , K_{26} \oplus K_{24} , K_{25} \oplus K_{23} , K_{24} \oplus K_{22}$$

これらに含まれる拡大鍵のうち 2 つ(たとえば、K_{31} と K_{30})をそれぞれ 2^{16}通り試行すれば、鍵生成部の構造から 128 ビットの鍵を求めることができる。この時に残る 128 ビット鍵候補は、最大で 2^{75}である。

(H) 鍵を絞る

候補として残った鍵 2^{75} は 128 ビットの鍵空間に比べて小さい。したがって、得られた暗号文対から対応する入力差分となるかを検証し、本物の鍵を求めればよい。

(2) 計算量

本方式の総計算量は、表 2における各ステップの計算量(試行する鍵量×計算量)の合計となるので、最大でも 2^{99}以下となり、鍵総当たり 2^{128}よりも小さい。

表 2 各ステップにおける試行する鍵量と計算量

各ステップ	各ステップにおける試行する鍵あたりの計算量	試行する鍵量
(C)AK_{30} を求める	計：2^{65} 2^{64} 対の入力差分に対する暗号文を 1 段復号：2^{65} counting schemeで残ったpairについてactual subkeyを計算：≪ 2^{65} 2 つの actual subkey について、AK_{30} を計算：$2\times2^{7}\times2$	AK_{31} ：2^{32}
(D)AK_{29} を求める	計：2^{62} 2^{61} 対の入力差分に対する暗号文をさらに 1 段復号：2^{62} counting scheme で残った pair について actual subkey を計算：≪ 2^{65}	(AK_{31},AK_{30})：2^{35}
(E)AK_{28} を求める	計：2^{60} 2^{59} 対の入力差分に対する暗号文をさらに 1 段復号：2^{60} counting scheme で残った pair について actual subkey を計算：≪ 2^{65}	(AK_{31},AK_{30},AK_{29}) ：2^{36}
(F)AK_{27}~AK_{22} を求める	AK_{27}：2^{58} AK_{26}：2^{56} AK_{25}：2^{54} AK_{24}：2^{52} AK_{23}：2^{50} AK_{22}：2^{48}	(AK_{31},⋯,AK_{28}) ： 2^{37} (AK_{31},⋯,AK_{27}) ： 2^{38} (AK_{31},⋯,AK_{26}) ： 2^{39} (AK_{31},⋯,AK_{25}) ： 2^{40} (AK_{31},⋯,AK_{24}) ： 2^{41} (AK_{31},⋯,AK_{23}) ： 2^{42}
(G)鍵を求める	計：2^{17} K_{31} を試行し K_{29},K_{27},K_{25},K_{23} を計算：2^{16} K_{30} を試行し K_{28},K_{26},K_{24},K_{22} を計算：2^{16}	(AK_{31},⋯,AK_{22}) ： 2^{43}
(H)鍵を絞る	鍵候補について、復号し検証する	(K_{31},⋯,K_{22}) ：2^{75}

2.2 線形解読

線形解読に関しては、暗号研究者による評価がなされている。これらは主に FEAL-N([M90])に対する解読法であるが、FEAL-N と FEAL-NX の違いは鍵処理の部分であるため、FEAL-NX に対しても同様と考えて良いであろう。

表 3 線形解読の主な結果

	線形解読とその応用
4 段	39([KK93])
8 段	2^{15}([MY92]),2^{23}([KR94]),12([AO95])

*: 表中の値は解読に必要な平文暗号文数

また、線形解読やその応用である解読法に関しては、表 3以外の結果は得られなかった。

2.3 高階差分解読

2.3.1 関数S

関数 S に関して、各出力ビットのブール多項式次数と、各次数の項数を表 4に示す。また付録 Bに各出力ビットの詳細なブール多項式を示す。

関数 S は GF(2^8)上の加算と同じブール式であり、出力ビットによって、最大次数や項数に大きな偏りがある。また、項の種類にも特徴がある。

表 4 関数 S 出力ビットブール多項式次数の項数

出力ビット / 次数 / 番号	1	2	3	4	5	6	7	8
1	2	2	2	2	2	2	2	2
2	1	1	1	1	1	0	1	1
3	2	2	2	2	0	0	2	2
4	4	4	4	0	0	0	4	4
5	8	8	0	0	0	0	8	8
6	16	0	0	0	0	0	16	16
7	0	0	0	0	0	0	32	32
8	0	0	0	0	0	0	64	0

2.3.2 関数f

出力ビット 9 から 16 は、関数 S を 1 度しか通っていないため、最大次数は順に、6,5,4,3,2,1,8,7 となる。最左バイトが次に次数が低いバイトであるが、出力ビット 6 の次数 7 が最小であり、他のビットはそれ以上であることが、容易に判断できる。

関数 f においては、最大次数が 1 という出力ビットが存在し、また、項の種類に偏りがある事も想像できる。しかし、関数 f を数段通過すれば、徐々に項の種類も多様になることが考えられ、FEAL-NX に対し、高階差分解読を適用することは困難であろう。

2.4 補間攻撃

関数 f は拡大鍵の排他的論理和、データの排他的論理和(単純な線形変換層)、2 種類の関数 S から構成されている。

FEAL-NX に補間攻撃を試行する場合、鍵の挿入や線形変換層が排他的論理和で構成されているため、素体 GF(2)を基礎体とする拡大体上で関数 S を表現することを考える。しかし、関数 S は modulo 256 の演算と巡回シフトの演算で構成されているため、拡大体上で簡明な式により表現することは難しいと思われる。

関数 f は、GF(2)上の加算と巡回シフト、modulo 256 の演算が混在しており、FEAL-NX に対して補間攻撃を試行することは困難である。

2.5 mod n 解読

2.5.1 関数S

関数 S の 2 ビット左巡回シフトで不変となるような数で mod を取ることが有効である考えられる。実際に関数 S では以下の式が成立する。

S0(x,y) mod 3 ≡ ((x+y) mod 256) mod 3
S1(x,y) mod 3 ≡ ((x+y+1) mod 256) mod 3

2.5.2 関数f

関数 S には上記の性質があるが、関数 f における鍵の挿入や、関数 f 出力を他方のデータと混ぜる処理で採用している排他的論理和演算を通過すると、上記の性質で発生する統計的偏りは失われてしまう。従って、上記の性質を利用しても多段でその統計的偏りを検出することは困難であろう。

2.6 鍵関連攻撃

鍵鍵関連攻撃の対象は簡易な鍵処理部を持つ暗号である。鍵 K と別の鍵 K*の間に大きな関係があり、かつそれぞれの鍵での暗号過程のデータを等しくすることができる平文/暗号文を見つけられれば攻撃は成功する。

FEAL-NX の場合、鍵処理が第3.1節で述べるような特徴をもっている。この特徴が現れる場合、鍵間に何らかの関連があれば鍵関連攻撃は適用可能であるが、鍵間に関連は見られない。よって FEAL-NX の鍵処理の構造では鍵関連攻撃は適用不可能だろう。

2.7 スライド攻撃

スライド攻撃の対象は同じ構造のラウンド関数の繰り返しによる暗号で、簡易な鍵処理部によって何段かスライドした拡大鍵が元の拡大鍵と同じ値になるような場合である。また、偶数段の拡大鍵が同じで奇数段の拡大鍵が同じ Feistel 暗号にも適用可能である。適用可能な条件として、スライドした分のラウンド関数に挿入される拡大鍵量が鍵長よりも短いことが挙げられる。

FEAL-NX の場合、鍵処理が第3.1節で述べるような特徴をもっている。この特徴が常に現れるならばスライド攻撃が適用可能であるが、出現頻度が 2^{128} の鍵空間に対して 2^{32} 程度しかない。よって FEAL-NX の鍵処理の構造ではスライド攻撃は適用不可能だろう。

2.8 中間一致攻撃

中間一致攻撃の対象は鍵を単純に 2 等分して得られる鍵をそれぞれの拡大鍵(各段に入力する鍵)として使用する場合である。

FEAL-NX の場合、鍵処理で生成される拡大鍵は鍵を単純に 2 等分したものではなく、単純な鍵の仮定によって平文/暗号文から中間値をみることは出来ない。よって FEAL-NX の鍵処理の構造では中間一致攻撃は適用不可能だろう。

2.9 タイミング攻撃、電力差分攻撃、故障利用攻撃

これらの解読法は、暗号アルゴリズムを実装した際に問題となるものであり、基本的には、どのようなアルゴリズムであっても、攻撃の対象となりうる。実装時に注意すべきである。

3 鍵処理の特性

3.1 拡大鍵が6段毎に繰り返す場合

FEAL-NX の鍵処理には次のような特性がある。

ある鍵の場合、鍵処理 3 段ごとに同じ値を出力する場合がある。この場合、関数 f に挿入される拡大鍵(16bit)は 6 段毎に同じ鍵値となる。鍵の最左端 32 ビットがすべて 0 の場合に発生し、個数は 2^{32} 程度と考えられる。

表 5は、このような鍵の例である。

表 5 拡大鍵が 6 段毎に繰り返す例

鍵	拡大鍵
00000000 39899a4b 9e2fc7b4 58a65d00	0000 0000 0000 0000 3989 9a4b … 以下繰り返し
00000000 86db0ea6 0a587a59 73837400	0000 0000 0000 0000 86db 0ea6 … 以下繰り返し

*: 表内は全て 16 進表記

また、鍵処理部 6 段ごと、つまり、拡大鍵 12 段ごとに、同じ値を繰り返す場合も存在する。繰り返す拡大鍵のパタンは、0, a, b, 0, c, d（ただし、a,b,c,d はある 32 ビット値）となる。

同様に、鍵処理部 3n 段(n=1,2,3,4,…)ごとに拡大鍵が繰り返す場合がある。

3.2 単射性

鍵処理の構造から、鍵処理から出力される拡大鍵(32 ビット)を任意に 5 つ集めると、鍵(128 ビット)を一意に求めることができる。従って、鍵生成は単射構造であり、異なる鍵から生成される拡大鍵に同じものは存在しない。

4 まとめ

差分解読により、選択入力差分とそれに対応する暗号文が 2^{64} 個（データ量は 2^{63}）を入手できるという条件のもとで、秘密鍵の総当たり探索よりも少ない計算量(最大でも 2^{99})で、鍵を求めることができるだろう。(第2.1節)

鍵処理の構造から、拡大鍵が 6 段毎に繰り返す秘密鍵が、2^{32} 程度見つかった。ただし、この特性によるなんらかの解読法は見つからなかった。(第3.1節)

上記 2 点は、いずれも机上の計算であり、実際の運用では、問題とはならないと考えられる。

我々が、解析した範囲では、FEAL-NX には、致命的な問題は見つからなかった。

また、FEAL-N に対して、既に発表されている論文[付録 C]からも、FEAL-NX の安全性に問題となる記述は見つからなかった。

付録A 参考文献

[FEAL-NX_s] http://info.isl.ntt.co.jp/feal-nx/index-j.html 暗号技術仕様書[和文] および正誤表.

[FEAL-NX_e] http://info.isl.ntt.co.jp/feal-nx/index-j.html 自己評価書[和文].

[A99] 青木和麻呂、"不能差分利用攻撃について"、SCIS'99、1999 (In Japanese).

[AAO95]S.Araki, K.Aoki, K.Ohta, "The Best Linear Expression Search of FEAL," SSCIS95, 1995 (In Japanese).

[AKM97] Kazumaro Aoki, Kunio Kobayashi, Shiho Moriai, "Best Differential Characteristic Search of FEAL," FES'97, 1997.

[AO95]K.Aoki, K.Ohta, "Differential-Linear Cryptanalysis of FEAL-8," SCIS95, 1995 (In Japanese).

[AOAM94] K.Aoki, K.Ohta, S.Araki, M.Matsui, "Linear Cryptanalysis of FEAL-8 (Experimentation Report)," ISEC94-6, 1994.

[B88] Bert den Boer, "Cryptanalysis of F.E.A.L.," EUROCRYPT'88, 1988.

[B94] Eli Biham, "On Matsui's Linear Cryptanalysis," EUROCRYPT'94, 1994.

[BBS99] Eli Biham, Alex Biryukov, Adi Shamir, “Miss in the Middle Attack on IDEA and Khufu," FSE'99, 1999.

[BS90] Eli Biham, Adi Shamir, "Differential Cryptanalysis of DES-like Cryptosystems (extended abstract),” CRYPTO'90, 1990.

[BS91] Eli Biham, Adi Shamir, "Differential Cryptanalysis of Feal and N-Hash," EUROCRYPT'91, 1991.

[FA87] Walter Fumy, Siemens AG, “On the F-function of FEAL,” CRYPTO’87, 1987.

[GC90] Henri Gilbert, Guy Chasse, “A STATISTICAL ATTACK OF THE FEAL-8 CRYPTOSYSTEM,” CRYPTO’90, 1990.

[K91] T. Kaneko, "A known plaintext cryptanalysis attack on FEAL-4," ISEC91-25, 1991.

[KK93] M. Kurita, T. Kaneko, "A known plaintext attack of FEAL-4," SCIS93-3B, 1993.

[KR94] B.S.Kaliski, M.J.B. Robshaw, "Linear Cryptanalysis Using Multiple Approximations and FEAL," FSE94, 1994.

[M90] S.Miyaguchi, “The FEAL Cipher Family,” CRYPTO’90, 1990.

[MK94] 増田 孝志, 金子 敏信, “FEAL 暗号方式の解読における松井法と差分方程式の関係 On the Relation of Matui's Method and Differential Equation Method,” SCIS94, 1994 (In Japanese).

[MY92] M. Matsui, A. Yamagishi, "A New Method for Known Plaintext Attack of FEAL Cipher," EUROCRYPT'92, 1992.

[OA94-1] K.Ohta, K.Aoki, "Linear Cryptanalysis of the Fast Data Encipherment Algorithm," ISEC94-5, 1994.

[OA94-2] K.Ohta, K.Aoki, "Linear Cryptanalysis of the Fast Data Encipherment Algorithm," CRYPTO'94, 1994.

[SM87] Akihiro Shimizu, Shoji Miyaguchi, "Fast Data Encipherment Algorithm FEAL," EUROCRYPT'87, 1987.

[TG91] A. Tardy-Corfdir and H. Gilbert, "A Known Plaintext of FEAL-4 and FEAL-6," CRYPTO'91, 1991.

[TOD93] Y. Tsunoo, E. Okamoto, H. Doi, "Analytical Known Plain-text Attack for FEAL-4 and Its Implovement," SCIS93-3A, 1993.

[TOU94] Y. Tsunoo, E. Okamoto, T. Uematsu, "Ciphertext Only Attack for One-way function of the MAP using One Ciphertext," CRYPTO'94, 1994.

付録B 初期鍵、終期鍵を移動した FEAL-NX

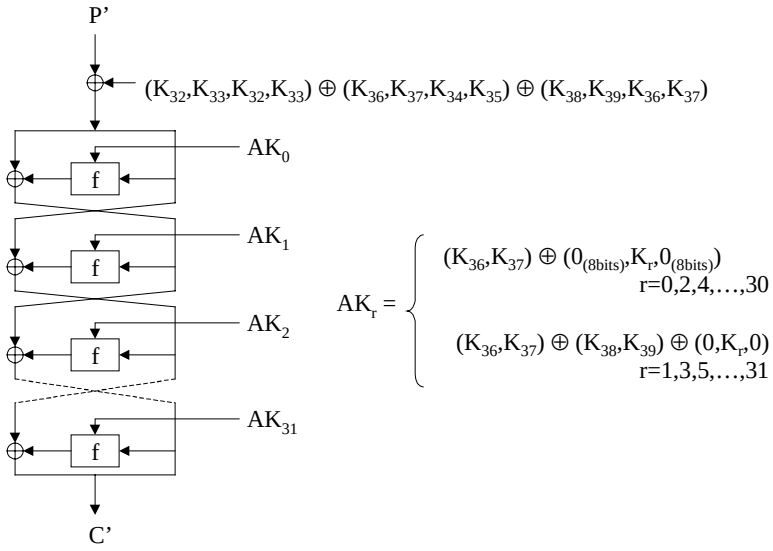

図 1 等価変形したデータランダム(暗号化アルゴリズム)

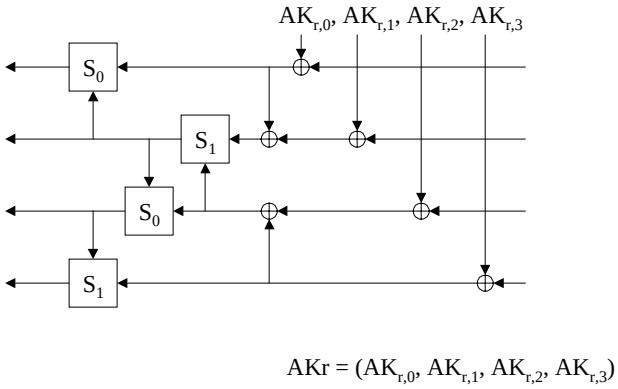

図 2 等価変形した関数 f

付録C 関数 S のブール多項式表現

付録C.1 S0

x1,x2,⋯,x8 および y1,y2,⋯,y8 は関数 S の入力ビットを表し、z1,z2,⋯,z8 は S0 の出力ビットを表す。x1,y1,z1 を、それぞれ最左端ビット(MSB)とする。

z1 = x3 ⊕ y3 ⊕ x4･y4 ⊕ x4･x5･y5 ⊕ y4･x5･y5 ⊕ x4･x5･x6･y6 ⊕ x4･y4･x6･y6 ⊕ y4･x5･x6･y6 ⊕ y4･y5･x6･y6
⊕ x4･x5･x6･x7･y7 ⊕ x4･x5･y6･x7･y7 ⊕ x4･y5･x6･x7･y7 ⊕ x4･y5･y6･x7･y7
⊕ y4･x5･x6･x7･y7 ⊕ y4･x5･y6･x7･y7 ⊕ y4･y5･x6･x7･y7 ⊕ y4･y5･y6･x7･y7
⊕ x4･x5･x6･x7･x8･y8 ⊕ x4･x5･x6･y7･x8･y8 ⊕ x4･x5･y6･x7･x8･y8 ⊕ x4･x5･y6･y7･x8･y8
⊕ x4･y5･x6･x7･x8･y8 ⊕ x4･y5･x6･y7･x8･y8 ⊕ x4･y5･y6･x7･x8･y8 ⊕ x4･y5･y6･y7･x8･y8
⊕ y4･x5･x6･x7･x8･y8 ⊕ y4･x5･x6･y7･x8･y8 ⊕ y4･x5･y6･x7･x8･y8 ⊕ y4･x5･y6･y7･x8･y8
⊕ y4･y5･x6･x7･x8･y8 ⊕ y4･y5･x6･y7･x8･y8 ⊕ y4･y5･y6･x7･x8･y8 ⊕ y4･y5･y6･y7･x8･y8
z2 = x4 ⊕ y4 ⊕ x5･y5 ⊕ x5･x6･y6 ⊕ y5･x6･y6 ⊕ x5･x6･x7･y7 ⊕ x5･y6･x7･y7 ⊕ y5･x6･x7･y7 ⊕ y5･y6･x7･y7
⊕ x5･x6･x7･x8･y8 ⊕ x5･x6･y7･x8･y8 ⊕ x5･y6･x7･x8･y8 ⊕ x5･y6･y7･x8･y8
⊕ y5･x6･x7･x8･y8 ⊕ y5･x6･y7･x8･y8 ⊕ y5･y6･x7･x8･y8 ⊕ y5･y6･y7･x8･y8
z3 = x5 ⊕ y5 ⊕ x6･y6 ⊕ x6･x7･y7 ⊕ y6･x7･y7 ⊕ x6･x7･x8･y8 ⊕ x6･y7･x8･y8 ⊕ y6･x7･x8･y8 ⊕ y6･y7･x8･y8
z4 = x6 ⊕ y6 ⊕ x7･y7 ⊕ x7･x8･y8 ⊕ y7･x8･y8
z5 = x7 ⊕ y7 ⊕ x8･y8
z6 = x8 ⊕ y8
z7 = x1 ⊕ y1 ⊕ x2･y2 ⊕ x2･x3･y3 ⊕ y2･x3･y3 ⊕ x2･x3･x4･y4 ⊕ x2･y3･x4･y4 ⊕ y2･x3･x4･y4 ⊕ y2･y3･x4･y4
⊕ x2･x3･x4･x5･y5 ⊕ x2･x3･y4･x5･y5 ⊕ x2･y3･x4･x5･y5 ⊕ x2･y3･y4･x5･y5
⊕ y2･x3･x4･x5･y5 ⊕ y2･x3･y4･x5･y5 ⊕ y2･y3･x4･x5･y5 ⊕ y2･y3･y4･x5･y5
⊕ x2･x3･x4･x5･x6･y6 ⊕ x2･x3･x4･y5･x6･y6 ⊕ x2･x3･y4･x5･x6･y6 ⊕ x2･x3･y4･y5･x6･y6
⊕ x2･y3･x4･x5･x6･y6 ⊕ x2･y3･x4･y5･x6･y6 ⊕ x2･y3･y4･x5･x6･y6 ⊕ x2･y3･y4･y5･x6･y6
⊕ y2･x3･x4･x5･x6･y6 ⊕ y2･x3･x4･y5･x6･y6 ⊕ y2･x3･y4･x5･x6･y6 ⊕ y2･x3･y4･y5･x6･y6
⊕ y2･y3･x4･x5･x6･y6 ⊕ y2･y3･x4･y5･x6･y6 ⊕ y2･y3･y4･x5･x6･y6 ⊕ y2･y3･y4･y5･x6･y6
⊕ x2･x3･x4･x5･x6･x7･y7 ⊕ x2･x3･x4･x5･y6･x7･y7 ⊕ x2･x3･x4･y5･x6･x7･y7 ⊕ x2･x3･x4･y5･y6･x7･y7
⊕ x2･x3･y4･x5･x6･x7･y7 ⊕ x2･x3･y4･x5･y6･x7･y7 ⊕ x2･x3･y4･y5･x6･x7･y7 ⊕ x2･x3･y4･y5･y6･x7･y7
⊕ x2･y3･x4･x5･x6･x7･y7 ⊕ x2･y3･x4･x5･y6･x7･y7 ⊕ x2･y3･x4･y5･x6･x7･y7 ⊕ x2･y3･x4･y5･y6･x7･y7
⊕ x2･y3･y4･x5･x6･x7･y7 ⊕ x2･y3･y4･x5･y6･x7･y7 ⊕ x2･y3･y4･y5･x6･x7･y7 ⊕ x2･y3･y4･y5･y6･x7･y7
⊕ y2･x3･x4･x5･x6･x7･y7 ⊕ y2･x3･x4･x5･y6･x7･y7 ⊕ y2･x3･x4･y5･x6･x7･y7 ⊕ y2･x3･x4･y5･y6･x7･y7
⊕ y2･x3･y4･x5･x6･x7･y7 ⊕ y2･x3･y4･x5･y6･x7･y7 ⊕ y2･x3･y4･y5･x6･x7･y7 ⊕ y2･x3･y4･y5･y6･x7･y7
⊕ y2･y3･x4･x5･x6･x7･y7 ⊕ y2･y3･x4･x5･y6･x7･y7 ⊕ y2･y3･x4･y5･x6･x7･y7 ⊕ y2･y3･x4･y5･y6･x7･y7
⊕ y2･y3･y4･x5･x6･x7･y7 ⊕ y2･y3･y4･x5･y6･x7･y7 ⊕ y2･y3･y4･y5･x6･x7･y7 ⊕ y2･y3･y4･y5･y6･x7･y7
⊕ x2･x3･x4･x5･x6･x7･x8･y8 ⊕ x2･x3･x4･x5･x6･y7･x8･y8 ⊕ x2･x3･x4･x5･y6･x7･x8･y8
⊕ x2･x3･x4･x5･y6･y7･x8･y8 ⊕ x2･x3･x4･y5･x6･x7･x8･y8 ⊕ x2･x3･x4･y5･x6･y7･x8･y8
⊕ x2･x3･x4･y5･y6･x7･x8･y8 ⊕ x2･x3･x4･y5･y6･y7･x8･y8 ⊕ x2･x3･y4･x5･x6･x7･x8･y8
⊕ x2･x3･y4･x5･x6･y7･x8･y8 ⊕ x2･x3･y4･x5･y6･x7･x8･y8 ⊕ x2･x3･y4･x5･y6･y7･x8･y8
⊕ x2･x3･y4･y5･x6･x7･x8･y8 ⊕ x2･x3･y4･y5･x6･y7･x8･y8 ⊕ x2･x3･y4･y5･y6･x7･x8･y8
⊕ x2･x3･y4･y5･y6･y7･x8･y8 ⊕ x2･y3･x4･x5･x6･x7･x8･y8 ⊕ x2･y3･x4･x5･x6･y7･x8･y8
⊕ x2･y3･x4･x5･y6･x7･x8･y8 ⊕ x2･y3･x4･x5･y6･y7･x8･y8 ⊕ x2･y3･x4･y5･x6･x7･x8･y8
⊕ x2･y3･x4･y5･x6･y7･x8･y8 ⊕ x2･y3･x4･y5･y6･x7･x8･y8 ⊕ x2･y3･x4･y5･y6･y7･x8･y8
⊕ x2･y3･y4･x5･x6･x7･x8･y8 ⊕ x2･y3･y4･x5･x6･y7･x8･y8 ⊕ x2･y3･y4･x5･y6･x7･x8･y8
⊕ x2･y3･y4･x5･y6･y7･x8･y8 ⊕ x2･y3･y4･y5･x6･x7･x8･y8 ⊕ x2･y3･y4･y5･x6･y7･x8･y8
⊕ x2･y3･y4･y5･y6･x7･x8･y8 ⊕ x2･y3･y4･y5･y6･y7･x8･y8 ⊕ y2･x3･x4･x5･x6･x7･x8･y8
⊕ y2･x3･x4･x5･x6･y7･x8･y8 ⊕ y2･x3･x4･x5･y6･x7･x8･y8 ⊕ y2･x3･x4･x5･y6･y7･x8･y8
⊕ y2･x3･x4･y5･x6･x7･x8･y8 ⊕ y2･x3･x4･y5･x6･y7･x8･y8 ⊕ y2･x3･x4･y5･y6･x7･x8･y8
⊕ y2･x3･x4･y5･y6･y7･x8･y8 ⊕ y2･x3･y4･x5･x6･x7･x8･y8 ⊕ y2･x3･y4･x5･x6･y7･x8･y8
⊕ y2･x3･y4･x5･y6･x7･x8･y8 ⊕ y2･x3･y4･x5･y6･y7･x8･y8 ⊕ y2･x3･y4･y5･x6･x7･x8･y8
⊕ y2･x3･y4･y5･x6･y7･x8･y8 ⊕ y2･x3･y4･y5･y6･x7･x8･y8 ⊕ y2･x3･y4･y5･y6･y7･x8･y8
⊕ y2･y3･x4･x5･x6･x7･x8･y8 ⊕ y2･y3･x4･x5･x6･y7･x8･y8 ⊕ y2･y3･x4･x5･y6･x7･x8･y8
⊕ y2･y3･x4･x5･y6･y7･x8･y8 ⊕ y2･y3･x4･y5･x6･x7･x8･y8 ⊕ y2･y3･x4･y5･x6･y7･x8･y8
⊕ y2･y3･x4･y5･y6･x7･x8･y8 ⊕ y2･y3･x4･y5･y6･y7･x8･y8 ⊕ y2･y3･y4･x5･x6･x7･x8･y8
⊕ y2･y3･y4･x5･x6･y7･x8･y8 ⊕ y2･y3･y4･x5･y6･x7･x8･y8 ⊕ y2･y3･y4･x5･y6･y7･x8･y8
⊕ y2･y3･y4･y5･x6･x7･x8･y8 ⊕ y2･y3･y4･y5･x6･y7･x8･y8 ⊕ y2･y3･y4･y5･y6･x7･x8･y8
⊕ y2･y3･y4･y5･y6･y7･x8･y8
z8 = x2 ⊕ y2 ⊕ x3･y3 ⊕ x3･x4･y4 ⊕ y3･x4･y4 ⊕ x3･x4･x5･y5 ⊕ x3･y3･x5･y5 ⊕ y3･x4･x5･y5 ⊕ y3･y4･x5･y5
⊕ x3･x4･x5･x6･y6 ⊕ x3･x4･y5･x6･y6 ⊕ x3･y4･x5･x6･y6 ⊕ x3･y4･y5･x6･y6
⊕ y3･x4･x5･x6･y6 ⊕ y3･x4･y5･x6･y6 ⊕ y3･y4･x5･x6･y6 ⊕ y3･y4･y5･x6･y6
⊕ x3･x4･x5･x6･x7･y7 ⊕ x3･x4･x5･y6･x7･y7 ⊕ x3･x4･y5･x6･x7･y7 ⊕ x3･x4･y5･y6･x7･y7
⊕ x3･y4･x5･x6･x7･y7 ⊕ x3･y4･x5･y6･x7･y7 ⊕ x3･y4･y5･x6･x7･y7 ⊕ x3･y4･y5･y6･x7･y7
⊕ y3･x4･x5･x6･x7･y7 ⊕ y3･x4･x5･y6･x7･y7 ⊕ y3･x4･y5･x6･x7･y7 ⊕ y3･x4･y5･y6･x7･y7
⊕ y3･y4･x5･x6･x7･y7 ⊕ y3･y4･x5･y6･x7･y7 ⊕ y3･y4･y5･x6･x7･y7 ⊕ y3･y4･y5･y6･x7･y7
⊕ x3･x4･x5･x6･x7･x8･y8 ⊕ x3･x4･x5･x6･y7･x8･y8 ⊕ x3･x4･x5･y6･x7･x8･y8 ⊕ x3･x4･x5･y6･y7･x8･y8
⊕ x3･x4･y5･x6･x7･x8･y8 ⊕ x3･x4･y5･x6･y7･x8･y8 ⊕ x3･x4･y5･y6･x7･x8･y8 ⊕ x3･x4･y5･y6･y7･x8･y8
⊕ x3･y4･x5･x6･x7･x8･y8 ⊕ x3･y4･x5･x6･y7･x8･y8 ⊕ x3･y4･x5･y6･x7･x8･y8 ⊕ x3･y4･x5･y6･y7･x8･y8
⊕ x3･y4･y5･x6･x7･x8･y8 ⊕ x3･y4･y5･x6･y7･x8･y8 ⊕ x3･y4･y5･y6･x7･x8･y8 ⊕ x3･y4･y5･y6･y7･x8･y8
⊕ y3･x4･x5･x6･x7･x8･y8 ⊕ y3･x4･x5･x6･y7･x8･y8 ⊕ y3･x4･x5･y6･x7･x8･y8 ⊕ y3･x4･x5･y6･y7･x8･y8
⊕ y3･x4･y5･x6･x7･x8･y8 ⊕ y3･x4･y5･x6･y7･x8･y8 ⊕ y3･x4･y5･y6･x7･x8･y8 ⊕ y3･x4･y5･y6･y7･x8･y8
⊕ y3･y4･x5･x6･x7･x8･y8 ⊕ y3･y4･x5･x6･y7･x8･y8 ⊕ y3･y4･x5･y6･x7･x8･y8 ⊕ y3･y4･x5･y6･y7･x8･y8
⊕ y3･y4･y5･x6･x7･x8･y8 ⊕ y3･y4･y5･x6･y7･x8･y8 ⊕ y3･y4･y5･y6･x7･x8･y8 ⊕ y3･y4･y5･y6･y7･x8･y8

付録C.2 S1

x1,x2,⋯,x8 および y1,y2,⋯,y8 は関数 S の入力ビットを表し、z'1,z'2,⋯,z'8 は S1 の出力ビットを表す。x1,y1,z'1 を、それぞれ最左端ビット(MSB)とする。また、多項式中の z1,z2,⋯,z8 は S0 の出力ビットブール多項式を表す(付録 C.1)。

z'1 = z1 ⊕ z2･z3･z4･z5･z6
z'2 = z2 ⊕ z3･z4･z5･z6
z'3 = z3 ⊕ z4･z5･z6
z'4 = z4 ⊕ z5･z6
z'5 = z5 ⊕ z6
z'6 = z6
z'7 = z7 ⊕ z8･z1･z2･z3･z4･z5･z6
z'8 = z8 ⊕ z1･z2･z3･z4･z5･z6

付録D FEAL-N に対する解析の歴史

FEAL が発表されてからの FEAL-N に対する解析を、年代順に記載しておく。FEAL-N と FEAL-NX の違いは鍵処理部のみである。そのため、ほとんどの解析は FEAL-NX に置き換えて見ることができる。

表 6 FEAL-N 解析記録

発表年	文献	段数	データ量	計算量	
1988	[B88]	4	100～10,000		選択平文攻撃
1990	[GC90]	8	10,000		選択平文攻撃
1991	[K91]	4	78		既知平文攻撃
1991	[TG91]	4	1000		既知平文攻撃
		6	20000		
1991	[BS91]	8	1,000		差分攻撃
		31	2^{63}		
1992	[MY92]	4	5		既知平文攻撃
		6	100		
		7	2^{14}		
		8	2^{15}	＜ 2^{64}	
		8	2^{28}	＜ 2^{50}	
1993	[TOD93]	4	1	2^{9}	中間メッセージ法
		5	1	2^{17}	
		6	1	2^{25}	
1993	[KK93]	4	36		既知平文攻撃
1994	[B94]	8	2^{24}		線形解読
1994	[OA94-1] [OA94-2] [AOAM94]	8	2^{16} 実験では 2^{25}		線形解読
1994	[KR94]	8	実験で 2^{23}		線形解読
1994	[MK94]	4	5		差分解読法
		6	100		
		7	16,000		
1995	[AO95]	8	12		差分線形解読