付録H: SC2000 に対する諸調査報告書

1. はじめに

情報処理振興事業協会(IPA)詳細評価対象暗号 SC2000 に対して調査した結果を報告する。アルゴリズムの記述は省略した。応募書類仕様書を参照頂きたい。本報告書の構成は以下の通りである。

2 章 S-box に関する調査

3 章 F 関数に関する調査

4 章 暗号系全体に関する調査

2. S-box に関する調査

2.1. ブール代数次数に関する調査

SC2000 では、S4、S5、S6 の 3 種類の S-box が使用されている。表 1、2、3 に SC2000 で用いられている S-box の各 bit の最大次数, 総項数及び最大次数項を示す。表 1、2、3 は、Si-box (i=4,5,6)の入力を $x=(x_1,x_2,...,x_i)$, 出力を $y=(y_1,y_2,...,y_i)$ とし、最大次数項は、例えば x_1 の項が入っていない 3 次項を $x^3[1]$、x_1、x_2 の項が入っていない 3 次項を $x^3[1,2]$と表記する。

S4、S5、S6 は、それぞれ、4,5,6 の入出力ビット数を持つ全単射関数であり、項数の期待値 $2^3,2^4,2^5$ の近辺に各出力ビットの項数は分布しており、特に偏りはみられない。

出力 bit	最大次数	総項数	最大次数項
y1	3	8	$x^3[4],x^3[3]$
y2	3	7	$x^3[4],x^3[3]$
y3	3	9	$x^3[1]$
y4	3	7	$x^3[2]$

表 1. S4 の最大次数, 総項数及び最大次数項

出力 bit	最大次数	総項数	最大次数項
y1	3	11	$X^3[3,5],x^3[2,5],x^3[2,4],x^3[1,3],x^3[1,2]$
y2	3	14	$x^3[3,4],x^3[2,4],x^3[1,2]$
y3	3	18	$x^3[4,5],x^3[3,5],x^3[2,5],x^3[2,4],x^3[2,3],x^3[1,5],x^3[1,4],x^3[1,3],x^3[1,2]$
y4	3	13	$X^3[4,5],x^3[3,5],x^3[1,5],x^3[1,3],x^3[1,2]$
y5	3	13	$x^3[3,5],x^3[3,4],x^3[2,4],x^3[1,5],x^3[1,4],x^3[1,2]$

表 2. S5 の最大次数, 総項数及び最大次数項

出力 bit	最大次数	総項数	最大次数項
y1	5	27	$x^5[5],x^5[4],x^5[3]$
y2	5	28	$x^5[5],x^5[4],x^5[2],x^5[1]$
y3	5	35	$x^5[6],x^5[4],x^5[3],x^5[2],x^5[1]$
y4	5	31	$x^5[5],x^5[4]$
y5	5	38	$x^5[6],x^5[5],x^5[4],x^5[3],x^5[2],x^5[1]$
y6	5	34	$x^5[4],x^5[1]$

表 3. S6 の最大次数, 総項数及び最大次数項

2.2. 補間多項式項数

各 S-box の補間多項式項数は下記の表に示す通りである。なお、S5 で原始多項式が 0x25 のときと、S6 で原始多項式 0x43 のときに項数が他の原始多項式のときに比べて著しく小さい特徴がある。この原始多項式は、それぞれ S5、S6 の設計に用いられた原始多項式である。現在のところ、この特徴を利用した補間攻撃は SC2000 に対し見つかっていない。

原始多項式	項数	原始多項式 （相反多項式）	項数
0x13	13	0x19	11

表 4.S4 の補間多項式項数

原始多項式	項数	原始多項式 (相反多項式)	項数
0x25	5	0x29	25
0x3d	26	0x2f	26
0x37	25	0x3b	26

表 5. S5 の補間多項式項数

原始多項式	項数	原始多項式 (相反多項式)	項数
0x43	7	0x61	61
0x67	63	0x73	63
0x6d	61	0x5b	62

表 6. S5 の補間多項式項数

3. F 関数に関する調査

3.1. ブール展開式項数（6 次まで）

F 関数は 32 ビットブロック×2 を入力し、32 ビットブロック×2 を出力する関数である。F 関数は図 1 に示すように S 関数、M 関数、L 関数から構成されている。L 関数には mask 値が入力される。その値は 0x55555555 と 0x33333333 である。ここでは mask 値が 0x55555555 のとき F5 関数、mask 値が 0x33333333 のとき F3 関数と呼ぶことにする。F5 関数と F3 関数とそれぞれに対して 64bit 入出力でブール展開式項数をそれぞれ求めた。（図 1 参照）結果は下記に示す通りである。

F 関数の非線形要素である S5,S6 のブール代数次数は、それぞれ 3 次、5 次であるため、低次項に集中している様子が伺える。1 段消去型攻撃では、線形化して攻撃方程式を解くことが有効と考える。

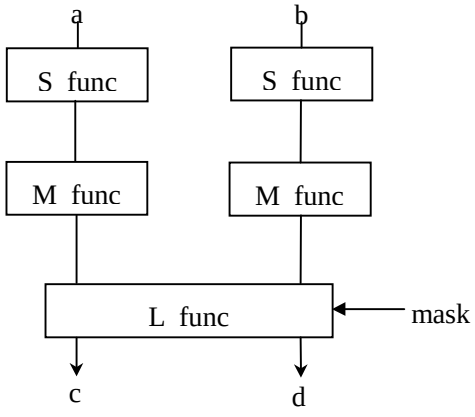

図 1. F 関数

次数	平均	最大値	最小値
1 次	49.219	42	11
2 次	106.219	84	28
3 次	122.344	96	33
4 次	45.656	40	9
5 次	17.531	22	3

表 7. F5 関数のブール展開式項数

次数	平均	最大値	最小値
1 次	49.219	42	11
2 次	106.219	84	28
3 次	122.344	96	33
4 次	45.656	40	9
5 次	17.531	22	3

表 8. F3 関数のブール展開式項数

4. 暗号系全体に関する調査

4.1. 高階差分特性(bit oriented 8 階差分まで)

解析の方法

SC2000 では、B 関数、R 関数という 2 種類の攪拌関数があるために B 関数、R 関数を個別に 8 階差分まで解析しても、暗号系全体として見たときの評価が困難であると思われる。そこで、一番左の 32 ビットブロックに対して bit oriented n階差分を鍵 100 通りに対してn=1 から解析し、差分値が all 0 という結果が得られたら、段数を 1 段伸ばすという方法で解析を行った。

結果を下記の表に示す。3 段で 6 階差分が 0 となる平文組が見つかり、4 段まで延ばすと 8 階差分まで差分値が 0 となる平文組は見つからなかった。

通過段(m 段)	bit oriented n 階差分値							
	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8
I-B(1 段)	—	all 0 (i)						
I-B-I-R(2 段)	—	all 0 (ii)						
I-B-I-R-R(3 段)	—	—	—	—	—	all 0 (iii)		
I-B-I-R-R-I-B(4 段)	—	—	—	—	—	—	—	—

表 9. m 段通過後のn階差分値

(i) ${}_{32}C_2$ 通りすべてで 2 階差分値＝all 0

(ii) 32 ビットを 6、5、5、5、5、6 という小ブロックに区切ったときのそれぞれの小ブロックをまたいで変数を 2 ビット選んだ場合(426 通り)で 2 階差分値=all 0

(iii) (ii)で述べた小ブロックのそれぞれから変数を 1 ビット選んだ場合(22500 通り)で 6 階差分値＝all 0

後述のように、B 関数、R 関数はそれぞれ、3 次、5 次であるが、2階差分(i) (ii)が 0 となるのは、B 関数の bitslice 構造の影響である。

4.2. 高階差分特性(32 階差分)

入力の 32 ビットブロックを左から順に A,B,C,D とするとき、A を全通り廻す 32 階差分値を、異なる平文固定値に対し調べた。ここで、攪拌関数は B 関数と R 関数であり、B 関数、R 関数段通過後の出力差分値に着目している。このとき各段の出力差分値を 32 ビットブロックで左から H[0],H[1],H[2],H[3]と表すことにする。求めた結果を下記の表に示す。これより 5 段通過後まで、H[2],H[3]の 32 階差分値が 0 であると考えられる。

通過段	32 階差分値 (固定値 B=0,C=0,D=0)	32 階差分値 (固定値 B=0x12345678,C=0,D=0)
I-B-I-R-R-I-B 4段	H[0]=0,H[1]=0,H[2]=0,H[3]=0	H[0]=0,H[1]=0,H[2]=0,H[3]=0
I-B-I-R-R-I-B-I-R 5段	H[0]=0xFB697785,H[1]=0x1A7583C6, H[2]=0,H[3]=0	H[0]=0x65A4FF8B,H[1]=0xE5AF0814, H[2]=0,H[3]=0
I-B-I-R-R-I-B-I-R-R 6段	H[0]=0xCF3D96A3,H[1]=0xCA5695ED, H[2]=0xFB697785,H[3]=0x1A7583C6	H[0]=0x4DBF4DD0,H[1]=0x007E4E60, H[2]=0x65A4FF8B,H[3]=0xE5AF0814

表 10. 各段通過後の 32 階差分値

4.3. SQUARE 型攻撃に関する調査

S6 を全通り廻して balance させるという観点から入力平文の 32 ビットブロック A,B,C,D の上位 6 ビットづつを全通り廻す 24 階差分値を鍵 100 通りに対して各段ごとに求めた。各段の出力差分値を左から H[0],H[1],H[2],H[3]と表すことにする。求めた結果を下記の表に示す。これより 5 段通過後まで H[2],H[3]の 24 階差分値が 0 であると考えられる。

通過段	24 階差分値
I-B-I-R-R-I-B 4段	H[0]=0,H[1]=0,H[2]=0,H[3]=0
I-B-I-R-R-I-B-I-R 5段	H[0]=変数,H[1]=変数,H[2]=0,H[3]=0
I-B-I-R-R-I-B-I-R-R 6段	H[0]=変数,H[1]=変数,H[2]=変数,H[3]=変数

表 11. 各段通過後の 24 階差分値

4.4. 形式的代数次数に関する調査

SC2000 で用いられている関数は、I 関数、B 関数、R 関数である。I 関数は拡大鍵を排他的論理和する関数なので、1 次である。B 関数は S4 が 3 次であることより、3 次である。R 関数は S5 が 3 次、S6 が 5 次であり、5 次である。自己評価書では、R 関数内の L 関数において、マスクをとって隣の 32 ビットデータブロックと XOR するという演算による次数の降下の可能性を考慮し、最低保障次数 2 次と見積もっている。しかしながら、3.1 節の調査結果から R 関数の全出力ビットは、5 次になっている事が確認されている。

以下R関数各関数の形式的代数次数を下記の表に示す。R 関数の(2次)は、自己評価書の評価である。

関数	形式的代数次数
I 関数	1 次
B 関数	3 次
R 関数	5 次(2 次)

表 12. 各関数の形式的代数次数

表 12 をもとに暗号系全体について形式代数次数を見積もった。結果を下記の表に示す。自己評価書と同じく

R 関数の次数を 2 次と評価した時の次数を参考のため括弧内に示す。ここで、段数は B 関数と R 関数の段数の合計であり、I 関数はカウントしていない。(各 32 ビットブロックが 4 つあるがその形式的代数次数のうち最小値を記載する)

通過段数	各段通過後の形式的代数次数
1 段	3 次
2 段	3 次
3 段	15 次(6 次)
4 段	128 次(36 次)
5 段	(36 次)
6 段	(72 次)
7 段	(128 次)

表 13. 暗号系全体の形式的代数次数

4.5. 暗号系の高階差分耐性

前節まで及び、付録Jの線形和攻撃の中で、最も効果的な高階差分は、4.3 節のものであり、5 段目出力 64 ビットの 24 階差分が 0 となる。これを使い、SC2000 の高階差分攻撃耐性を調べる。6 段目の R 関数の拡大鍵 64 ビットを求める 1 段消去型攻撃で考える。R 関数についての 3 節の考察から、攻撃方程式に、本文 3.2 節線形化攻撃の手法を適用する。R 関数には、S5 ボックスが 8 個、S6 ボックスが 4 個使われ、それぞれ、代数次数 3 次、5 次である。各 S ボックスの出力に現れる、項の種類数は S5 が 26 項、S6 が 63 項である。従って、R 関数を線形化したときの項数は L=26*8+63*4=460 となる。攻撃方程式は、64 ビット幅であり、必要な 24 階差分組数は、$M = \lfloor 460/64 \rfloor = 7.1 \approx 2^3$ であり、平文組数は $2^{24} M = 2^{27}$ である。計算量は

$$T_{6段} = 2^{24} \frac{ML}{12} \approx 2^{33}$$

の R 関数計算量である。ここで、上式の分母は、R 関数には、S ボックスが 12 個ある事を考慮したものである。

2 段消去型攻撃は、B 関数に係わる 128 ビットを総当たりして、前述の 1 段攻撃を適用すれば良い。3 段消去型では R 関数に係わる 64 ビット鍵を総当たりして 2 段消去型攻撃を実行する。解読に必要な計算量及び平文組数を本文(3.10)式で計算すれば、表 14 である。

ここで、発見した高階差分攻撃法では、8 段まで可能であるが、自己評価書では、鍵長 128 ビットで 17 段以上、192,256 ビットで 19 段以上であれば、改良型高階差分攻撃は成功しないと見積もっている。その評価は、代数次数の見積もりが表 13 の括弧内の数字で与えられことを前提にしており、やや次数の過小評価(安全側の評価)の傾向が見られる。

攻撃のタイプ	適用段数	必要選択平文数	段関数計算量
24 階差分	6 段	2^{27}	2^{33}
24 階差分+2 段消去型	7 段	2^{27}	2^{161}
24 階差分+3 段消去型	8 段	2^{27}	2^{225}

表 14. 高階差分攻撃の計算量見積もり