付録Ⅰ：MARS 暗号の代数的調査

1. はじめに

本稿は、情報処理振興事業協会(IPA)の詳細評価対象暗号である MARS について、耐代数的攻撃法という観点から調査した。アルゴリズムの説明は省略したので、必要に応じ応募書類仕様書を参照頂きたい。本報告書は以下のように構成している。

2章　S-box についての調査

3章　Round 関数（E 関数）についての調査

4章　MARS の Cryptographic Core についての調査

2. S-box について

2.1 ブール代数次数

S-box のブール代数次数を調査した。この S-box は 9 ビット入力 32 ビット出力である。このとき、入力を $x=(x_8,x_7,\cdots,x_1,x_0)$、出力を $y=(y_{31},y_{30},\cdots,y_1,y_0)$としている。結果は、以下の表 1 ようになり、最大次数 9 次、最小次数 8 次であった。9 ビット入力の関数として、期待される平均項数 256 に近い値が偏り無く出ている。

出力 bit	最大次数	総項数	出力 bit	最大次数	総項数
y_{31}	8	240	y_{15}	9	249
y_{30}	8	246	y_{14}	8	252
y_{29}	9	264	y_{13}	8	267
y_{28}	9	249	y_{12}	8	263
y_{27}	8	246	y_{11}	9	260
y_{26}	9	252	y_{10}	9	251
y_{25}	8	252	y_9	9	258
y_{24}	9	242	y_8	9	254
y_{23}	9	264	y_7	9	251
y_{22}	9	257	y_6	9	230
y_{21}	8	259	y_5	9	248
y_{20}	9	255	y_4	9	243
y_{19}	8	267	y_3	9	246
y_{18}	8	243	y_2	8	267
y_{17}	8	255	y_1	9	239
y_{16}	9	261	y_0	9	255

表 1　　S-box の出力ビット位置におけるブール代数次数

2.2 S-box の補間多項式項数

MARS の S-box は 9 ビット入力であるため、テーブルの要素が 512 個あるが、これは各 256 個の要素をもつテーブル S1 と S2 を結合して成り立っている。S1 と S2 それぞれについて GF(2^8)の多項式表現した場合の次数と項数を原始多項式を変えて調査した。原始多項式は、各次数の係数を高次項から並べ、それを 16 進表現してある。即ち、原始多項式 0x11d は $x^8+x^4+x^3+x^2+1$ を表す。また、出力は S1、S2 共に 32 ビット出力であり、下位から 8 ビットずつに区切った小ブロックの並びと考えた。結果を表 2 及び 3 に示す。

結果は、最小 253 項、最大 256 項であり、8 ビット入力の関数としての最大値 256 項に近い項数である。特定の原始多項式で項数が少なくなる事は無く、原始多項式の選択で、補間攻撃の効率が上がることは期待薄である。

原始多項式	項数 31～24 , 23～16 , 15～8 , 7～0				原始多項式 (相反多項式)	項数 31～24 , 23～16 , 15～8 , 7～0			
0x11d	255	255	256	256	0x171	254	255	255	255
0x169	255	254	255	256	0x12d	256	254	254	255
0x1e7	255	255	254	255	0x1cf	256	255	255	255
0x12b	253	256	255	256	0x1a9	255	256	255	255
0x165	255	254	256	256	0x14d	256	255	253	255
0x163	256	256	255	253	0x18d	256	253	256	255
0x15f	256	256	256	255	0x1f5	254	254	255	255
0x1c3	254	254	256	255	0x187	255	255	256	255

表 2　　S1 について 8 ビットずつ GF(2^8)の多項式表現した場合の項数

原始多項式	項数 31～24 , 23～16 , 15～8 , 7～0				原始多項式 (相反多項式)	項数 31～24 , 23～16 , 15～8 , 7～0			
0x11d	255	255	256	255	0x171	255	254	256	254
0x169	255	256	256	256	0x12d	256	255	256	256
0x1e7	255	256	255	256	0x1cf	256	255	255	254
0x12b	255	256	256	255	0x1a9	256	256	255	256
0x165	253	254	256	256	0x14d	256	254	256	256
0x163	254	256	255	256	0x18d	255	256	256	254
0x15f	255	254	255	255	0x1f5	256	255	254	255
0x1c3	256	255	253	255	0x187	256	255	254	255

表 3　　S2 について 8 ビットずつ GF(2^8)の多項式表現した場合の項数

※「項数」の下の数字はビット位置を示す。

2.3 S-box の高階差分特性

MARS の S-box を構成している S1、S2 それぞれについての高階差分特性を調査した。

S1 と S2 に分けて考えると、それぞれ 8 ビット入力、32 ビット出力の S-box と見ることができるので、それぞれについて 8 階まですべてのパターンの高階差分組に対し、出力差分が 0 となる差分組の個数を調査した。

その結果、S1、S2 ともに固定値に依存せず、8 階まですべてのパターンの高階差分組に

対し、出力差分が 0 になるものは見つからなかった。

3. Round 関数（E 関数）について

MARS は E 関数と呼ばる Round 関数をもち、以下の図 1 ような構造をしている。この E 関数は、32 ビット入力、96 ビット出力(R,M,L…各 32 ビット)である。

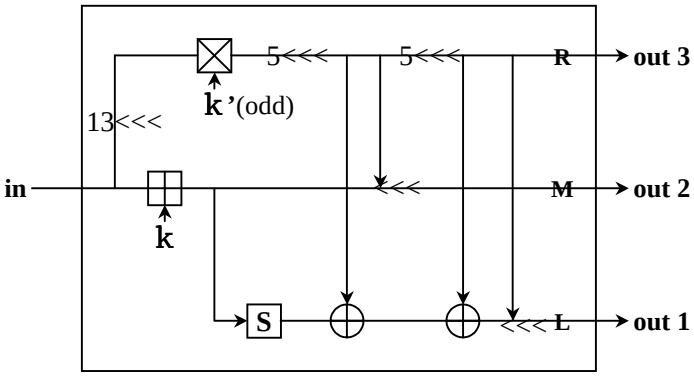

図 1　　E 関数

3.1　ブール展開式項数(6 次項まで)

MARS で用いられている E 関数について、入力拡大鍵を 0 として、出力,L についてのブール展開式表現時における 1〜6 次項までの項数を調査した。結果を以下、表 4 に示す。調査項目としては、項数の最大値、最小値、平均値である。S-box は 9 ビット入力 32 ビット出力である。ここで、鍵 k が 0 ということにより、入力の下位 9 ビットがそのまま S-box の入力となる。その後 out3 からのデータの排他的論理和やデータ依存ビットシフトも 0 となり、結局、入力の下位 9 ビットだけにしか依存していない。そのためここでの項数の期待値は ${}_9C_n/2$ (n…次数)で表される。ほぼ、期待値に近い項数の平均値が出ており、特に偏りは見られない。

次数	平均	最大値	最小値	期待値
1 次	4.219	8	1	4.5
2 次	16.906	24	8	18
3 次	41.625	49	33	42
4 次	62.719	73	54	63
5 次	61.656	72	50	63
6 次	42.750	53	34	42

表 4　　E 関数の出力 L についてのブール多項式項数

3.2　高階差分特性

MARS の E 関数について、bit-oriented で 1〜8 階までの高階差分特性を全パターン、現在調査中である。

4. MARS の Cryptographic Core について

4.1 形式的代数次数

まずは E 関数の out3(R)につながるラインに注目する。入力が、まず 13 ビット左巡回シフトされ、鍵と乗算される。このとき 2^{32} を法とする乗算のため、鍵乗算後の最大次数は 31 次となる。以後の固定巡回シフトでは次数は上がらないので、out3(R)では考えられる最大次数は 31 次となる。

次に out2(M)につながるラインに注目する。この場合、入力に鍵が算術加算されているので、鍵加算後の最大次数は 31 次となる。次にデータ依存型巡回シフトを通過するが、このデータ依存型巡回シフトは、out3(R)ラインの下位 5 ビットに依存している。このデータ(5 ビット)依存型巡回シフトでは、入力 1 次に対して出力は 6 次となる。その為、最大次数は 32 次を超えてしまうので out2(M)ラインの最大次数は 32 次となる。

次に out1(L)につながるラインに注目する。この部分では、out2(M)ラインの下位 9 ビットに依存して、S-box の入力が決まる。この S-box は最高 9 次(最低 8 次)であるので、このラインも 32 次を超えてしまう。その為、out1(L)ラインの最大次数は 32 次となる。

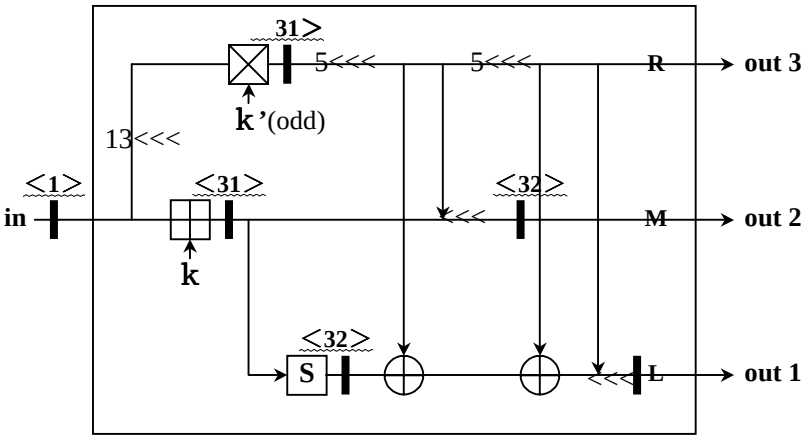

図 2　　E 関数の各部における次数

4.2 SQUARE 攻撃耐性

図 3 のように Cryptographic core 部の前半部の 4 つの 32 ビットワードを A、B、C、D とする。まずはじめに E 関数について考える（4.1 参照）。E 関数の out3(R)に注目してみると、ブール代数次数を増加させているのは鍵（k'(odd)）の乗算のみであり、これは入力に関して全単射処理である。そのため、図 3 において入力が全通り廻ったとき、out3(R)も全通り廻る。よって、Cryptographic core 部の前半部 1 段において A_0 に 32 階差分を与え、B_0、C_0、D_0 を“ 0 ”とすると、E 関数を通過した後の C1 の 32 階差分は“ 0 ”となる。さらに、D_1 は、A_0 を固定回シフト（左 13 ビットシフト）しただけなので、この部分も全通り廻り、32 階差分値も“ 0 ”である（図 3 参照）。

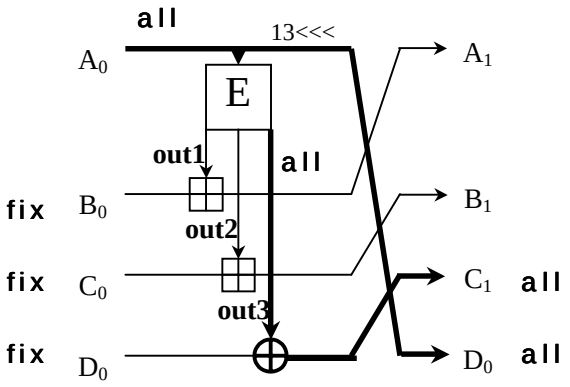

図 3　　Cryptographic core 部の前半部

次に、図 4 のように A_0、B_0、C_0 を fix、D_0 を変数（32 ビット）に選ぶと、Cryptographic core 前半部の構造から、4 段目で初めて変数が E 関数に入力される。このとき、他の 3 ワードは fix である。このように変数を定めた時、上記の 32 階差分の関係が 4 段目に現れる。従って、Cryptographic core 前半 5 段目に SQUARE 攻撃が適用可能である。そのため、よって、4 段目出力の C_4 と D_4 の 32 階差分値は鍵に依存しないで常に“ 0 ”となる。このことは計算機実験により確認した。

１段消去型攻撃で 5 段を攻撃するならば、次の攻撃方程式となる。

$$\bigoplus_{A\in V^{32}} \left[\left\{\left(D_5 \times k'(odd)\right) <<< 10\right\} \oplus C_5\right] = 0 \tag{1}$$

$$\bigoplus_{A\in V^{32}} \left[B_5 - \left\{\left(D_5 >>> 13\right) + k\right\} <<< \left\{\left(D_5 \times k'(odd)\right) <<< 5 \mod 2^5\right\}\right] = 0 \tag{2}$$

(1)、(2)式を解くことにより Cryptographic core 部の前半 5 段目で用いる鍵（k’(odd)、k）が求められる。

5. 高階差分攻撃耐性

現在の判っている最も効果的な高階差分は 4.2 節の SQUARE 型の 32 階差分攻撃である。これを使って攻撃したときの攻撃可能段数及び計算量を見積もる。

式(1)の拡大鍵 *k*’(odd)を求める際は、総当たりでは無く、鍵の下位ビットから順次求めていく方法が効果的である。*k*’(odd)の最下位 2 ビットは仕様により 11 と定められているので、3 ビット目から計算することになる。鍵の乗算結果は 10 ビット左巡回シフトされるので、式(1)の 13 ビット目のみ 1 ビットに着目する事になる。この 1 ビットの方程式で、1 ビットの鍵 2 通り(0 又は 1)を篩にかける事になる。１ビットの攻撃方程式は、偽鍵でも 1/2 の確率

で成立する。偽鍵が生き残っても上位ビットの方程式でさらなる絞り込みが可能であるが、一つの偽鍵に対し、1 ビット上の方程式では、候補数が 2 倍に増加するので、計算量の指数関数的増大を押さえるには、偽鍵生き残り数の期待値を 1/2 未満にする事が必要である。それには、32 階差分組を 2 組用意すれば良い。鍵の 3 ビット目が定まれば、4 ビット目に関し、同じ手順を踏む。これを繰り返して、鍵の最上位ビットまで定める事が可能である。この時の計算量は、推定すべき鍵ビット数が30ビットであることから、$30*2*2*2^{32} \approx 2^{39}$ 回の乗算計算である。

式(2)に付いても同様に、鍵 k の下位ビットから順次定めていけば、最上位ビットを除く 31 ビットが確定し、そのときの計算量は $31*2*2*2^{32} \approx 2^{39}$ 回の加算と乗算である。なお、必要な 32 階差分 2 組は、式(1)を解くために使用したものを使えば良く、新たな高階差分組は必要ない。以上の手順で 5 段目の拡大鍵（k'(odd)、k）が求められる。E 関数の計算量を、シフト演算を無視して、乗算 2 回、加算 1 回相当と見積もるならば、

$$T_{5段} = 2^{39}$$

回の E 関数計算量であり、必要平文数は 2^{33} 組である。

さらに、２段消去型攻撃で 6 段を攻撃するならば、6 段目の拡大鍵 64 ビットを総当たりして、上述の１段消去攻撃を適用すればよい。計算量を減らす為に、少し多めの平文を用意するのであれば、5 段目の *k'*(odd)の 3 ビット目を絞り込む際の偽鍵生き残り数の期待値を 1/2 未満にすればよい。この時、32 階差分組は 65 組必要であり、平文組数では $2^{32} * 65 \approx 2^{39}$ 組となる。計算量は

$$T_{6段} = 2^{39} * 2^{64} * 2 = 2^{104}$$

回の E 関数計算である[12]。

同様な考え方で、256 ビットの秘密鍵の総当たり計算量以下で解読が可能となる範囲を示せば表 5 である。

攻撃のタイプ	適用段数	必要選択平文数	段関数計算量
32 階差分	5 段	2^{33}	2^{39}
32 階差分+2 段消去型	6 段	2^{39}	2^{104}
32 階差分+3 段消去型	7 段	2^{40}	2^{168}
32 階差分+4 段消去型	8 段	2^{40}	2^{232}

表 5.　高階差分攻撃の計算量見積もり

[1] 解読計算量は、2 組の 32 階差分を使って、鍵候補 2^{65} 組を 2^{64} 組に絞り、それをさらに次の 32 階差分で絞る事を繰り返すと考えて評価した。

[2] 正確に評価すれば、5 段目 *k'*(odd)の 4 ビット目の計算においては、6 段目鍵 64 ビットの総当たりは必要なく、計算量はこの 1/30 程度と評価される。

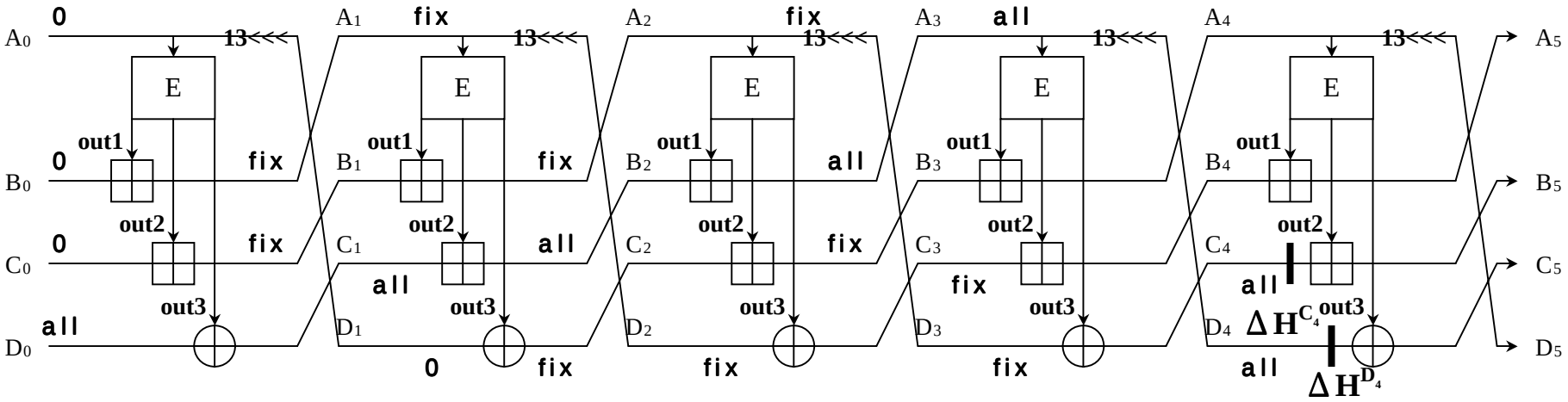

図 4　　Cryptographic Core　前半部　5 段分