CIPHERUNICORN-A に対する安全性について

2001 年 12 月 19 日

日本電信電話株式会社

神田 雅透

CIPHERUNICORN-A に対する安全性について

日本電信電話株式会社

December 19, 2001

Abstract

本稿では，CIPHERUNICORN-A の差分特性確率について，設計者の立場から評価を実施した。その結果，差分解読法では，仕様である 16 段について，高い確率で攻撃が成功することはないとの結論に達した。

1 安全性検証に関する考え方

本稿では，CIPHERUNICORN-A の差分解読法に対する安全性を検証する。今回は設計者の立場より評価を行うものとし，仕様である 16 段が攻撃可能かどうかについてのみ検討した。つまり，攻撃可能段数を検討したものではないことに注意されたい。

CIPHERUNICORN-A の検討すべき特徴点としては，

- 本流部での Feistel 型構造の効果
- A3 関数の効果
- 一時鍵生成部の効果

が挙げられる。もしも攻撃可能段数を求めようとするならば，これらの効果について厳密な調査を実施する必要がある。しかし，暗号仕様が安全であるか否かだけを検証するのであれば，必ずしも厳密に評価する必要はなく，適切な考察の下での安全性上界を調べればよい。この方針の下，安全性評価上，有効だと考える以下のような考察を行った。なお，副鍵の 32 ビット加算については排他的論理和とみなす。

本流部の Feistel 型構造の効果

本流部は，A3 関数，定数乗算，T0 関数，定数乗算，7 段の固定 Feistel 構造，2 段の一時鍵依存関数，そして，最後に一時鍵の挿入が行われる。

まず，A3 関数後の定数乗算から固定 Feistel 構造までの特性を考える。CIPHERUNICORN-A で利用する定数乗算は $\bmod 2^{32}$ での演算であるため，入力差分で差分の立っている最下位ビットと，それより上位のビットについてのみ差分波及が起こる。すなわち，差分の立っている最下位ビットが 31 ビット目（LSB) であれば出力差分の全ビットが差分波及の対象となる。しかし，入力差分の 0 ビット目 (MSB) にしか差分が立っていない場合，出力差分の MSB にしか差分は波及しない。つまり，入力差分が 0x80000000 であれば確率 1 で出力差分が 0x800000000 になる。同様の考え方をすると，上位 8 ビットに差分が立っている，すなわち 0x**000000（**はオールゼロではない）であるとすると，確率 1 で出力差分が 0x##000000 になる。以上のことは，定数乗算部については入力差分が上位に集まっているほど高い確率での正確な見積もりが可能になることを示している。

例として，入力差分として，(0x80000000,0x8A2240A7) を考える。

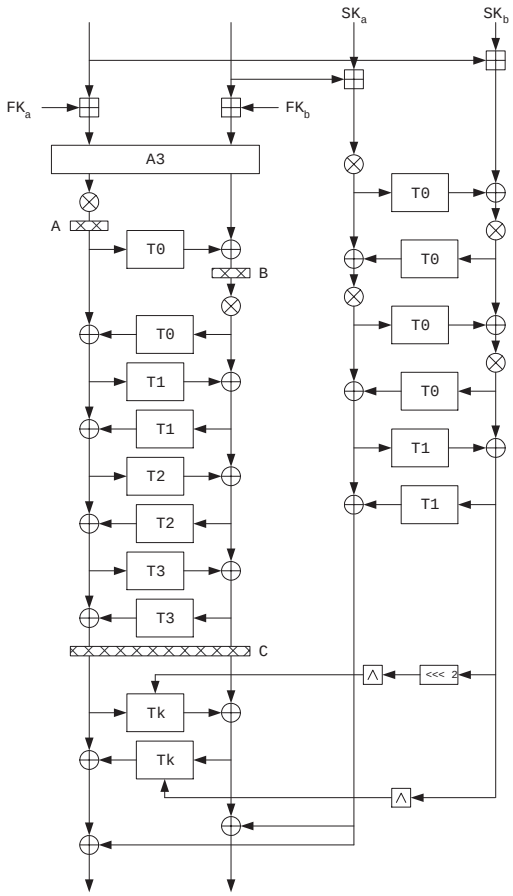

Figure 1: CIPHERUNICORN-A のラウンド関数

このとき，上述した理由によって，定数乗算を通過したところ (図 1 A の場所) での差分も確率 1 で同じ 0x80000000 となる。次に，T0 関数の差分特性を考慮すると，確率 2^{-7} で T0 関数の出力差分が 0x8A2240A7 となるものが存在する。このとき，定数乗算前のところ (図 1 B の場所) の差分はゼロとなる。そのあとは，すべての (固定)T 関数への入力差分はゼロであるので，同じ差分のまま伝播し，図 1 C の場所での差分は (0x80000000, 0x00000000) となる。

同様に，入力差分を (0xE6000000 0xB581090C) とする。このとき，定数乗算での差分伝播がランダムに起こると仮定すると，上位 6 ビットがランダムな振舞いをする (7 ビット目は必ず 1 になる) ので，A のところで 0xE6000000 となるものが確率 2^{-6} で発生する。T0 関数では同様に確率 2^{-7} で B における差分をゼロとするものが存在するので，この場合，確率 2^{-13} で図 1 C の場所での差分は (0x80000000, 0x00000000) となると期待される。

A3 関数の効果

A3 関数は，64 ビット入力 x に対し，$(x) \oplus (x <<< 23) \oplus (x <<< 41)$ を出力する関数である。なお，$<<<$ はローテーションを表す。これを各ビットごとに詳しく書き表すと，表 1 のようになる。なお，$[x]$ は入力の第 x ビット目の値を表す。

Table 1: A3 関数の入出力ビット関係

出力ビット位置	関連入力ビット	出力ビット位置	関連入力ビット
0	$[0] \oplus [23] \oplus [41]$	32	$[32] \oplus [55] \oplus [9]$
1	$[1] \oplus [24] \oplus [42]$	33	$[33] \oplus [56] \oplus [10]$
2	$[2] \oplus [25] \oplus [43]$	34	$[34] \oplus [57] \oplus [11]$
3	$[3] \oplus [26] \oplus [44]$	35	$[35] \oplus [58] \oplus [12]$
4	$[4] \oplus [27] \oplus [45]$	36	$[36] \oplus [59] \oplus [13]$
5	$[5] \oplus [28] \oplus [46]$	37	$[37] \oplus [60] \oplus [14]$
6	$[6] \oplus [29] \oplus [47]$	38	$[38] \oplus [61] \oplus [15]$
7	$[7] \oplus [30] \oplus [48]$	39	$[39] \oplus [62] \oplus [16]$
8	$[8] \oplus [31] \oplus [49]$	40	$[40] \oplus [63] \oplus [17]$
9	$[9] \oplus [32] \oplus [50]$	41	$[41] \oplus [0] \oplus [18]$
10	$[10] \oplus [33] \oplus [51]$	42	$[42] \oplus [1] \oplus [19]$
11	$[11] \oplus [34] \oplus [52]$	43	$[43] \oplus [2] \oplus [20]$
12	$[12] \oplus [35] \oplus [53]$	44	$[44] \oplus [3] \oplus [21]$
13	$[13] \oplus [36] \oplus [54]$	45	$[45] \oplus [4] \oplus [22]$
14	$[14] \oplus [37] \oplus [55]$	46	$[46] \oplus [5] \oplus [23]$
15	$[15] \oplus [38] \oplus [56]$	47	$[47] \oplus [6] \oplus [24]$
16	$[16] \oplus [39] \oplus [57]$	48	$[48] \oplus [7] \oplus [25]$
17	$[17] \oplus [40] \oplus [58]$	49	$[49] \oplus [8] \oplus [26]$
18	$[18] \oplus [41] \oplus [59]$	50	$[50] \oplus [9] \oplus [27]$
19	$[19] \oplus [42] \oplus [60]$	51	$[51] \oplus [10] \oplus [28]$
20	$[20] \oplus [43] \oplus [61]$	52	$[52] \oplus [11] \oplus [29]$
21	$[21] \oplus [44] \oplus [62]$	53	$[53] \oplus [12] \oplus [30]$
22	$[22] \oplus [45] \oplus [63]$	54	$[54] \oplus [13] \oplus [31]$
23	$[23] \oplus [46] \oplus [0]$	55	$[55] \oplus [14] \oplus [32]$
24	$[24] \oplus [47] \oplus [1]$	56	$[56] \oplus [15] \oplus [33]$
25	$[25] \oplus [48] \oplus [2]$	57	$[57] \oplus [16] \oplus [34]$
26	$[26] \oplus [49] \oplus [3]$	58	$[58] \oplus [17] \oplus [35]$
27	$[27] \oplus [50] \oplus [4]$	59	$[59] \oplus [18] \oplus [36]$
28	$[28] \oplus [51] \oplus [5]$	60	$[60] \oplus [19] \oplus [37]$
29	$[29] \oplus [52] \oplus [6]$	61	$[61] \oplus [20] \oplus [38]$
30	$[30] \oplus [53] \oplus [7]$	62	$[62] \oplus [21] \oplus [39]$
31	$[31] \oplus [54] \oplus [8]$	63	$[63] \oplus [22] \oplus [40]$

したがって，左半分のうち上位 8 ビットに出力差分を集めるような入力差分，すなわち，出力差分の 8 ビット目から 31 ビット目までをゼロとするようなを入力差分となるための条件を表 1 から求めると，以下のようになる。

$$
\begin{aligned}
&[49] = [54] = [8] \oplus [31] = [3] \oplus [26] = [13] \oplus [36] \\
&[50] = [9] \oplus [32] = [4] \oplus [27] \\
&[51] = [10] \oplus [33] = [5] \oplus [28] \\
&[52] = [11] \oplus [34] = [6] \oplus [29] \\
&[53] = [12] \oplus [35] = [7] \oplus [30]
\end{aligned}
$$

Table 2: 出力差分の 8-31 ビットがゼロとなるときの A3 関数の入出力ビット関係

出力ビット位置	関連入力ビット	出力ビット位置	関連入力ビット
0	$[41] \oplus [46]$		
1	$[42] \oplus [47]$		
2	$[43] \oplus [48]$		
3	$[44] \oplus [49]$		
4	$[45] \oplus [50]$		
5	$[46] \oplus [51]$		
6	$[47] \oplus [52]$		
7	$[48] \oplus [53]$		
32	$[50] \oplus [55]$	48	$[2] \oplus [7]$
33	$[51] \oplus [56]$	49	$[3] \oplus [8]$
34	$[52] \oplus [57]$	50	$[4] \oplus [9]$
35	$[53] \oplus [58]$	51	$[5] \oplus [10]$
36	$[54] \oplus [59]$	52	$[6] \oplus [11]$
37	$[55] \oplus [60]$	53	$[7] \oplus [12]$
38	$[56] \oplus [61]$	54	$[8] \oplus [13]$
39	$[57] \oplus [62]$	55	$[32] \oplus [37]$
40	$[58] \oplus [63]$	56	$[33] \oplus [38]$
41	$[59] \oplus [0]$	57	$[34] \oplus [39]$
42	$[60] \oplus [1]$	58	$[35] \oplus [40]$
43	$[61] \oplus [2]$	59	$[36] \oplus [41]$
44	$[62] \oplus [3]$	60	$[37] \oplus [42]$
45	$[63] \oplus [4]$	61	$[38] \oplus [43]$
46	$[0] \oplus [5]$	62	$[39] \oplus [44]$
47	$[1] \oplus [6]$	63	$[40] \oplus [45]$

$$[0] \oplus [23] \oplus [46] = 0,\ [1] \oplus [24] \oplus [47] = 0,\ [2] \oplus [25] \oplus [48] = 0$$
$$[14] \oplus [37] \oplus [55] = 0,\ [15] \oplus [38] \oplus [56] = 0,\ [16] \oplus [39] \oplus [57] = 0$$
$$[17] \oplus [40] \oplus [58] = 0,\ [18] \oplus [41] \oplus [59] = 0,\ [19] \oplus [42] \oplus [60] = 0$$
$$[20] \oplus [43] \oplus [61] = 0,\ [21] \oplus [44] \oplus [62] = 0,\ [22] \oplus [45] \oplus [63] = 0$$

この条件をもとに，その他の出力差分の各ビットについて入力差分との関係をみると表 2 のようになる。A3 関数は線形変換であるため，上記の条件を満たす限りにおいて，表 2 は確率 1 で成立する。

これらの情報をもとに，図 1 B のところの全ビットがゼロ差分になるものがあるかどうかを調査した。

その結果，128 通りの入力差分について，A3 関数の出力差分の上位 8 ビットが $0x80$ となり，確率 2^{-7} で図 1 B のところの全ビットがゼロ差分になるものが見つかった。このうち，もっとも入力ハミング重みが最小となる入力差分は，$(0x04A65520, 0x028A5388)$ のときのハミング重み 20 である。このとき，A3 関数を通過した出力差分は $(0x80000000, 0x8A2240A7)$ となる。

また，上位 8 ビットが非ゼロとなるという条件に緩めると，32640 通りの入力差分について，定数乗算の結果が同じ出力差分になるという仮定のもと，T0 関数の確率が 2^{-7} で図 1 B のところの全ビットがゼロ差分になるものが見つかった。このうち，もっとも入力ハミング重みが最小となる入力差分は，$(0x0200C984, 0x04130160)$ のときのハミング重み 14 である。このとき，A3 関数を通過した出力差分は $(0xE6000000, 0xB581090C)$ となる。

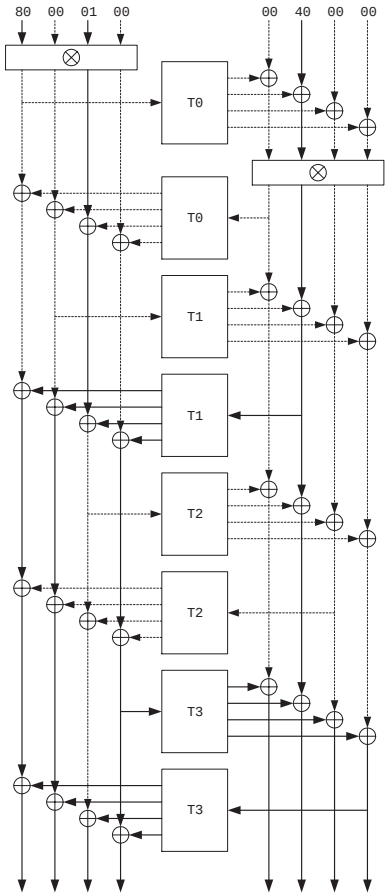

Figure 2: active s-box が 3 個の例

一方，入力差分として，$(0x80000000, 0x00000000)$ を考えた場合，A3 関数を通ることによって，$(0x80000100, 0x00400000)$ となる。図 1 B の直前の定数乗算の特性を考慮すると，下位ビットに差分が経つことは，B における差分予測を困難な（確率が小さくなる）方向へ作用するはずであるから，A における上位 8 ビットには差分がたたないほうが全体として差分特性確率を高いまま維持できるものと期待される。その仮定のもと，truncated vector での探索的な考え方をすると，図 2 のような active s-box が 3 個になるものが見つかる。このときの出力差分として，必ずしも $(0x04A65520, 0x028A5388)$ となるものが存在するとはいえないが，少なくとも truncated vector の意味で一致することから，定数乗算部での確率低下を除き，確率 $(2^{-7})^3 = 2^{-21}$ と見積もることとする。

以上の考察から，A3 関数での効果は，入力差分のハミング重みが少ないときには比較的効果的に入力差分を散らばらせ，定数乗算の影響を増す一方，逆にハミング重みが多いときにはハミング重みを減少させ，定数乗算の影響を減じてしまう可能性がある。

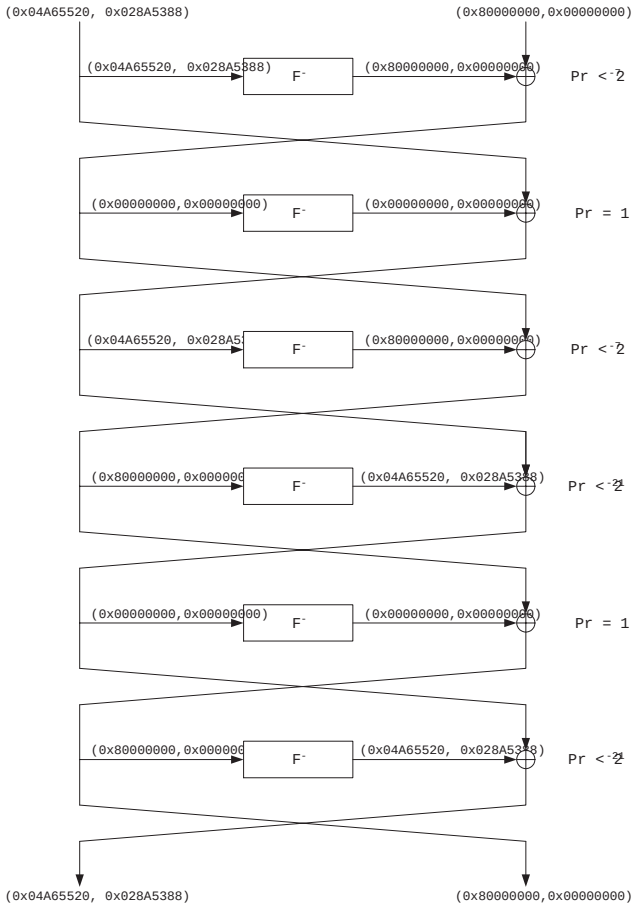

Figure 3: 一時鍵生成部を除いたラウンド関数での繰り返し表現の例

本流部のみによる考察のまとめ

一時鍵生成部を除いた部分 (図 1 の C より上部) だけを考えた場合，CIPHERUNICORN-A は全単射関数となっている。そこで，繰り返し表現を考えると，図 3 に示すものが,(存在したならば) かなり高確率な差分特性経路といえるであろう。

この結果を用いると，仕様と同じ 16 段で差分特性確率の上界値が 2^{-128} を下回ることがわかる。4 段目と 6 段目の定数乗算での影響を無視しているので，実際にはこの見積もりよりは安全であると期待される。ただし，今回の検討結果では必ずしも最良のものを見つけたというわけではないので，ここで検討したような本流部だけのラウンド関数であるとしたら，確実に差分解読法に対して安全であるとの確証は，今回の検討結果からだけではもてない。その意味において，CIPHERUNICORN-A の一時鍵生成部は重要な役割を果たしているものと期待される。

一時鍵生成部の効果

一時鍵生成部における効果としては二つ有り，一つは T 関数の構造を変える効果，もう一つが本流部に差し込まれる “副鍵” としての効果である。

このうち，T 関数の構造を変える効果について，もともと関連するビットが合計 4 ビットと少

ない上，Feistel 構造のラウンド関数をデータによって変えることになるため，その効果が設計者にとって必ず有利になる (望ましい) かどうかは即断できない。例えば，固定の全単射関数であれば，非ゼロ差分が入力されたとき，当然，非ゼロ差分が出力される。逆に，ゼロ差分が入力されれば，ゼロ差分が出力される。しかしながら，データによって構造を変えるということは，たとえ個々の関数としてみれば全単射であったとしても，非ゼロ差分の入力に対しゼロ差分が出力されたり，ゼロ差分の入力にもかかわらず非ゼロ差分が出力されるといった状況が起こりうる。こういった，通常の全単射関数では起こらないような状況を発生させることは，そのこと自体が確かに予測困難にしており，全体としてみれば安全性の向上に寄与しているとも言えなくはない。だが，一方で，こういった通常では起こらない特徴を利用して，逆に攻撃に利用させる懸念もある。そういった意味において，一時鍵生成部により T 関数の構造を変えるという効果については過大評価も過小評価もできないというのが現状であろうと考える。

これに対し，もう一方の “副鍵” としての効果はかなり効果が高いものと期待する。

つまり，本流部では，入力ハミング重みが多いとき，(一時鍵生成部の影響を除くと) 高い確率で存在する差分特性経路が存在する場合が少なからずあることは上述したとおりである。それに対し，一時鍵生成部は 4 つの定数乗算を利用するため，入力ハミング重みが多いときほど，逆によりランダム関数に近づくと期待してよかろう。つまり，“副鍵” として差し込まれる値の予測がかなり難しくなることを意味する。一方，入力ハミング重みが少ないとき，一時鍵生成部の特性はもしかすると高い確率で予測できるかもしれない。しかし，そのときには，A3 関数の効果によって，本流部だけでランダム化の影響が大きく出ていると期待される。このように，本流部の A3 関数の効果と一時鍵生成部での “副鍵” の効果では，入力データのハミング重みの大小に関連して，本流部での攻撃者に有利となる特性を少なからず互いに打ち消しあう効果があり，有用なものであると判断する。

2 本検討の最終結論

本稿では，CIPHERUNICORN-A の差分解読法に対する安全性について検証を実施した。その結果，一時鍵生成を完全に除いた，本流部（固定構造の部分のみ）だけでの安全性を見た場合には，16 段における差分特性確率の上界値が 2^{-128} を下回るであろうとの傍証を導いた。ただし，この検討結果は，一部の定数乗算の影響を無視しているという設計者に不利な要素（厳しい評価）と，最良のものを探索した結果ではないという設計者に有利な要素（甘い評価）とが混在しているため，これをもって，安全であるか否かの結論が直接導かれるわけではない。

一方，一時鍵生成部の存在は，ラウンド関数での安全性を向上させる方向へ作用することはおそらく間違いないと考えられる。特に，入力データのハミング重みの大小に関連して，本流部での攻撃者に有利となる特性を少なからず打ち消す効果があり，有用なものであると判断する。

したがって，本流部だけの安全性評価結果と一時鍵生成部での特に “副鍵” の効果を考え合わせれば，最終的に CIPHERUNICORN-A の仕様である 16 段について，高い確率で差分解読法によって攻撃が成功することはなく，安全であると考える。

以上