詳細評価報告書
Camellia 攻撃評価

平成 13 年 1 月 10 日

1 概要

1.1 評価結果概要

Camellia に対して安全性評価を行った。

いくつかの攻撃を試行した範囲では、FL/FL^{-1} 関数を除いた変形 Camellia 8 段が、秘密鍵総当たりより少ない計算量で鍵を推定することが可能であろうという結果になった。

また秘密鍵 5 バイトと中間鍵 6 バイトから、不明な秘密鍵 1 バイトを計算できる場合が存在する。

我々が行った評価の限りでは、安全性に関する問題は見つからなかった。FL/FL^{-1} 関数で使用される鍵値によっては、全体がバイト単位での処理のみになってしまうことからも、より詳細に byte-oriented な評価を続けた方がよいであろう。

1.2 定義

本報告書では以下の記述を用いる。その他、特に注意書きのない限り Camellia の暗号技術仕様書([CAMELLIA_s])の記述に従うものとする。

Δx	: x からの差分値
Prob{a=b}	: a=b の成立確率

2 差分解読、線形解読

Camellia の設計方針や、Camellia の明快な構造からも明らかなように、純粋な差分/線形解読への耐性は十分あると予想できる。

そこで、差分/線形解読に対する評価は行わず、その他の評価を行った。

3 バイト多項式による解析

P 関数は実装効率性の観点から排他的論理和のみで構成され、また差分攻撃や線形攻撃に対する安全性の観点から分岐数が最良となるように設計されている。

P 関数は行列要素が 0 または 1 のみの 8 行 8 列の行列として表現できる。

$$\begin{pmatrix} z'_8 \\ z'_7 \\ z'_6 \\ z'_5 \\ z'_4 \\ z'_3 \\ z'_2 \\ z'_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} z_8 \\ z_7 \\ z_6 \\ z_5 \\ z_4 \\ z_3 \\ z_2 \\ z_1 \end{pmatrix}$$

P 関数の逆行列も同様に表現できる。

$$\begin{pmatrix} z_8 \\ z_7 \\ z_6 \\ z_5 \\ z_4 \\ z_3 \\ z_2 \\ z_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} z'_8 \\ z'_7 \\ z'_6 \\ z'_5 \\ z'_4 \\ z'_3 \\ z'_2 \\ z'_1 \end{pmatrix}$$

P 関数の逆行列より以下の式が成立する。

$z_1 = z'_2 \oplus z'_3 \oplus z'_4 \oplus z'_6 \oplus z'_7 \oplus z'_8$

$z_2 = z'_1 \oplus z'_3 \oplus z'_4 \oplus z'_5 \oplus z'_7 \oplus z'_8$

$z_3 = z'_1 \oplus z'_2 \oplus z'_4 \oplus z'_5 \oplus z'_6 \oplus z'_8$

$z_4 = z'_1 \oplus z'_2 \oplus z'_3 \oplus z'_5 \oplus z'_6 \oplus z'_7$

$z_5 = z'_1 \oplus z'_2 \oplus z'_5 \oplus z'_7 \oplus z'_8$

$z_6 = z'_2 \oplus z'_3 \oplus z'_5 \oplus z'_6 \oplus z'_8$

$z_7 = z'_3 \oplus z'_4 \oplus z'_5 \oplus z'_6 \oplus z'_7$

$z_8 = z'_1 \oplus z'_4 \oplus z'_6 \oplus z'_7 \oplus z'_8$

これらの式より、ラウンド関数の出力 5～6 バイトの排他的論理和はラウンド関数の入力 1 バイトと対応する副鍵 1 バイトで表現できるという性質がある(以降バイト多項式と呼ぶ)。

例えば入力 x_1 の 1 バイトについてみると、

$s_1(x_1 \oplus k_1) = z'_2 \oplus z'_3 \oplus z'_4 \oplus z'_6 \oplus z'_7 \oplus z'_8$

となり、他の入力バイトについても同様に表現できる。

このバイト多項式を利用して、FL 関数と FL^{-1} 関数を除いた変形 Camellia5 段について以下に示す 2 つの手法を適用する。

これらの手法は、バイト多項式を利用した最も単純な攻撃の例である。これを利用して解読に発展できる可能性がある。

3.1 手法1

バイト多項式は排他的論理和のみで構成されているため、差分についても同様なことがいえる。つまり、以下の式が成立する。

$\Delta z_1 = \Delta z'_2 \oplus \Delta z'_3 \oplus \Delta z'_4 \oplus \Delta z'_6 \oplus \Delta z'_7 \oplus \Delta z'_8$

$\Delta z_2 = \Delta z'_1 \oplus \Delta z'_3 \oplus \Delta z'_4 \oplus \Delta z'_5 \oplus \Delta z'_7 \oplus \Delta z'_8$

$\Delta z_3 = \Delta z'_1 \oplus \Delta z'_2 \oplus \Delta z'_4 \oplus \Delta z'_5 \oplus \Delta z'_6 \oplus \Delta z'_8$

$\Delta z_4 = \Delta z'_1 \oplus \Delta z'_2 \oplus \Delta z'_3 \oplus \Delta z'_5 \oplus \Delta z'_6 \oplus \Delta z'_7$

$\Delta z_5 = \Delta z'_1 \oplus \Delta z'_2 \oplus \Delta z'_5 \oplus \Delta z'_7 \oplus \Delta z'_8$

$\Delta z_6 = \Delta z'_2 \oplus \Delta z'_3 \oplus \Delta z'_5 \oplus \Delta z'_6 \oplus \Delta z'_8$

$\Delta z_7 = \Delta z'_3 \oplus \Delta z'_4 \oplus \Delta z'_5 \oplus \Delta z'_6 \oplus \Delta z'_7$

$\Delta z_8 = \Delta z'_1 \oplus \Delta z'_4 \oplus \Delta z'_6 \oplus \Delta z'_7 \oplus \Delta z'_8$

上記 8 つの式のうちどの式を利用しても良いが、攻撃例として最下位バイトの式(1)を使用することにする。

[攻撃に利用した式]

$\Delta z_8 = \Delta z'_1 \oplus \Delta z'_4 \oplus \Delta z'_6 \oplus \Delta z'_7 \oplus \Delta z'_8$ ・・・・・ (1)

攻撃概要図を図 3.1 に示す。

図 3.1 より、5 段で式(2)が確率 1 で成立する。

[5 段で成立する式]

$(\Delta x0_9 \oplus \Delta x0_{12} \oplus \Delta x0_{14} \oplus \Delta x0_{15} \oplus \Delta x0_{16}) \oplus (\Delta z1'_1 \oplus \Delta z1'_4 \oplus \Delta z1'_6 \oplus \Delta z1'_7 \oplus \Delta z1'_8)$

$\oplus (\Delta z3'_1 \oplus \Delta z3'_4 \oplus \Delta z3'_6 \oplus \Delta z3'_7 \oplus \Delta z3'_8) \oplus (\Delta z5'_1 \oplus \Delta z5'_4 \oplus \Delta z5'_6 \oplus \Delta z5'_7 \oplus \Delta z5'_8)$

$= \Delta x5_9 \oplus \Delta x5_{12} \oplus \Delta x5_{14} \oplus \Delta x5_{15} \oplus \Delta x5_{16}$ ・・・・・ (2)

入力のバイト差分として(3)の差分を与える。

$(\Delta x0_1, \Delta x0_2, \Delta x0_3, \Delta x0_4, \Delta x0_5, \Delta x0_6, \Delta x0_7, \Delta x0_8)$,

$(\Delta x0_9, \Delta x0_{10}, \Delta x0_{11}, \Delta x0_{12}, \Delta x0_{13}, \Delta x0_{14}, \Delta x0_{15}, \Delta x0_{16})$

$= (0,0,0,0,0,0,0,0),(0,1,1,0,0,0,0,1)$ ・・・・・ (3)

$\Delta x0_{10}, \Delta x0_{11}, \Delta x0_{16}$ のうち少なくとも 1 バイトに差分があれば良い。

このようなバイト差分を与えると 1 段目ラウンド関数の $\Delta z1'_8$ は 0 となる。さらに式(4)より $z2_2, z2_3, z2_8$ に差分を与えても、2 段目ラウンド関数出力 $\Delta z2'_8$ は 0 となるため、3 段目の入力差分 $\Delta x3_8$ を 0 とすることができる。

$\Delta z2'_8 = \Delta z2_1 \oplus \Delta z2_4 \oplus \Delta z2_5 \oplus \Delta z2_6 \oplus \Delta z2_7$ ・・・・・ (4)

このことから、5 段で成立する式(2)は式(5)のように変形できる。

[手法 1 の式]

$$\Delta s_1(\Delta x5_8) = (\Delta x0_9 \oplus \Delta x0_{12} \oplus \Delta x0_{14} \oplus \Delta x0_{15} \oplus \Delta x0_{16}) \oplus (\Delta x5_9 \oplus \Delta x5_{12} \oplus \Delta x5_{14} \oplus \Delta x5_{15} \oplus \Delta x5_{16})$$ 成立確率 1

・・・・・ (5)

式(5)において、置換表 s_1 の入力差分と出力差分が既知となり、s_1 のすべての入出力差分とそのときの入力対の表(差分表)を用いて、5 段目の副鍵 1 バイト $k5_8$ を 2 組の選択平文でほぼ 100%の確率で求めることができる($\Delta x5_8=0$ の場合は 3 組必要)。

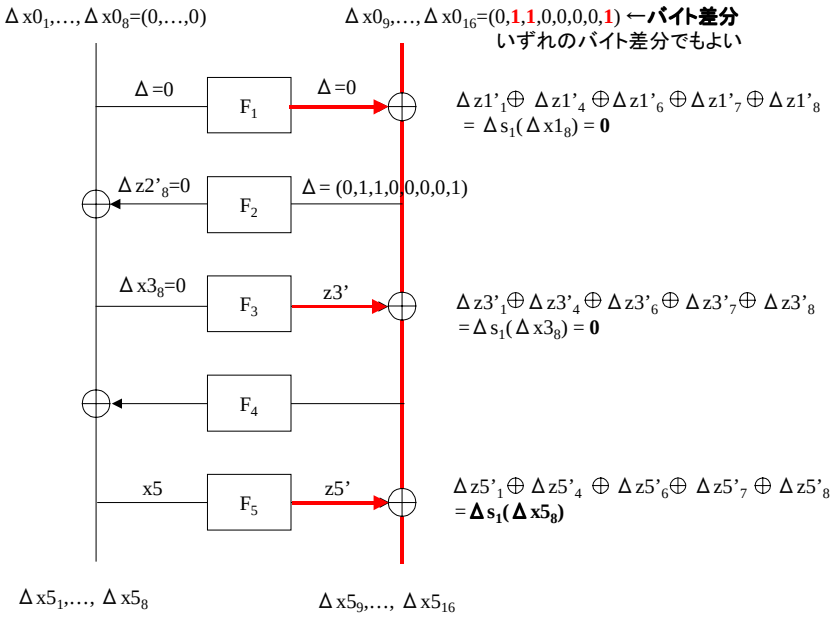

図 3.1 FL 関数と FL^{-1} 関数を除いた変形 Camellia5 段への手法 1 の適用

3.2 手法2

8 つのバイト多項式のうちどの式を利用しても良いが、攻撃例として最下位バイトの式(6)を使用することにする。

[攻撃に利用した式]

$$z_8 = z'_1 \oplus z'_4 \oplus z'_6 \oplus z'_7 \oplus z'_8 \quad \cdots\cdots \quad (6)$$

攻撃概要図を図 3.2 に示す。

図 3.2 より 1,3,5 段目のラウンド関数において、以下の式(7)が確率 1 で成立する。

$$\begin{aligned} s_1(x1_8 \oplus k1_8) &= z1'_1 \oplus z1'_4 \oplus z1'_6 \oplus z1'_7 \oplus z1'_8 \\ s_1(x3_8 \oplus k3_8) &= z3'_1 \oplus z3'_4 \oplus z3'_6 \oplus z3'_7 \oplus z3'_8 \quad \cdots\cdots \quad (7) \\ s_1(x5_8 \oplus k5_8) &= z5'_1 \oplus z5'_4 \oplus z5'_6 \oplus z5'_7 \oplus z5'_8 \end{aligned}$$

5 段で成立する式(8)に(7)を代入すると、式(9)となる。

[5 段で成立する式]

$$\begin{aligned} &(x0_9 \oplus x0_{12} \oplus x0_{14} \oplus x0_{15} \oplus x0_{16}) \oplus (z1'_1 \oplus z1'_4 \oplus z1'_6 \oplus z1'_7 \oplus z1'_8) \\ &\quad \oplus (z3'_1 \oplus z3'_4 \oplus z3'_6 \oplus z3'_7 \oplus z3'_8) \oplus (z5'_1 \oplus z5'_4 \oplus z5'_6 \oplus z5'_7 \oplus z5'_8) \\ &= x5_9 \oplus x5_{12} \oplus x5_{14} \oplus x5_{15} \oplus x5_{16} \quad \cdots\cdots \quad (8) \end{aligned}$$

$$\begin{aligned} &(x0_9 \oplus x0_{12} \oplus x0_{14} \oplus x0_{15} \oplus x0_{16}) \oplus s_1(x1_8 \oplus k1_8) \oplus s_1(x3_8 \oplus k3_8) \oplus s_1(x5_8 \oplus k5_8) \\ &= x5_9 \oplus x5_{12} \oplus x5_{14} \oplus x5_{15} \oplus x5_{16} \quad \cdots\cdots \quad (9) \end{aligned}$$

ここで、「$x1_8$, $x3_8$ = 定数」となるように平文を与えれば、$s_1(x1_8 \oplus k1_8) \oplus s_1(x3_8 \oplus k3_8)$ が定数となり式(10)となる。

[手法 2 の式]

$$\begin{aligned} s_1(x5_8 \oplus k5_8) &= (x0_9 \oplus x0_{12} \oplus x0_{14} \oplus x0_{15} \oplus x0_{16}) \\ &\quad \oplus (x5_9 \oplus x5_{12} \oplus x5_{14} \oplus x5_{15} \oplus x5_{16}) \oplus \text{定数} \quad \text{成立確率 1} \\ &\quad \cdots\cdots \quad (10) \end{aligned}$$

平文に式(11)のようなバイト差分を与えた時に「$x1_8$, $x3_8$ = 定数」となる。

$$\begin{aligned} &(\Delta x0_1, \Delta x0_2, \Delta x0_3, \Delta x0_4, \Delta x0_5, \Delta x0_6, \Delta x0_7, \Delta x0_8), \\ &\quad (\Delta x0_9, \Delta x0_{10}, \Delta x0_{11}, \Delta x0_{12}, \Delta x0_{13}, \Delta x0_{14}, \Delta x0_{15}, \Delta x0_{16}) \\ &= \quad (0,0,0,0,0,0,0,0),(0,1,1,0,0,0,0,1) \quad \cdots\cdots \quad (11) \end{aligned}$$

$\Delta x0_{10}, \Delta x0_{11}, \Delta x0_{16}$ のうち少なくとも 1 バイトに差分があれば良い。

手法 2 の式(10)を使って、1 文目で定数と鍵 $k5_8$ の組が 256 組求まり、2 文目でその 256 組に対して手法 2 の式(10)が成立するかを確認する。成立したときの鍵 $k5_8$ が求めたい鍵である。

これより、5 段目の副鍵 1 バイト $k5_8$ は選択平文 2 文で求めることが可能である(鍵を 1 つに絞り込めないこともあるため、もう 1 文で確認するとほぼ 100%の確率で鍵 $k5_8$ が求まる)。

3.1 節の手法 1 では、1 組の選択平文対(2 文)では 1 バイトの鍵を 1 つに絞り込むことはできないが、手法 2 では選択平文 2 文で鍵を 1 つに絞り込めることがある。よって、手法 2 のほうが若干ではあるが効率が良い。

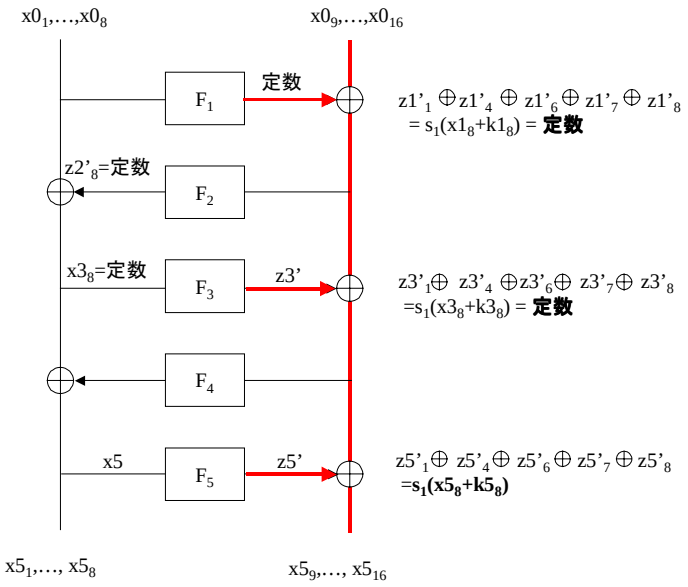

図 3.2 FL 関数と FL^{-1} 関数を除いた変形 Camellia5 段への手法 2 の適用

4 BOOMERANG 攻撃

Camellia に Boomerang 攻撃が適用できるかどうかを検討した。以下に FL 関数と FL^{-1} 関数を除いた変形 Camellia6 段解読のステップを示す。以下の解読法は、k$_2$(F 関数の入力 x$_2$ に排他的論理和する鍵)を推定する例である。差分を変えることにより、k$_1$ から k$_8$ まで推定することが出来る。

<step.0>まず図 4.1 のように入力する選択平文を P,P'とし、それらに対応する暗号文を C,C'とする。また選択暗号文 D,D'に対応する平文を Q,Q'とする。

<step.1> P と P'の差分ΔP0 = 0a000000 0xxxxx00 となるような入力を選ぶ。このとき暗号化の過程で、ラウンド関数 3 段通過後の差分ΔP3 = 0ssssss00 0a000000 がある確率で成立する。ここで a はある任意の値を意味する。また x はラウンド関数 1 段通過後の差分ΔP1 = 00000000 0a000000 となるような値を選ぶ。s は置換表 s$_2$ で入力差分 a を入れた時に、取り得る差分の一つである。

<step.2> C との差分ΔC0 が 0A000000　0XXXXX00 となるような D、また C'との差分ΔC'0 が 0A000000 0YYYYY00 となるような D'を選ぶ。ここで X と Y は 6 段目の

F 関数の k_2 を仮定し、ΔC1 = ΔC'1 = 0A000000 00000000 となるような値を選ぶ。また A はΔP で与える入力差分 a が置換表 s_2 を通過した時に出力差分として取り得るものとし、その中で出現確率が高いものを選ぶことによって Boomerang 攻撃が成立する確率が大きくなる。このとき復号過程で、ラウンド関数 3 段通過後の差分ΔC3 = 0A000000 0SSSSS00、ΔC'3 = 0A000000 0TTTTT00 がそれぞれある確率で成立する。S と T は置換表 s_2 に入力差分 A を入れた時に取りうる差分の一つである。

<step.3>Q と Q'の差分ΔQ を調べる。ΔQ0= 0a000000 0yyyyy00 となっていれば、高い確率で、図 4.1 のような状態になっていると考えられる。ここで y はラウンド関数 1 段通過後の差分ΔQ1 = 00000000 0a000000 となるような値である。このような状態の成立確率 Pb は

$$\begin{aligned} Pb = & Prob\{\Delta P1 = 00000000\ 0a000000\} \\ & \times Prob\{\Delta C1 = \Delta C'1 = 0a000000\ 00000000\} \\ & \times Prob\{\Delta C3 \oplus \Delta C'3 \oplus \Delta P3 = 0ssssss00\ 0a000000\} \\ & \times Prob\{\Delta Q3 \rightarrow \Delta Q1 \mid \Delta Q3 = 0ssssss00\ 0a000000, \\ & \qquad \Delta Q1 = 00000000\ 0a000000\} \\ = & 2^{-7} \times 2^{-7} \times 2^{-1} \times 2^{-6} \times 2^{-7} = 2^{-28} \end{aligned}$$

この値は、偶然にΔQ0 = 0a000000 0yyyyy00 となる場合に比べて非常に大きいので、区別可能である。

<step.4>1 段目と 6 段目について、その入力値がわかり置換表 s_2 について入力差分と出力差分がわかるので、その置換表とセットになっている鍵 8 ビットずつは 2 つの候補に絞れる。残った候補について、ΔP0 の a を代えて同じことを繰り返せば鍵は１つずつに絞ることが出来る。

以上により選択平文/選択暗号文 2^{30} 程度で、FL 関数と FL^{-1} 関数を除いた変形 Camellia6 段に対して Boomerang 攻撃による鍵の推定が可能である。

次に、6 段の Boomerang 攻撃を用いて、8 段に拡張する場合を検討した。1 段暗号化後の差分が 0a000000 0xxxxx00 となるように、1 段目の鍵 5 バイトを仮定し、差分が 0xxxxx00 efghijkl であるような、平文 P と P'を与える。また、暗号文側も同様に 8 段目の鍵 5 バイトを仮定し、1 段復号後の C,C'の差分がそれぞれ 0A000000 0XXXXX00,0A000000 0YYYYY00 となるように暗号文 D と D'を選ぶ。

この時、攻撃が成立する確率は、2^{-108}(= $2^{-40} \times 2^{-40} \times 2^{-28}$) である。つまり、データ量 2^{110} 程度で、1 段目の鍵 5 バイト、2 段目の鍵 1 バイト、7 段目の鍵 1 バイト、8 段目の鍵 5 バイトを推定することが可能であると考えられる。

9 段以上については、Boomerang 攻撃による有効な方法は見つからなかった。

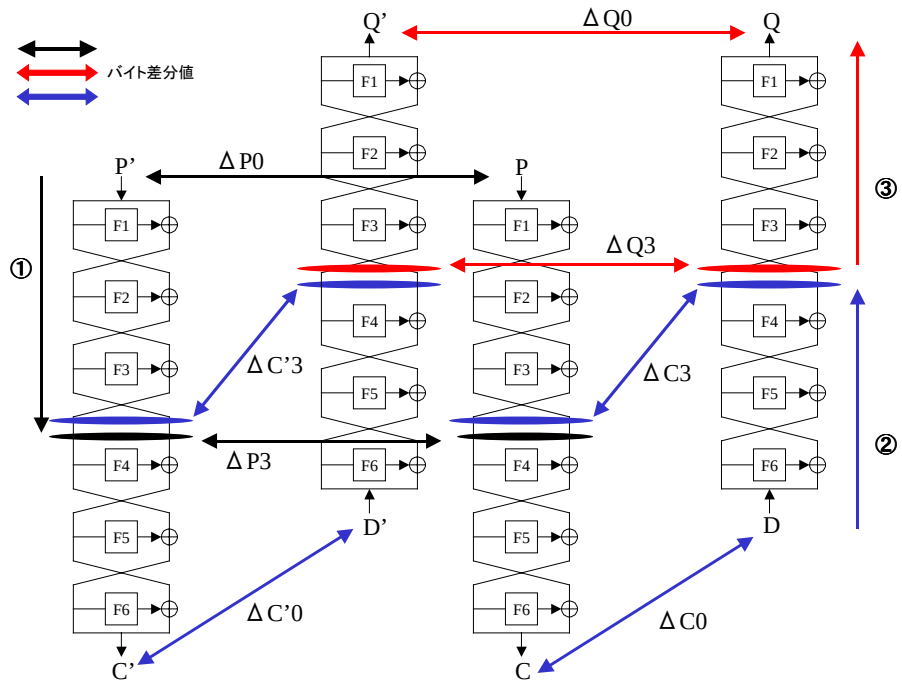

図 4.1 FL 関数と FL^{-1} 関数を除いた変形 Camellia6 段への Boomerang 攻撃の適用

5 SQUARE 攻撃

Camellia のラウンド関数は全単射関数である。全単射な関数は、入力が全通り与えられると、出力の総和は 0 になる(バランスしている)。このことから図 5.1 のような Feistel 型暗号について、次のステップを行うことで 6 段程度の場合、鍵の推定が可能になることを示す。

1. x が取りうる全ての値となるように、P_R を総当たりする。P_L は固定する。x をすべて排他的論理和すると 0 になる。つまり、バランスしている。
2. 全単射な関数 F2 の出力である y の値は全ての値を取る。
3. y'も同様に全ての値を取るので、z も全ての値となっている。
4. x と z はそれぞれに全ての値を取っているので、z'はバランスしている。
5. ここで、F5 と F6 の鍵を仮定し a を求める。
6. 鍵の仮定が正しければ a と C_R を排他的論理和した値もバランスしているはずである。バランスしていなければ、仮定した鍵は棄却できる。

以上により全単射なラウンド関数を持つ Feistel 型暗号は 6 段で鍵の推定が可能であることがわかる。しかし、ステップ 5 での鍵の仮定量が秘密鍵長よりも大きければ有効な解読手法とは言えない。

ここで Camellia のラウンド関数を考えてみる。Camellia のラウンド関数の場合、P 関数の仕様によりステップ 1 の総当りをあるバイトについてのみ行う。つまり、256 通りの値を与える。ステップ 2 で影響が波及するバイトは 5〜6 バイトしかない。ステップ 3 で初めて全バイトに影響が波及することになるが、各バイトにおいてはそれぞれバランスしている。ステップ 6 でチェックを 1 バイトについてのみ行うとすれば、ステップ 5 で仮定が必要になる鍵量は最小の場合 13 バイトである。

これは Camellia の秘密鍵長 128 ビットよりも短いため、この解析手法は Camellia のラウンド関数に置き換えても有効であると言えるだろう。

この場合、鍵候補として残る数を約 $1/2^8$ と見積もると、鍵の推定のためには 2^{104} 程度の平文対が必要になるため、FL 関数と FL^{-1} 関数を除いた変形 Camellia6 段は選択平文 2^{104}、計算量 2^{112} 程度で副鍵の一部が推定可能となるだろう。

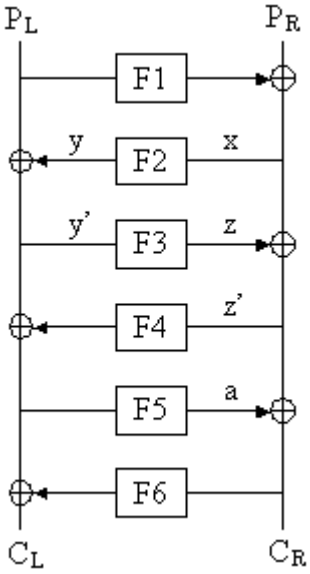

図 5.1 Feistel 型暗号

6 丸め差分攻撃

丸め差分攻撃はバイト単位で処理する暗号に適用される攻撃法である。Camellia は E2 と同様にラウンド関数はバイト単位で処理されている。E2 は[MT99]で 7 段のバイト差分特性が発見されている。Camellia は E2 からの改善を図っていると思われるが、変更点の 1 つである置換表に対する「4 つの異なる置換表を作ることによって、丸め差分攻撃[MT99]に対する安全性が多少改善される」という自己評価書 2.3 節の記述に関して検討した。

[MT99]では以下の特性確率 $P(\Delta x, \Delta y)$において、

$$P(\Delta x, \Delta y) = \mathrm{Prob}\{\, s(x) \oplus s(x \oplus \Delta x) = s(y) \oplus s(y \oplus \Delta y) \,\}$$

$\Delta x \neq \Delta y$ の場合には、平均 2^{-8} となるが、$\Delta x = \Delta y$ の場合には、大きな値になるという特性を用いている。s_1 と s_4 は入力を巡回シフトしている関係にあるので、本特性は $\Delta x = (\Delta y <<< 1)$ の場合に成立する。このことにより、s_4 は丸め差分攻撃[MT99]に対する安全性改善には寄与していないことになる。s_2 および s_3 は、s_1 の出力を 1 ビット左および右巡回シフトした関係にある。これは、[MT99]の第 4 節で用いられている事実($\Delta x = \Delta y$ の条件のもとでは、 $P(\Delta x, \Delta y)$は 2^{-8} よりも大きくなる)には該当せず、丸め差分攻撃に対する安全性改善に寄与していると言える。

丸め差分解読に対する強度は P 層に依存するが、もし丸め差分解読に対する安全性改善を置換表のみに頼るのであれば、s_4 の代わりに $s_1(x) <<< 2$ を用いることも選択肢の一つであろう。

7 高階差分攻撃

高階差分攻撃は暗号器を平文全ビットのブール式で表現した場合、代数次数が低い場合やブール式の項が疎な場合、解読が可能になることがある。

Camellia のラウンド関数において、代数次数を上げる働きをするものは置換表のみである。そこで、置換表 s_1 について、ブール式で表現したときの次数と項数の調査を実施した。結果を表 7.1 に示す。

また、置換表 s_2,s_3 は s_1 の出力を単純に 1 ビット巡回シフトしたものであり、また、置換表 s_4 は s_1 の入力を 1 ビット巡回シフトしたものであるため、ブール式としての次数と項数は同じである。

表 7.1 より、4 種の置換表の最高次数は全出力ビット 7 次であり、項数も取り得る項数の半分程度存在する(疎ではない)ことが確認できた。

さらに、次数や項数は置換表を通過するたびにはじめは急激に増加すると期待できる。例えば、2 回の置換表を通過すると次数は 7^2、3 回通過すると次数は 7^3 になる。よって、Camellia は高階差分攻撃に対して安全であると期待できる。

表 7.1 置換表 s_1 の次数と項数

ビット番号	1 次	2 次	3 次	4 次	5 次	6 次	7 次	8 次
1	6	11	23	38	27	18	4	0
2	7	16	24	39	28	9	8	0
3	3	17	28	35	28	11	3	0
4	7	13	27	39	33	11	4	0
5	3	17	26	36	32	12	3	0
6	3	14	36	34	26	15	5	0
7	6	14	29	36	23	14	7	0
8	3	15	31	36	23	15	3	0

取り得る項数を表 7.2 に示す。

表 7.2 取り得る項数

1 次	2 次	3 次	4 次	5 次	6 次	7 次	8 次
8	28	56	70	56	28	8	0

8 補間攻撃

補間攻撃はラウンド関数などの暗号化関数が適当な代数系の上で単純な式で表現できるときに有効な攻撃法である。Camellia のラウンド関数は副鍵の排他的論理和と置換表を利用した S 関数、排他的論理和のみの P 関数により構成されている。また、ラウンド関数はバイト単位でのみ処理されている。

そこで、まず適当な代数系としてガロア体 $GF(2^8)$を選択し、置換表が単純な式で表現できるか検討した。

すべての 8 次の既約多項式(30 式)から構成したガロア体 $GF(2^8)$上で、4 種の置換表の項数と最高次数を調査した。結果を表 8.1 に示す。取り得る最大項数は 255、最大次数は 254 である。

補間攻撃が可能になるには、置換表の最高次数が低い場合や項数が少ない場合である。表 8.1 から明らかなように Camellia の置換表はガロア体 $GF(2^8)$上での式が複雑であり、ガロア体 $GF(2^8)$上での補間攻撃に対しては安全であると思われる。

表 8.1 置換表の $GF(2^8)$上での多項式表現

置換表	項数	最高次数
s_1	250～255	すべて 254
s_2	251～255	すべて 254
s_3	252～255	すべて 254
s_4	252～255	すべて 254

置換表はハードウェアの小型設計が可能となるように $GF(2^8)$上の要素は部分体 $GF(2^4)$上の多項式としても表現可能なように設計されている。そこで次に部分体 $GF(2^4)$上でも検討した。部分体 $GF(2^4)$を構成するための既約多項式は、g 関数で使用している x^4+x+1 を使用した。置換表 s_1 は以下の式になる。

$$s_1(x) = h(g(f(0xc5 \oplus x))) \oplus 0x6e$$

[f 関数] Y = f(X)

$X = x_0 \| x_1$, $Y = y_0 \| y_1$ とすると、

$y_0 = b \cdot x_0 \oplus 7 \cdot x_0^2 \oplus c \cdot x_0^8 \oplus a \cdot x_1 \oplus c \cdot x_1^2 \oplus f \cdot x_1^4 \oplus a \cdot x_1^8$

$y_1 = 7 \cdot x_0 \oplus e \cdot x_0^2 \oplus 9 \cdot x_0^4 \oplus 9 \cdot x_0^8 \oplus x_1 \oplus 2 \cdot x_1^2 \oplus 6 \cdot x_1^4 \oplus 7 \cdot x_1^8$ (係数は 16 進)

[g 関数] $Y = g(X)$

$X = x_0 \| x_1$, $Y = y_0 \| y_1$ とすると、

$y_0 = (x_0 \oplus f_1 \cdot x_1) / (x_0(x_0 \oplus f_1 \cdot x_1) \oplus f_0 \cdot x_1^2)$

$y_1 = (x_1) / (x_0(x_0 \oplus f_1 \cdot x_1) \oplus f_0 \cdot x_1^2)$ (ここで $f_0=9$, $f_1=1$)

[h 関数] $Y = h(X)$

$X = x_0 \| x_1$, $Y = y_0 \| y_1$ とすると、

$y_0 = 8 \cdot x_0 \oplus 8 \cdot x_0^2 \oplus b \cdot x_0^4 \oplus 9 \cdot x_0^8 \oplus 2 \cdot x_1 \oplus 6 \cdot x_1^2 \oplus b \cdot x_1^4 \oplus e \cdot x_1^8$

$y_1 = 4 \cdot x_0 \oplus e \cdot x_0^2 \oplus d \cdot x_0^4 \oplus 7 \cdot x_0^8 \oplus 4 \cdot x_1 \oplus c \cdot x_1^2 \oplus 2 \cdot x_1^4 \oplus e \cdot x_1^8$ (係数は 16 進)

これより、g 関数の y_0, y_1 の分母が同じ式をしていることがわかる。有理式による補間攻撃を考えた場合、分母が共通であることがやや気にはなるが、2 回のアフィン変換を g 関数の前後で実行することにより、部分体 GF(2^4)上での有理式が複雑になり、部分体 GF(2^4)上での有理式を用いた補間攻撃に対しては安全になっていると思われる。

また、入出力を GF(2^4)の元 2 個とみなし、出力が入力のどのような多項式として表されるかを調査した。結果を表 8.2 に示す。

すべての 4 次の既約多項式(3 式)から構成したガロア体 GF(2^4)上での 4 種の置換表の項数は以下の通りである。取り得る最大項数は 255 である。出力=$y_0 \| y_1$ とする。

表 8.2 より明らかなように Camellia の置換表は部分体 GF(2^4)上での式が複雑であり、部分体 GF(2^4)上での補間攻撃に対しては安全であると思われる。

表 8.2 置換表の部分体 GF(2^4)上での多項式表現

置換表	y_0 の項数	y_1 の項数
s_1	237～245	238～245
s_2	234～236	238～245
s_3	237～238	232～246
s_4	241～243	235～242

9 鍵生成部の特性

Camellia の鍵生成部には以下のような特性がある。第 3 章で述べたように、ラウンド関数の入出力間には

$$s_1(x_8 \oplus k_8) = z_1 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8$$

という関係式が成り立つ。鍵生成部の場合、ラウンド関数での副鍵の挿入に相当するものとして定数Σが挿入されているので、上記関係式は

$$s_1(x_8) = z_1 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8$$

と書き換えられる。

また秘密鍵 128 ビットの場合の鍵生成部は図 9.1 のように等価変形可能である。そこで、KA_R の 8 バイト目と KA_L の 1,4,6,7,8 バイト目が求まっているとすると、F4 の出力 5 バイトの排他的論理和($z_1 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8$)が求まる。よって、F2 の入力 1 バイトも求まる。

さらに、K_{LL} の 5 バイトが求まっていると、1 段目の F の出力 1 バイトを計算できるので、K_{LR} の 1 バイトが求まる。

これはラウンド関数の入力バイトいずれに対しても言える特性である。すなわち、KA の 6,7 バイトと K_{LL} の 5,6 バイトから K_{LR} の 1 バイトを求めることができる。しかし、この特性は Camellia の鍵生成部を解読可能にするような特性ではないだろう。

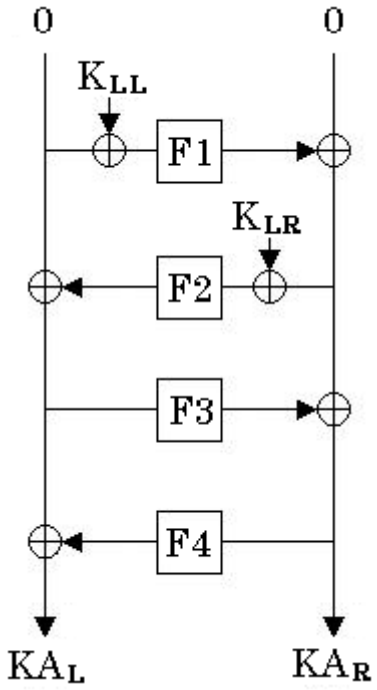

図 9.1 Camellia 鍵生成部(秘密鍵長 128 ビット)等価変換

10 弱鍵

Camellia に弱鍵が存在するかどうかを検討してみた。鍵生成部の構造上、秘密鍵を副鍵としてそのまま使用している段があるため全ての段が同じ鍵となるような等価鍵は明らかに存在しない。また副鍵に使用する場合のローテイト数も各段で異なるため、スライド攻撃や鍵関連攻撃に利用できるような繰り返す鍵も存在しないであろう。

11 その他

S 関数において、$(s_1,s_2,s_3,s_4,s_2,s_3,s_4,s_1)$という置換表の並びを選択した基準が不明である。

12 まとめ

今回 Camellia を評価した範囲では、安全性に関して特に問題はみつからなかった。

いくつかの攻撃を試行した範囲では、FL/FL^{-1} 関数を除いた変形 Camellia 8 段が、秘密鍵総当たりより少ない計算量で鍵を推定することが可能であろうという結果になった。仕様通りの Camellia では、18 段(鍵長 128 ビット)もしくは 24 段(鍵長 192/256 ビット)なので、直接脅威となることはないと考えられる。また鍵生成部の構造より、秘密鍵 5 バイトと中間鍵 6 バイトから、不明な秘密鍵 1 バイトを計算できる場合が存在する。

ラウンド関数がバイト単位の処理のみで構成されており、FL/FL-1 関数で使用される鍵によっては、全体がバイト単位での処理のみになってしまうところが少々気になる。構造的に、より詳細に byte-oriented な評価を続けた方がよいであろう。

ラウンド関数が簡単な構造なので、ラウンド関数を推定する際に必要な鍵量が少ない点が解読者には有利である。

全体的には、評価した範囲では Camellia は安全であるという結論に達した。

A 参考文献

[CAMELLIA_s] http://info.isl.ntt.co.jp/camellia/index-j.html 暗号技術仕様書[和文].

[CAMELLIA_e] http://info.isl.ntt.co.jp/camellia/index-j.html 自己評価書[和文].

[A00] K.Aoki, "Practical Evaluation of Security against Generalized Interpolation Attack," IEICE Vol.E83-A,No1,pp.33-38, 2000.

[BW00] A.Biryukov,D.Wagner, "Advanced Slide Attacks," EUROCRYPT2000, LNCS1807, pp.589-606, Springer-Verlag, 2000.

[CV95] F. Chabaud and S. Vaudenay, "Links Between Differential and Linear Cryptanalysis," EUROCRYPT'94, LNCS950, pp.356-365, Springer-Verlag, 1995.

[DKR97] J.Daemen, L.Knudsen, V.Rijmen, "The Block Cipher SQUARE," FSE'97, LNCS 1267, pp.149-165, 1997.

[K00] http://info.isl.ntt.co.jp/camellia/index-j.html Knudsen の評価報告書 (Analysis of Camellia).

[M94] M.Matsui, "On Correlation Between the Order of S-boxes and the Strength of DES," EUROCRYPT'94, LNCS950, pp.366-375, 1994.

[MSAK99] S.Moriai,M.Sugita,K.Aoki,M.Kanda, "Security of E2 against Truncated Differential Cryptanalysis," SAC'99 pp.106-117, 2000.

[MT99] M.Matsui,T.Tokita, "Cryptanalysis of a Reduced Version of the Block Cipher E2," FSE'99 LNCS 1636,pp.71-80.

[W99] D.Wagner, "The Boomerang Attack," FSE'99, LNCS 1636, pp.156-170, 1999.

[Y00] http://info.isl.ntt.co.jp/camellia/index-j.html Yiqun の評価報告書 (A Note on the Block Cipher Camellia).