軽量型ブロック暗号 TWINE の高階差分特性

小菅　悠久 †　　岩井　啓輔 †　　田中　秀磨 †　　黒川　恭一 †

† 防衛大学校
239-8686 神奈川県横須賀市走水 1-10-20
{em53036,iwai,hidema,kuro}@nda.ac.jp

あらまし　TWINE は 2011 年に NEC によって提案された軽量型ブロック暗号である．一般化フェイステル構造を採用した 36 段構成の 64 ビットブロック暗号であり，鍵長は 80 ビット及び 128 ビットを選択できる．本稿では，TWINE が 4 ビット S-box を採用していることに注目し，4〜36 階までの高階差分特性を 4 ビット処理単位で網羅的に調査した結果を示す．その結果，12 段目の出力の一部において 24 階差分値が 0 となることを発見した．128 ビット鍵長を仮定した場合，この性質を利用した多段消去型攻撃の適用により 19 段まで攻撃可能である．

High-Order Differential characteristic of Light-weight block cipher TWINE

Haruhisa Kosuge†　　Keisuke Iwai †　　Hidema Tanaka†　　Takakazu Kurokawa†

†National Deffence Academy.
Hashirimizu 1-10-20 Yokosuka-shi, Kanagawa-Pref Japan 239-8686, JAPAN
{em53036,iwai,hidema,kuro}@nda.ac.jp

Abstract　TWINE is a light-weight block cipher proposed by NEC in 2011. This cipher is 64-bit block cipher which adopts the generalized Feistel structure with 36 rounds and there exists 80-bit and 128-bit key length versions. Focusing on 4-bit S-box, we analyze the high-order differential characteristic of TWINE using 4th〜36th differential in 4-bit oriented manner. As a result we found that 24th order differential makes part of an output of 12 round be 0 and by using the multi rounds elimination method up to 19 round can be attacked under the 128-bit key length condition.

1　はじめに

近年の通信機器の小型化，高性能化及び多様な分野での普及に伴い，小規模でリソースの限られているハードウェアへの実装に適した軽量型ブロック暗号の開発に注目が集まっており，ISO/IEC 29192-2 においてその標準化が行われている [1]．ここでは，安全性要件及び実装要件を満たす軽量型ブロック暗号として PRESENT[2] と CLEFIA[3] が採択されている．また，その他にも DESL[4], HIGHT[5], PRINTcipher[6] 等が開発されている．

本稿で注目する TWINE は，先に述べたニーズに応じるために安全性及び高速性を担保しつつ，小規模なハードウェアや組み込みソフトウェア等の多様な環境への実装が可能なように設計されている [7]．また，NEC が開発した認証付暗号利用モードである CLOC では，使用するブロック暗号として AES 及び TWINE が推奨

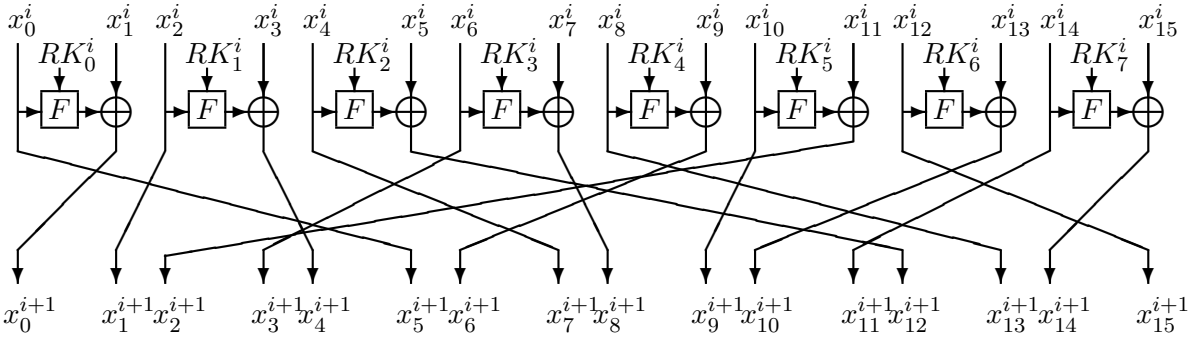

図 1: TWINE の段関数

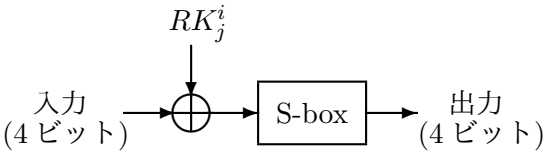

図 2: TWINE の F 関数

されている [8].

TWINE の安全性については，開発者らによる不能差分攻撃及び飽和攻撃に対する評価がなされている．また，Çoban らによる Biclique 攻撃 [9]，Boztaş らによる中間一致攻撃 [10]，Bogdanov らによる関連鍵攻撃 [11] 等の安全性評価がなされている．本稿では，TWINE が 4 ビット S-box を採用していることに注目し，4〜36 階までの高階差分特性を 4 ビット処理単位で網羅的に調査した結果を示す．その結果，24 階差分で 12 段の出力の一部が 0 となることを発見し，128 ビット鍵長において多段消去型攻撃の適用により 19 段まで攻撃可能であることが分かった．

2 TWINE

2.1 構造

TWINE は Type-2 一般化フェイステル構造 [12] 採用し，鍵長に関わらず F 関数及び拡散層から成る段関数を 36 段有する（最終段は拡散層を持たない）．図 1 に各段の段関数を，図 2 に F 関数の構造をそれぞれ示す．段関数では入力を 4 ビットずつの 16 ブロックに分割し，偶数ブロックの値を F 関数に入力し，その出力と隣接した奇数ブロックの値を XOR した後，通常の Type-2 一般化フェイステル構造とは異なる拡散層でブロック単位での置換を行う．F 関数は 4 ビット入力 4 ビット出力の S-box を使用した全単射関数である．復号については，鍵を逆の順番で F 関数に入力することを除けば暗号化と同一のアルゴリズムで行えるため，これが高速性及び軽量性の根拠となっている．

鍵スケジュール部はビット単位での置換を行わない等，構造が単純であることから，鍵の生成を高速で実行できる長所を持つが，短所として副鍵の間に関連性が発生するという脆弱性が存在する．その為，一部の副鍵の値を推定した際，他の副鍵の中に一意的に値が決まるものが存在する．この脆弱性について，開発者は中間一致攻撃や関連鍵攻撃等への耐性は十分備えていると述べている [7].

2.2 安全性評価

開発者による安全性評価では，不能差分攻撃と飽和攻撃が最も強力な攻撃法とされている．不能差分攻撃を鍵長 128 ビットの場合に適用すると，選択平文数 $2^{52.21}$ により 24 段構成が攻撃可能である．同様に鍵長 80 ビットの場合は選択平文数 $2^{61.39}$ により 23 段構成まで攻撃可能なことが示されている．また飽和攻撃を適用した場合は，鍵長 128 ビットに対し選択平文数 $2^{62.81}$ により 23 段構成が，鍵長 80 ビットに対し 2^{62} 個の選択平文により 22 段構成が攻撃可能であることが示されている．また差分／線形攻撃による評価は，S-box の最大確率が 2^{-4} であることを示し，15 段の最小 Active S-box 数が 32 個であることから，16 段構成以上に対し攻撃が不可能であると結論している [7].

開発者以外でも，複数の研究者が TWINE の安全性評価を行っている．Çoban らは Biclique

攻撃で全段の攻撃が可能であり，その際に必要な選択平文数が2^{60}であるとしている [9]．Boztaşらは中間一致攻撃で 25 段構成までの攻撃が可能であり，その際に必要な選択平文数が2^{48}であるとしている [10]．Bogdanov らは関連鍵攻撃で 27 段構成までの攻撃が可能であり，その際に必要な選択平文数が$2^{62.29}$であるとしている [11]

3 高階差分攻撃

3.1 高階差分

$F(X;K)$ を，$GF(2)^n \times GF(2)^s \mapsto GF(2)^n$ の関数とする．これを暗号化関数と考えれば，入力 X 及び出力 Y が n ビット，鍵 K が s ビットであり，以下のように記述できる．

$$Y = F(X;K) \tag{1}$$

また，$(\alpha_0, \alpha_1 .... \alpha_{N-1})$ を $GF(2)^n$ 上で線形独立な N 個のベクトルとし，これらによって張られる部分空間を $V_{[\alpha_0,\alpha_1 .... \alpha_{N-1}]}$ と表す．関数 $F(X;K)$ に対する $(\alpha_0, \alpha_1 .... \alpha_{N-1})$ を用いた N 階差分値は以下のように計算できる．ただし鍵 K は定数である．

$$\Delta^N_{V_{[\alpha_0,\alpha_1 .... \alpha_{N-1}]}} Y = \bigoplus_{A \in V_{[\alpha_0,\alpha_1 .... \alpha_{N-1}]}} F(X \oplus A;K) \tag{2}$$

以下では，$V_{[\alpha_0,\alpha_1 .... \alpha_{N-1}]}$ は平文の入力差分を表すため，誤解の恐れがない場合は記述を省略し，Y の N 階差分値を $\Delta^N Y$ と略記する．Y の N 階差分特性は，$F(X;K)$ の入力 X に対する代数次数で決定される．もし $F(X;K)$ の X に対する代数次数が d であるならば，以下の性質が成立する．

$$deg_x\{F(X;K)\} = d \rightarrow \begin{cases} \Delta^{d+1} Y = 0 \\ \Delta^d Y = const \end{cases} \tag{3}$$

なお，$const$ の値は定数である鍵 K の値と平文の固定値によって決定される．

3.2 攻撃方程式の導出

攻撃方程式の例として，r 段構成の DES 型フェイステル構造の暗号化関数を仮定する．暗号文 x^r と最終段の副鍵 K^r を入力とする $G(x^r;K^r)$ をこの暗号化関数の最終段の段関数に対する逆関数とする．$r-1$ 段の N 階差分値が定数となる場合，r 段出力の N 階差分値を逆算するための暗号文 x^r の集合を $\mathbf{C}$ とすると，最終段の副鍵 K^r に対し，以下が成立する．

$$\bigoplus_{x^r \in \mathbf{C}} G(x^r;K^r) = const \tag{4}$$

従って未知の副鍵 K^r に対して上式は K^r を求めるための攻撃方程式とみなせる．

副鍵 K^r が取り得るすべての値に対し，式 (4) の成立不成立を調査する．この時，式 (4) を成立させる副鍵 K^r は正しい値の候補となる．以上より高階差分を用いた攻撃方程式を導出するためには，事前に出力の高階差分値が定数となること及びそれを満たす選択平文が既知であることが条件となる．

式 (4) から，TWINE における攻撃方程式の導出について示す．TWINE は変形 Type-2 一般化フェイステル構造を持つことから，ここでは奇数番目の出力ブロックの高階差分値に注目する．今，r 段構成の TWINE を仮定し，$r-1$ 段目の出力の N 階差分値が全ての出力ブロックで 0 となったとする．TWINE の構造から，$(x^r_{o-1}, x^r_o)(o = 1, 3, 5, ..., 15)$ のようにペアとなった暗号文組を用いて，以下の攻撃方程式が導出できる．

$$\bigoplus_{x^r \in \mathbf{C}} F(x^r_{o-1}; RK^r_{\lfloor o/2 \rfloor}) \oplus x^r_o = 0 \tag{5}$$

ただし，$\lfloor i \rfloor$ はガウスの床関数とする．

3.3 飽和攻撃

飽和攻撃は，Daemon らによってブロック暗号 SQUARE に対する攻撃手法として提案された選択平文攻撃であり，代数的性質に注目する攻撃法である [13]．暗号化関数の各段における入力及び出力が n ビット単位の m 個のブロック

で構成されていると仮定する．各ブロックの値 X の集合 $\{X_i|Xi \in \{0,1\}^n, 0 \leq i \leq 2^n\}$ は，以下の 4 つの状態に分類できる．

$$Constant(C) \ : \ \forall i,j \ X_i = X_j \quad (6)$$

$$All(A) \ : \ \forall i,j \ i \neq j \Leftrightarrow X_i \neq X_j \quad (7)$$

$$Balance(B) \ : \ \bigoplus_i X_i = 0 \quad (8)$$

$$Unknown(U) \ : \ \text{その他} \quad (9)$$

状態 A に注目する攻撃手法は，SQUARE 攻撃と同等である．もし，a 個の入力ブロックに対し，性質 A を適用するならば，$n \times a$ 階差分を利用した高階差分攻撃と同等に見なせる．もしブロックの状態が U 以外であった場合，その $n \times a$ 階差分値は常に 0 となり，U の場合は未知数となる．1 段目への入力値及び r 段における出力値の状態の各ブロックの状態が，常に同様の関連性を有するとき，この暗号化関数は r 段目で飽和特性を有し，$\{S_0^0, S_1^0, ..., S_{m-1}^0\} \rightarrow^r \{S_0^r, S_1^r, ..., S_{m-1}^r\}$ と表現する．

TWINE の開発者は，上記 4 つの状態を用いて理論的に飽和特性を探索した結果，以下の二つの飽和特性が 15 段において存在すると述べている [7]．なお，複数の連続した i 個のブロックの状態 S が同一である場合は S^i と略記する．

$$\{A^{12}, C, A^3\} \rightarrow^{15} \{U^3, B, U^5, B, U^3, B, U, B\}$$
$$\{A^6, C, A^9\} \rightarrow^{15} \{U, B, U^3, B, U, B, U^3, B, U^4\}$$

これらの飽和特性は，15 段の 60 階差分値の一部が 0 となることを意味している．

開発者らは，上記のように飽和特性のルールに基づいて代数的攻撃手法に対する安全性評価を実施している．この手法は，理論的実験に基づいている．一方，本稿では代数的攻撃に対する安全性を計算機実験により実施することを目的とする．

4 計算機実験

4.1 実験の概要

第 2.1 節で示したように，TWINE の F 関数は 4 ビット入力 4 ビット出力であるから，取り得る高階差分の入力組み合わせに対し網羅的に計算機実験を行う．F 関数が 4 ビット入出力の全単射関数であることから，その代数次数は 3 次である．従って，3 階差分及び 4 階差分を利用して計算機実験を行った．計算機実験環境を表 1 に示す．また使用する変数を表 2 のように定義する．観測する出力の高階差分値は 0 もしくは定数であり，それぞれ非ゼロもしくは別の値に変化した時に出力の代数次数が飽和したと判断する．このときの段数を評価することで，TWINE の高階差分特性を調査した．平文の固定値及び秘密鍵はランダムに与え，高階差分を計算する変数部分のみ制御した．平文固定値及び副鍵は，最低 4 通り用いて実験を行った．

表 1: 計算機実験環境

OS	Red Hat Enterprise Linux 5
CPU	Intel Xenon X5680 3.33GHz
メモリ	10GB
コンパイラ	gcc 4.1.2

表 2: 計算結果の表記に用いる変数

$X_{\{a_0,a_1,...,a_{n-1}\}}$	$a_0, a_1, ..., a_{n-1}$ で示す入力のブロックが 3 階差分又は 4 階差分であること示す．
$Y_{0\{b_0,b_1,...,b_{n-1}\}}$	$b_0, b_1, ..., b_{n-1}$ で示す出力のブロックの高階差分値が 0 になること示す．
$Y_{c\{b_0,b_1,...,b_{n-1}\}}$	$b_0, b_1, ..., b_{n-1}$ で示す出力のブロックの高階差分値が定数になること示す．
$X \rightarrow Y$	ある入力差分の取り方に対し，出力のブロックの高階差分値が 0 又は定数となることが常に成立することを示す．

4.2 実験結果

前述のように，TWINE の F 関数は 4 ビット入出力であるため，平文固定値及び副鍵 (それぞれ 4 ビット) に対する全通りの場合について

表 3: 3 階差分及び 6 階差分の計算結果

階数	段数	入力差分の取り方と高階差分値の関係
3	7	$X_{\{1\}} \rightarrow Y_{0\{11,13\}}Y_{c\{3\}}$, $X_{\{3\}} \rightarrow Y_{0\{1,9\}}Y_{c\{7\}}$, $X_{\{5\}} \rightarrow Y_{0\{7,15\}}Y_{c\{1\}}$, $X_{\{7\}} \rightarrow Y_{0\{11,13\}}Y_{c\{1\}}$, $X_{\{9\}} \rightarrow Y_{0\{3,5\}}Y_{c\{11\}}$, $X_{\{11\}} \rightarrow Y_{0\{1,9\}}Y_{c\{15\}}$, $X_{\{13\}} \rightarrow Y_{0\{7,15\}}Y_{c\{9\}}$, $X_{\{15\}} \rightarrow Y_{0\{3,5\}}Y_{c\{13\}}$,
6	9	$X_{\{1,11\}} \rightarrow Y_{0\{1,3\}}$, $X_{\{3,7\}} \rightarrow Y_{0\{5,9\}}$, $X_{\{5,15\}} \rightarrow Y_{0\{13,15\}}$, $X_{\{9,13\}} \rightarrow Y_{0\{7,11\}}$

表 4: 4 階〜24 階差分の計算結果

階数	段数	入力差分の取り方と高階差分値の関係
4	9	$X_{\{1\}}, X_{\{5\}}, X_{\{11\}}, X_{\{15\}} \rightarrow Y_{0\{1,3,13,15\}}$, $X_{\{3\}}, X_{\{7\}}, X_{\{9\}}, X_{\{13\}} \rightarrow Y_{0\{5,7,9,11\}}$
8	10	$X_{\{0,1\}}, X_{\{2,3\}}, X_{\{12,13\}}, X_{\{14,15\}} \rightarrow Y_{0\{1,3,13,15\}}$, $X_{\{4,5\}}, X_{\{6,7\}}, X_{\{8,9\}}, X_{\{10,14\}} \rightarrow Y_{0\{5,7,9,11\}}$
12	11	$X_{\{1,2,3,\}}, X_{\{5,8,9\}}, X_{\{6,7,11\}}, X_{\{12,13,15\}} \rightarrow Y_{0\{1,3,13,15\}}$, $X_{\{0,1,3\}}, X_{\{4,5,9\}}, X_{\{7,10,11\}}, X_{\{13,14,15\}} \rightarrow Y_{0\{5,7,9,11\}}$
16	11	$X_{\{0,1,2,3\}}, X_{\{4,5,8,9\}}, X_{\{6,7,10,11\}}, X_{\{12,13,14,15\}} \rightarrow Y_{0\{1,3,5,7,9,11,13,15\}}$
20	11	$X_{\{0,1,3,10,11\}}, X_{\{1,2,3,6,7\}}, X_{\{4,5,13,14,15\}}, X_{\{8,9,12,13,15\}} \rightarrow Y_{0\{0,1,3,4,5,7,9,10,11,13,15\}}$, $X_{\{0,1,7,10,11\}}, X_{\{2,3,6,7,11\}}, X_{\{4,5,9,14,15\}}, X_{\{5,8,9,12,13\}} \rightarrow Y_{0\{1,2,3,5,6,7,8,9,11,12,13,15\}}$
24	12	$X_{\{0,1,2,3,6,7\}}, X_{\{0,1,2,3,10,11\}}, X_{\{2,3,12,13,14,15\}}, X_{\{8,9,12,13,14,15\}} \rightarrow Y_{0\{1,3,13,15\}}$, $X_{\{0,1,6,7,10,11\}}, X_{\{2,3,6,7,10,11\}}, X_{\{4,5,8,9,12,13\}}, X_{\{4,5,12,13,14,15\}} \rightarrow Y_{0\{5,7,9,11\}}$

4 通りの 3 階差分を計算した．その結果，平文固定値及び副鍵に依存せず，3 階差分を算出するためのビット位置の選択によって定数値が決定されることが分かった．この結果から，3 階差分特性では出力が 3 次以下を保証する 3 階差分値を決定し，その値が出力段数を経ても維持されることで評価した．前節で示したように，3 階差分を基本として 15 差分まで 3 階単位で網羅的に高階差分特性を調査した．その結果を表 3 に示す．なお，表 3 では紙面の制限のため，例として 3 階差分及び 6 階差分の結果のみ示す．なお，3 階差分では，7 段目まで出力の高階差分値は 0 又は定数となり，3 次以下であることが分かった．6 階差分以上では，高階差分値が定数となる場合が観測されなかった．6 階差分の場合，出力が 6 次以下となるのは 9 段までである．9 階差分，12 階差分，15 階差分では，高階差分値が 0 となるブロックが増加するものの，6 階差分と同様に 10 段以降では飽和することが分かった．

TWINE の F 関数は，4 ビット入出力であるため，4 階差分値は平文固定値及び副鍵に依存せず常に 0 となる．この性質より，段数が増加しても代数次数が飽和しない限り高階差分値は 0 となる．4 階差分を基本として，4 階差分から 36 階差分まで 4 階単位で網羅的に高階差分特性を調査した結果を表 4 に示す．表 4 には 4〜24 階差分の計算結果を示すとともに，各階数で高階差分値が 0 となる最高段数において，最も多くのブロックの高階差分値が 0 となる入力差分の取り方のみ記述する．4 階差分では，9 段目まで，出力が 3 次以下，8 階差分では 10 段目の出力が 7 次以下であることが分かった．11 段目については，12 階差分で高階差分値が 0 となり，出力が 11 次以下であることが分かった．しかし，12 段以降については，16 階及び 20 階に階数を増加させても高階差分値が 0 とならなかった．更に階数を増加させ，24 階差分を行うと 12 段目の高階差分値が 0 となり，出力が 23 次であることが分かった．しかし，36 階まで階数を増加させても，13 段目は代数次数が飽和することが分かった．

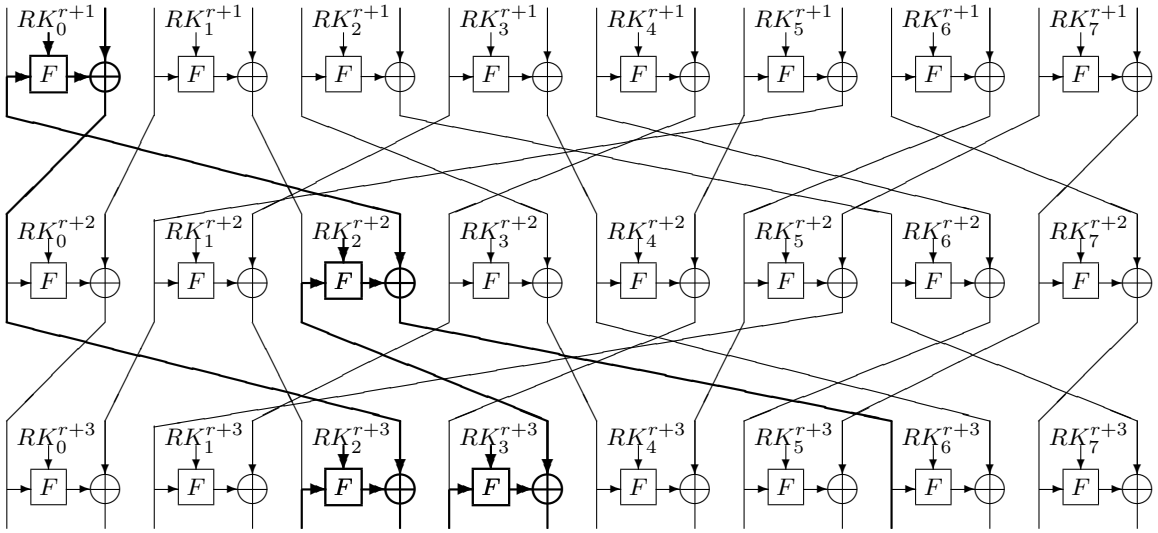

図 3: 3 段消去攻撃の例

4.3 実験結果の解析

TWINE の構造から，もし r 段出力の奇数ブロックの高階差分値が 0 であった場合は $r+1$ 段の，偶数ブロックの高階差分値が 0 であった場合は $r+2$ 段の F 関数について，式 (5) の攻撃方程式が適用できる．

攻撃に必要な計算量については，攻撃方程式を導出するためには 2^N 個の暗号文を用いるため，鍵の候補 1 つあたりの攻撃方程式の計算量は 2^N となる．なお，ここでの計算量の単位は F 関数の計算とする．攻撃方程式の解法に全数探索を適用すると，$2^4 \times 2^N$ 回の計算量となる．また，この計算で候補を推定できる最終段の副鍵は 1 個であるため，全ての副鍵を特定するためには異なるブロックの組み合わせを用いて 8 回同様の計算を行う必要がある．さらに，複数存在する副鍵の候補から正しい副鍵を特定するには，これまでの計算を異なる 2 つの平文固定値に対して行う必要がある．したがって最終段のすべての副鍵を特定する場合，合計で $2^N \times 2^4 \times 8 \times 2 = 2^{N+8}$ 回の計算量となる．選択平文数については，2 つの異なる平文固定値に対し N 階差分を行うため，$2^N \times 2 = 2^{N+1}$ 個の選択平文が必要になる．

5 考察

5.1 多段消去型攻撃の適用例

第 3.2 節で示した攻撃方程式は，全数探索を行う副鍵の数を増やすことで多段消去型攻撃に拡張できる．例として 1 番目のブロックの入力値の高階差分値が 0 であることを利用して 3 段消去攻撃を行う際，全数探索を行う副鍵を図 3 に示す．以後，多段消去型攻撃を用いて最終段の全ての副鍵を特定するための計算量及び選択平文数を考察する．1 つの副鍵について，2^4 の全数探索を行うため，副鍵数が R とすると，1 回の攻撃方程式の計算量は $2^{4\times R}$ となる．また，1 回の攻撃方程式の計算で候補を推定できる最終段の副鍵数が 8 個以下である場合，複数の攻撃方程式を計算する必要がある．さらに，偶然一致する偽鍵を排除するためには，さらに別の攻撃方程式を用意する必要がある．1 つの攻撃方程式から 4 つのブール代数次数が導出できるので，$\lceil (4 \times R + 1)/4 \rceil$ 組以上の高階差分を計算するための平文数が必要になる．これは平文の固定値を変えることで得られる．図 3 の例については，5 組の高階差分を計算するための平文固定値が必要になる．

第 2.1 節で示したように，鍵スケジュール部には，他の副鍵との関係において一意的に値が決定する副鍵が複数存在する．また，既に特定した副鍵は全数探索を行う必要がない．これらの性質を利用すると，多段消去型攻撃の計算量を削減することができる．

以上を考慮して，多段消去型攻撃を行い，最終段のすべての副鍵を特定するために消去する段数，平文固定値数及び計算量の関係を鍵長ごとに表 5 及び表 6 に示す．

各鍵長における攻撃可能段数，階数，消去段数，選択平文数及び計算量を表 7 及び表 8 に示

す．24 階差分で 12 段目出力の一部のブロックの高階差分値が 0 となることを利用し，鍵長が 80 ビットの場合は 6 段消去を行い 18 段，128 ビットの場合は 7 段消去を行い 19 段まで攻撃可能であることが分かった．

表 5: 鍵長 80 ビットにおける多段消去型攻撃に必要な計算量

段数	副鍵数	平文固定値数	計算量
1	1	2	$(2^4)^1 * 8 * 2 = 2^8$
2	2	3	$(2^4)^2 * 8 * 3 = 2^{12.58}$
3	4	5	$(2^4)^4 * 8 * 5 = 2^{20.32}$
4	7	8	$(2^4)^7 * 3 * 8 + (2^4)^4 * 5 = 2^{32.59}$
5	10	11	$(2^4)^{10} * 11 + (2^4)^8 * 9 = 2^{43.46}$
6	13	14	$(2^4)^{13} * 14 + (2^4)^5 * 6 = 2^{55.81}$
7	17	18	$(2^4)^{17} * 18 = 2^{72.17}$

表 6: 鍵長 128 ビットにおける多段消去型攻撃に必要な計算量

段数	副鍵数	平文固定値数	計算量
1	1	2	$(2^4)^1 * 8 * 2 = 2^8$
2	2	3	$(2^4)^2 * 8 * 3 = 2^{12.58}$
3	4	5	$(2^4)^4 * 8 * 5 = 2^{20.32}$
4	7	8	$(2^4)^7 * 3 * 8 + (2^4)^4 * 5 = 2^{32.59}$
5	12	13	$(2^4)^{12} * 13 + (2^4)^8 * 9 = 2^{51.7}$
6	18	19	$(2^4)^{18} * 19 + (2^4)^5 * 6 = 2^{76.24}$
7	23	24	$(2^4)^{23} * 24 = 2^{96.58}$
8	28	29	$(2^4)^{28} * 29 = 2^{116.85}$

表 7: 鍵長 80 ビットにおける最大攻撃可能段数

攻撃可能段数	階数	消去段数	選択平文数	計算量
16	4	7	$2^4 * 18 = 2^{8.17}$	$2^{76.17}$
16	8	6	$2^8 * 14 = 2^{11.81}$	$2^{63.81}$
17	12	6	$2^{12} * 14 = 2^{15.81}$	$2^{67.81}$
18	24	6	$2^{24} * 14 = 2^{27.81}$	$2^{79.81}$

表 8: 鍵長 128 ビットにおける最大攻撃可能段数

攻撃可能段数	階数	消去段数	選択平文数	計算量
17	4	8	$2^4 * 29 = 2^{8.86}$	$2^{120.85}$
18	8	8	$2^8 * 29 = 2^{12.86}$	$2^{124.85}$
18	12	7	$2^{12} * 24 = 2^{16.58}$	$2^{108.58}$
19	24	7	$2^{24} * 24 = 2^{28.58}$	$2^{120.58}$

表 9: 他の研究との比較

	攻撃可能段数	計算量	選択平文数
開発者（不能差分攻撃）	24	$2^{115.10}$	$2^{52.21}$
開発者（飽和攻撃）	23	$2^{106.14}$	$2^{62.81}$
文献 [8]	36	$2^{126.82}$	2^{60}
文献 [9]	25	2^{122}	2^{48}
文献 [9]	27	$2^{123.5}$	$2^{62.29}$
本研究	19	$2^{120.58}$	$2^{28.58}$

5.2 拡散層に関する考察

開発者が述べているように，TWINE は独自の拡散層を採用している [7]．Zheng らが示した Type-2 一般化フェイステル構造はブロックを循環置換させるものである [14]．そこで，TWINE の拡散層をこの循環置換とした場合に，高階差分攻撃に対する耐性にどのような影響があるのかを計算機実験で確かめた．その際，TWINE が 24 階差分を用いたときに最も脆弱であったという計算機実験結果から，循環置換とした TWINE に対しても 24 階差分に対する安全性を評価することで比較する．その結果，23 段目において $X_{\{0,1,2,3,4,5,6\}} \rightarrow Y_{0\{9,11\}}$ が成立し，出力次数が 23 次以下であることを確認できた．

次に，第 2.1 節で示しものと同様の鍵スケジュール部が循環置換とした TWINE でも採用されていると仮定し，多段消去型攻撃を行う際の計算量を算出した．その際，第 5.1 節と同様に，副鍵の中には他の副鍵との関係で一意的に値が決定されるものが存在することを利用した．その結果，循環置換とした場合は，副鍵間の従属関係が TWINE 仕様の置換の場合に比べて多く存在するため，より効果的な攻撃が実施可能になった．10 段消去攻撃の場合，TWINE 仕様の置換は全数探索する副鍵数が 33 個で，$2^{137.09}$ 回の計算量となるが，循環置換とした場合副鍵数が 24 個で $2^{100.64}$ 回の計算量で行えることが分かった．従って，鍵長 128 ビット長において，24 階差分で 23 段目の 9 番目のブロックの高階差分値が 0 となることを利用し，10 段消去攻撃を行うと $2^{124.64}$ という計算量で 33 段構成の攻撃が可能になることが分かった．前項で示したとおり，同様の条件で TWINE 仕様の置換の場合は攻撃可能段数が 19 段であり，その差は 14 段となる．従って，TWINE の採用した拡散層が高階差分攻撃に対する耐性を向上させていることが確認できた．

5.3 他の研究との比較

128 ビット鍵長を仮定した場合の開発者及び他の研究者が研究を行った攻撃法と本稿の攻撃法との比較を表 9 に示す．攻撃可能段数につい

ては，本稿が示した攻撃法が最小であるが，選択平文数については，最も少ないデータ量で攻撃が行えることが分かる．第 3.3 節で示したように，開発者らが示した飽和攻撃は，本稿とほぼ同様の攻撃手法であると言える．本稿が示した攻撃法では，選択平文数は約 3.2GB というデータ量になるが，開発者らが示した攻撃法では，その $2^{34.23}$ 倍の選択平文数が必要となる．

6 おわりに

本研究では，TWINE が 4 ビット S-box を採用していることに着目し，4〜36 階差分までの高階差分特性を 4 ビット処理単位で網羅的に解析した．その結果，24 階差分で 12 段目出力の一部のブロックの高階差分値が 0 となることを発見し，鍵長が 80 ビット及び 128 ビットの場合における攻撃可能な最大段数がそれぞれ 18 段及び 19 段であることが分かった．

軽量型ブロック暗号は，過度に安全性を重視する場合，軽量性及び高速性を損なう可能性がある．そのため，軽量型ブロック暗号の安全性評価は他の要件を踏まえて行う必要がある．本稿では，TWINE の高階差分特性のみを評価したが，今後は軽量型ブロック暗号が具備すべき要件を総合的かつ具体的に勘案し，安全性評価を行う予定である．

参考文献

[1] ISO/IEC 29192-2, “Information technology - Security techniques - Lightweight cryptography - Part 2: Block ciphers,” January 2012.

[2] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT: An Ultra-Lightweight Block Cipher”, CHES’07, LNCS 4727, 2007, pp450-466.

[3] Sony Corporation, “The 128-bit blockcipher CLEFIA: Algorithm specification. Revision 1.0,” http://www.sony.net/Products/clefia/technical/data/clefia-spec-1.0.pdf.

[4] G. Leander, C. Paar, A. Poschmann, and K. Schramm, “New Lightweight DES Variants,” Fast Software Encryption-FSE’07, LNCS 4593, pp196-210.

[5] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K.Jeong, H. Kim, J. Kim, and S. Chee, “HIGHT: A New Block Cipher Suitable for Low-Resource Device,” CHES’06, LNCS 4249, pp46-59.

[6] L. R. Knudsen, G. Leander, A. Poschmann, and M. J. B. Robshaw, “PRINTcipher: A Block Cipher for IC-Printing,” CHES’10, LNCS 6225, 2010 pp16-32.

[7] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, “TWINE: Lightweight Block Cipher for Multiple Platforms,”Proc. of Selected Areas in Cryptography, LNCS 7707, 2013, pp339-354.

[8] T. Iwata, K. Minematsu, J. Guo, and S. Morioka, “CLOC: Compact Low-Overhead CFB,”http://competitions.cr.yp.to/round1/clocv1.pdf.

[9] Mustafa Çoban, Ferhat Karakoç, and Özkan Boztaş, “Biclique Cryptanalysis of TWINE,” Cryptology and Network Security, LNCS 7712, 2012, pp43-55.

[10] Özkan Boztaş, Ferhat Karakoç, and Mustafa Çoban, “Multidimensional Meet-in-the-Middle Attacks on Reduced-Round TWINE-128,” Lightweight Cryptography for Security and Privacy, LNCS 8162, 2013, pp55-67.

[11] Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen, and Jingyuan Zhao, “Key Difference Invariant Bias in Block Ciphers,”Advances in Cryptology - ASIACRYPT 2013, LNCS 8269, 2013, pp357-376.

[12] T. Suzaki, and K. Minematsu. “Improving the Generalized Feistel,” Fast Software Encryption-FSE’10, LNCS 6147, 2010, pp.19-39.

[13] J. Daemen, L. R. Knudsen, and V. Rijmen, “The Block Cipher Square,” 4th Fast Software Encryption Workshop, LNCS 1267, 1997, pp.149-165.

[14] Y. Zheng, T. Matsumoto, and H. Imai, “On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses,”Advances in Cryptology - CRYPTO’89, LNCS 435, 1989, pp.461-480.