KCipher-2の安全性に関する評価

白石善明
名古屋工業大学

2011年1月31日

目 次

1　はじめに

KCipher-2 は，SASC2007 において K2 という名前（商標上の理由で KCipher-2 に変更）で最初のバージョン [1] が示され，後に，SECRYPT2007 において K2 v2.0 という名前で初期化処理等を修正したバージョン [2] が示されている．32 ビットワードの FSR（feedback shift register）のフィードバック関数に対して DFC（dynamic feedback control）を行うというソフトウェア実装を想定した同期式ストリーム暗号である．IV サイズは 128 ビット，鍵サイズは 128 ビット，192 ビット，256 ビットの中から選択することができ，キーストリームとして 1 サイクルあたり 32 ビットワードを 2 つ出力する．

KCipher-2 は，32 ビットワード単位の演算と FSR に対するフィードバック関数の DFC 機構により，キーストリーム出力を効率化するとともに，キーストリーム出力の線形関係を特定困難にしている．さらに，ストリーム暗号 SNOW2.0[23] の FSM（finite state machine）を 2 つ結合した構造の非線形関数を有しており，キーストリーム出力の代数次数を増加させている．

KCipher-2 の安全性の議論は，KCipher-2 の提案論文 [1, 2] および第三者評価書 [3, 4] によるものが主であるが，現在，脆弱性は発見されていない．本報告では，提案論文および第三者評価書の内容をもとに，DFC 機構により制御される FSR の出力の周期と線形複雑度，KCipher-2 の識別攻撃，相関攻撃，代数攻撃，タイムメモリトレードオフ（TMTO）攻撃，推測決定（GD）攻撃，関連鍵/選択 IV 攻撃，KCipher-2 の出力の統計的性質について確認したことをまとめる．

なお，ストリーム暗号では攻撃者が意図したキーストリームを選択的に観測することが一般に困難であり，ブロック暗号の選択平文攻撃の一つである差分攻撃をそのまま適用できないため，これに類似する攻撃として関連鍵/選択 IV 攻撃を位置づけている．

以降，KCipher-2 に関する議論は，K2 v2.0[2] を想定する．

2　KCipher-2

2.1　構造

KCipher-2 は，図 1 のように，2 個の FSR（FSR-A ,FSR-B）と，4 個の内部レジスタ（$R1, R2, L1, L2$）を有する非線形関数と，DFC から構成

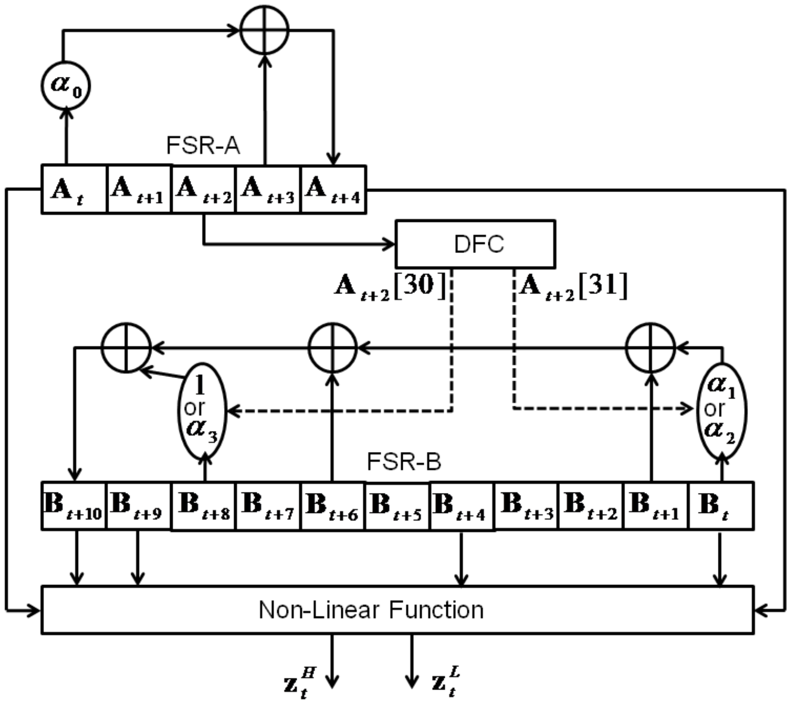

図 1: KCipher-2 の構造

されている．FSR-A は 5 段のレジスタを，FSR-B は 11 段のレジスタを持ち，各レジスタのサイズは 32 ビットである．$R1, R2, L1, L2$ のレジスタのサイズはそれぞれ 32 ビットである．FSR-A は固定のフィードバック関数により状態遷移し，FSR-B は FSR-A の出力を受ける DFC によりフィードバック関数が制御される．非線形関数は，FSR-A の A_t, A_{t+4} と，FSR-B の $B_t, B_{t+4}, B_{t+9}, B_{t+10}$ を入力し，キーストリームとして z_t^L, z_t^H を出力する．

FSR-A のフィードバック関数 $f_A(x)$ と FSR-B のフィードバック関数 $f_B(x)$ は，それぞれ次のように書かれる．

$$
\begin{aligned}
f_A(x) &= \alpha_0 x^5 + x^2 + 1, \\
f_B(x) &= (\alpha_1^{cl1_t} + \alpha_2^{1-cl1_t} - 1)x^{11} + x^{10} + x^5 + \alpha_3^{cl2_t} x^3 + 1.
\end{aligned}
$$

$cl1_t, cl2_t$ は，DFC の出力であり，次のような値である．ただし，$A_x[y]$ は，

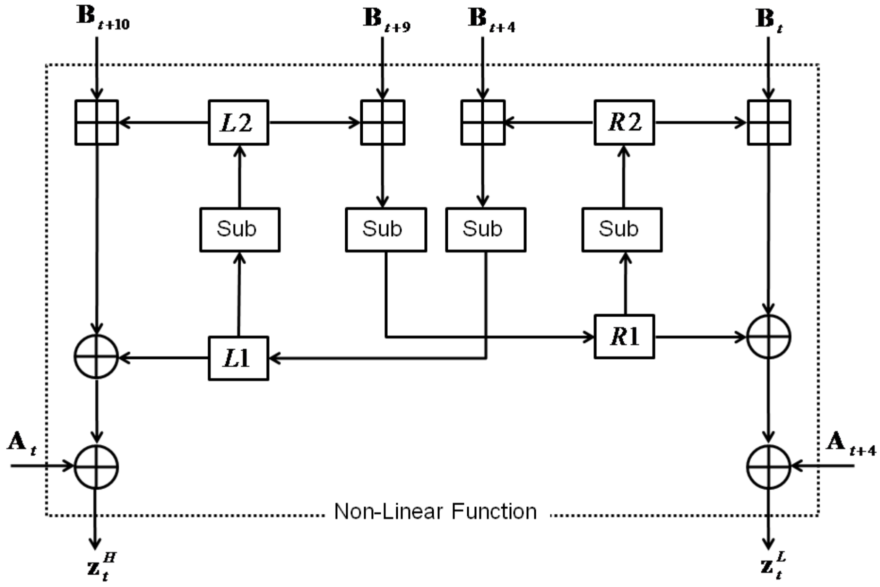

図 2: 非線形関数

FSR-A の x 番目のレジスタの y 番目のビット値とする．

$$cl1_t = A_{t+2}[30],\ cl2_t = A_{t+2}[31].$$

$\alpha_0, \alpha_1, \alpha_2, \alpha_3$ は，それぞれ次の多項式の根とする．

$$\begin{aligned}
&x^4 + \beta^{24}x^3 + \beta^3x^2 + \beta^{12}x + \beta^{71} \in GF(2^8)[x],\\
&x^4 + \gamma^{230}x^3 + \gamma^{156}x^2 + \gamma^{93}x + \gamma^{29} \in GF(2^8)[x],\\
&x^4 + \delta^{34}x^3 + \delta^{16}x^2 + \delta^{199}x + \delta^{248} \in GF(2^8)[x],\\
&x^4 + \zeta^{157}x^3 + \zeta^{253}x^2 + \zeta^{56}x + \zeta^{16} \in GF(2^8)[x].
\end{aligned}$$

$\beta, \gamma, \delta, \zeta$ は，それぞれ次の多項式の根とする．

$$\begin{aligned}
&x^8 + x^7 + x^6 + x + 1 \in GF(2)[x],\\
&x^8 + x^5 + x^3 + x^2 + 1 \in GF(2)[x],\\
&x^8 + x^6 + x^3 + x^2 + 1 \in GF(2)[x],\\
&x^8 + x^6 + x^5 + x^2 + 1 \in GF(2)[x].
\end{aligned}$$

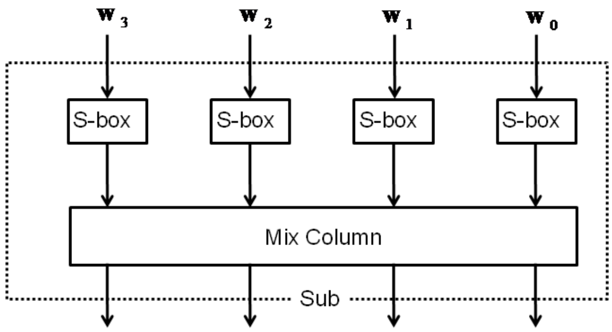

図 3: Sub() の構造

ここで，演算単位の 4 バイト（32 ビット）の値は，$Y_i \in GF(2^8), i = 0,1,2,3$ により，次のように表す．ただし，Y_3 を上位バイトとする．

$$Y = Y_3\alpha_0^3 + Y_2\alpha_0^2 + Y_1\alpha_0 + Y_0.$$

1 バイトの値は，$y_i \in GF(2), i = 0,1,\cdots,7$ により，次のように表す．ただし，y_7 を上位ビットとする．

$$y = y_7\beta^7 + y_6\beta^6 + \cdots + y_1\beta + y_0.$$

非線形関数は，図 2 のように，キーストリームを計算する部分と，内部レジスタを更新する部分から成る．キーストリーム $Z_t = (z_t^H, z_t^L)$ は，次のように計算される．

$$\begin{aligned} z_t^L &= B_t + R2_t + R1_t + A_{t+4}, \\ z_t^H &= B_{t+10} + L2_t + L1_t + A_t. \end{aligned}$$

内部レジスタ $R1, R2, L1, L2$ は，次のように更新される．

$$\begin{aligned} R1_{t+1} &= Sub(L2_t + B_{t+9}), \\ R2_{t+1} &= Sub(R1_t), \\ L1_{t+1} &= Sub(R2_t + B_{t+4}), \\ L2_{t+1} &= Sub(L1_t). \end{aligned}$$

$Sub()$ は，図 3 のように，8 ビット入力-8 ビット出力の S-box と，32 ビット入力-32 ビット出力の Mix Column から成る．これらは，ブロック暗号 AES[24] の S-box と Mix Column と同じものである．

2.2 初期化処理

KCipher-2 の初期化処理は，key loading step と internal state initialization step の 2 つに分けられる．

key loading step では，key scheduling algorithm により，128/ 192/256 ビット初期化鍵と，128 ビットの初期化ベクトル (IV) から，内部初期状態を生成する．key scheduling algorithm は，ブロック暗号 AES[24] の round key generation function と同じであり，128/192/256 ビットの初期化鍵を 384 ビットに拡張する．例えば，128 ビットの初期化鍵 $IK = (IK_0, IK_1, IK_2, IK_3)$ に対しては，次のようになる．ただし，$0 \leq i \leq 11$ とする．

$$\begin{aligned}
K_i &= IK_i, 0 \leq i \leq 3, \\
K_i &= K_{i-4} \oplus K_{i-1},\ i \neq 4n, \\
K_i &= K_{i-4} \oplus Sub((K_{i-1} << 8) \oplus (K_{i-1} >> 24)) \\
&\quad \oplus Rcon(i/4-1),\ i = 4n, \\
Rcon[i] &= (x^i \mod x^8 + x^4 + x^3 + x + 1, \\
&\quad 0x00, 0x00, 0x00),\ x = 0x02.
\end{aligned}$$

internal state initialization step では，$K_i, i = 0, 1, \cdots, 11$ と $IV = (IV_0, IV_1, IV_2, IV_3)$ により，次のように内部状態を初期化する．

$$\begin{aligned}
A_m &= K_{4-m}, m = 0, 1, \cdots, 4,\ B_0 = K_{10},\ B_1 = K_{11}, \\
B_2 &= IV_0,\ B_3 = IV_1,\ B_4 = K_8,\ B_5 = K_9,\ B_6 = IV_2, \\
B_7 &= IV_3,\ B_8 = K_7,\ B_9 = K_5,\ B_{10} = K_6.
\end{aligned}$$

そして $R1, R2, L1, L2$ を $0x00$ に設定して，24 サイクル動かして，内部状態を更新する．ただし，A_{j+4}, B_{j+10} は次のように更新する．

$$\begin{aligned}
A_{j+4} &= \alpha_0 A_{j-1} \oplus A_{j+2} \oplus z^L_{j-1}, \\
B_{j+10} &= (\alpha_1^{cl1_{j-1}} + \alpha_2^{1-cl1_{j-1}} - 1) B_{j-1} \oplus B_j \\
&\quad \oplus B_{j+5} \oplus \alpha_3^{cl2_{j-1}} B_{j+7} \oplus z^H_{j-1}.
\end{aligned}$$

キーストリームを 2^{64} ビット（2^{58} サイクル）出力すると，再初期化として，上記の処理が実行される．

3　KCipher-2の安全性

3.1　周期と線形複雑度

第三者評価書 [3] では，DFC によりフィードバック関数が制御される FSR（DFSR）の出力の周期と線形複雑度について，以下のように議論されている．

まず，DFSR の出力を定式化し（3.1.1 節），次に，interleaved sequence の定義とこれに関する補題を示し（3.1.2 節），そして，DFSR の出力を interleaved sequence と見なして周期と線形複雑度の上限を明らかにしている（3.1.3 節）．

3.1.1　DFSR の出力の表現

DFC によりフィードバック関数が制御される FSR-B の出力 $s_i(t)$ は，次のように書ける．ただし，時刻 t の FSR-B の出力を $b(t)$，FSR-A の周期を l_A，FSR-A の段数を n_A とおく．

$$\begin{aligned} s_i(t) &= b(l_A\ t+i), \\ l_A &= 2^{32n_A}-1,\ t\geq 0,\ i=0,1,\cdots,l_A-1. \end{aligned}$$

FSR-B の出力を $b(l_A\ t+i)$ から $b(l_A(t+1)+i)$ に変化させる遷移行列 M_i は，次のように書ける．ただし，FSR-B の状態を 1 つ進める遷移行列を $B_{00}, B_{01}, B_{10}, B_{11}$，FSR-B の段数を n_B，$GF(2^{32})$ 上の n_B 次正方行列の集合を $M_{n_B}(GF(2^{32}))$ とする．

$$\begin{aligned} M_i &= \prod_{k=1}^{l_A} B_k^{(i)} \in M_{n_B}(GF(2^{32})), \\ & \quad B_k^{(i)} \in \{B_{00}, B_{01}, B_{10}, B_{11}\}. \end{aligned}$$

ここで，$s_i(t)$ は，次のように書き直すことができる．ただし，FSR-B の出力を得るための線形関数を $\Pi()$，時刻 t の FSR-B の状態を $B(t)$ =

$\{b(t), b(t+1), ..., b(t+n_B-1)\}$ とする．

$$\begin{aligned} s_i(t) &= b(l_A\ t+i) = \Pi(B(l_A\ t+i)) \\ &= \Pi(M_i^t B(i)) = \Pi M_i^t B(i). \end{aligned}$$

FSR-A の出力により FSR-B のフィードバック関数が制御されることから，$s_i(t), i=0,1,\cdots,l_A-1$ における状態遷移と，$s_i(t+1), i=0,1,\cdots,l_A-1$ における状態遷移は同じであり，次の関係が成り立つ．

$$M_{i+1} = (B_1^{(i)})^{-1} M_i B_1^{(i)}.$$

M_i の特性多項式を $c_i(x) = \sum_{k=0}^{n} c_k x^k$ とおくと，次のような関係が成り立つ．すなわち，M_i は全て同じ特性多項式である．

$$c_i(x) = c_j(x),\ 0 \le \{i,j\} \le l_A - 1.$$

また，$s_i(t), t=0,1,\cdots$ について，次のような式が成り立つ．$c(x)$ の次数が n であることから linear span は n となる．

$$\begin{aligned} \sum_{k=0}^{n} c_k s_i(t+k) &= \sum_{k=0}^{n} c_k \Pi(B((t+k)l_A+i)) \\ &= \sum_{k=0}^{n} c_k \Pi M_i^{t+k} B(i) \\ &= \Pi M_i^t \sum_{k=0}^{n} c_k M_i^k B(i) \\ &= \Pi M_i^t c(M_i) B(i) = 0. \end{aligned}$$

3.1.2 Interleaved sequence

l を正整数とし，次数 n の $GF(q)$ 上の多項式を $f(x), f(0) \neq 0$，$GF(q)$ 上の系列を $\mathbf{u} = \{u(t)\}, t = i\ l + j, i = 0,1,\cdots, j = 0,1,\cdots,l-1$ とする．$j = 0,1,\cdots,l-1$ について，$\mathbf{u}_j = \{u(i\ l+j)\}_{i>=0}$ が $f(x)$ で生成されるとき，$\mathbf{u}$ を $GF(q)$ 上の interleaved sequence，$\mathbf{u}_j$ を $\mathbf{u}$ の component sequence と呼び，次の補題が示されている．

補題 1[5]：$\mathbf{u}$ は $f(x)$ と l で作られる $GF(q)$ 上の interleaved sequence とし，$h(x)$ は $GF(q)$ 上の $\mathbf{u}$ の最小多項式とする．このとき，1)linear span

が高だか $n\ l$ であるためには，$h(x)$ は $f(x^l)$ を割り切る，2)$\mathbf{u}$ の系列の周期は $ord(f)l$ を割り切る．

この補題は，interleaved sequence すなわち $s_i(t)$ の周期と線形複雑度の上限（周期の上限は $ord(f)\ l$，線形複雑度の上限は $n\ l$）を示している．

3.1.3 DFSR の出力の周期と線形複雑度の上限

$s_i(t)$ は，$GF(2^{32})$ 上の interleaved sequence であり，$l = l_A, f(x) = c(x)$（$f(x)$ の次数は最大352）であることから，周期の上限は $(2^{352}-1)(2^{160}-1)$ また線形複雑度の上限は $352(2^{160}-1)$ となる．

以上のように，第三者評価書 [3] において，DFC によりフィードバック関数が制御される FSR-B の出力の周期と線形複雑度の上限が示されている．下限については現在明らかにされておらず，第三者評価書 [4] において，実験により短い周期，小さな線形複雑度が存在しないことが述べられている．また，FSR-A の出力の周期が $2^{160}-1$ であることから，KCipher-2 のキーストリームは，再初期化を実施する 2^{64} ビットよりも十分長い周期が得られることが期待できると述べられている．

3.2 識別攻撃

KCipher-2 の提案論文 [2] では，キーストリームを真の乱数と区別する識別攻撃について，以下のように議論されている．

識別攻撃の基本的なアイデアは Coppersmith らの攻撃 [6]（3.2.1 節）に基づいている．SNOW 2.0 に対する識別攻撃として渡辺らの攻撃 [7, 8]（3.2.2 節）が挙げられており，これを KCipher-2 に適用した結果（3.2.3 節）が述べられている．

3.2.1 Coppersmith らの識別攻撃

提案論文 [2] で記されている識別攻撃の基本的なアイデアは，ブロック暗号に対する線形解読法における distinguisher の構成法を利用した Coppersmith らの識別攻撃 [6] に基づいている．

アルゴリズムを線形パートと非線形パートに分割して，線形パートの関係式を次のように書く．

$$\phi(s,t)=\bigoplus_j c_j s_{t+j}=0. \tag{1}$$

非線形パートについては，最良近似を探索して，そこで得られた近似式を次のように書く．ただし，左辺を $f(s,t)$，右辺を $g(z,t)$ とおく．

$$\bigoplus_i \Gamma_i s_{t+i}=\bigoplus_t \Gamma'_t z_t. \tag{2}$$

ここで，式 (1),(2) から，次のようなキーストリームからなる線形近似を構成できる．

$$\begin{aligned} 0 &= \bigoplus_i \Gamma_i \phi(s,t+i) \\ &= \bigoplus_j c_j f(s,t+j) \\ &= \bigoplus_j c_j g(z,t+j). \end{aligned} \tag{3}$$

式 (3) を distinguisher として用いると，識別攻撃を実施できるというものである．

3.2.2 渡辺らの識別攻撃

前節の改良版として，SNOW 2.0 を想定した渡辺らの識別攻撃 [7, 8] が示されており，提案論文 [2] では，KCipher-2 に対してこれが適用されている．

FSM の線形近似が次のような式で与えられていると仮定する．

$$\bigoplus_i \Gamma_i s_{t+i}=\bigoplus_j \Gamma'_j z_{t+j}. \tag{4}$$

一方，LFSR の内部状態 s は，任意のマスク値 Γ において，次のような関係式を満たす．

$$\Gamma(\bigoplus_j c_j s_{t+j})=0.$$

これは，次のような式で書き直すことができる．

$$\bigoplus_j (\Gamma c_j) s_{t+j} = 0. \tag{5}$$

ここで，マスク値 Γc_i について，式 (4) が大きな偏差で成り立つならば，式 (5) も大きな偏差を持つ．このとき，式 (4),(5) から，distinguisher として，次のような式が得られる．ただし，式 (4) の右辺を $g'(z,t)$ とおく．

$$\bigoplus_j c_j g'(z, t+j) = 0. \tag{6}$$

式 (4) の偏差は piling up lemma より次のような式で書くことができる．ただし，ϵ_i は FSM 内の線形近似の偏差とする．

$$\epsilon_{\mathrm{FSM}}(\Gamma) = 2^{n-1} \prod_{i=1}^{n} \epsilon_i.$$

式 (6) は式 (4) から成り，その偏差は次のような式で書くことができる．

$$\epsilon = 2^{m-1} \prod_{i=1}^{m} \epsilon_{\mathrm{FSM}}(\Gamma''_i).$$

3.2.3 渡辺らの識別攻撃の KCipher-2 への適用

前節で示した渡辺らの識別攻撃 [7, 8] を KCipher-2 に適用した結果が提案論文 [2] では示されている．

ここでは，議論を簡単にするために，DFC の出力ビットを $cl1_t = cl2_t = 0$ と固定した場合を想定している．

FSR-A のフィードバック関数と FSR-B のフィードバック関数から，内部状態に関する関係式は，次のように書ける．

$$\begin{aligned}
\phi_1(A,t) &= \bigoplus_i c1_j A_{t+i} \\
&= \alpha_0 A_t \oplus A_{t+3} \oplus A_{t+5} = 0, \\
\phi_2(B,t) &= \bigoplus_j c2_j B_{t+j} \\
&= \alpha_2 B_t \oplus B_{t+1} \oplus B_{t+6} \oplus B_{t+8} \oplus B_{t+11} = 0.
\end{aligned}$$

この2つの式を組み合わせると，任意のマスク値 Γ において，次のような関係式が得られる．

$$
\begin{aligned}
\Gamma\phi(s,t) &= \bigoplus_j \Gamma c1_j \phi_2(s, t+j) \\
&= \bigoplus_j \Gamma c_j s_{t+j} \\
&= \Gamma\alpha_0\alpha_2 s_t \oplus \Gamma\alpha_2(s_{t+3} \oplus s_{t+5}) \\
&\quad \oplus\Gamma\alpha_0(s_{t+1} \oplus s_{t+6} \oplus s_{t+8} \oplus s_{t+11}) \\
&\quad \oplus s_{t+4} \oplus s_{t+6} \oplus s_{t+9} \oplus s_{t+13} \oplus s_{t+14} \oplus s_{t+16} = 0. \qquad (7)
\end{aligned}
$$

KCipher-2 の FSM の2ラウンド出力の線形マスクは図4のようになり，線形近似式は次のように書ける．

$$
\begin{aligned}
\bigoplus_i \Gamma_i s_{t+i} &= \bigoplus_j \Gamma'_j z_{t+j} \\
&= \Gamma z_t^H \oplus \Lambda z_t^L \oplus \Phi z_{t+1}^H \oplus \Psi z_{t+1}^L. \qquad (8)
\end{aligned}
$$

ここで，マスク値 Γc_j について，式 (8) が大きな偏差で成り立つならば，式 (7) も大きな偏差を持つ．よって，式 (7),(8) を組み合わせると，distinguisher として，次のような式が得られる．ただし，式 (8) の右辺を $g'(z,t)$ とおく．

$$
\begin{aligned}
\bigoplus_j c_j g'(z,\ t+j) = \Gamma\{&\alpha_0\alpha_2 z_t^H \oplus \alpha_2(z_{t+3}^H \oplus z_{t+5}^H) \oplus \alpha_0(z_{t+1}^H \oplus z_{t+6}^H \oplus z_{t+8}^H \\
&\oplus z_{t+11}^H) \oplus z_{t+4}^H \oplus z_{t+6}^H \oplus z_{t+9}^H \oplus z_{t+13}^H \oplus z_{t+14}^H \oplus z_{t+16}^H\} \\
&\oplus\Lambda\{\alpha_0\alpha_2 z_t^L \oplus \alpha_2(z_{t+3}^L \oplus z_{t+5}^L) \oplus \alpha_0(z_{t+1}^L \oplus z_{t+6}^L \oplus z_{t+8}^L \\
&\oplus z_{t+11}^L) \oplus z_{t+4}^L \oplus z_{t+6}^L \oplus z_{t+9}^L \oplus z_{t+13}^L \oplus z_{t+14}^L \oplus z_{t+16}^L\} \\
&\oplus\Phi\{\alpha_0\alpha_2 z_{t+1}^H \oplus \alpha_2(z_{t+4}^H \oplus z_{t+6}^H) \oplus \alpha_0(z_{t+2}^H \oplus z_{t+7}^H \oplus z_{t+9}^H \\
&\oplus z_{t+12}^H) \oplus z_{t+5}^H \oplus z_{t+7}^H \oplus z_{t+10}^H \oplus z_{t+14}^H \oplus z_{t+15}^H \oplus z_{t+17}^H\} \\
&\oplus\Psi\{\alpha_0\alpha_2 z_{t+1}^L \oplus \alpha_2(z_{t+4}^L \oplus z_{t+6}^L) \oplus \alpha_0(z_{t+2}^L \oplus z_{t+7}^L \oplus z_{t+9}^L \\
&\oplus z_{t+12}^L) \oplus z_{t+5}^L \oplus z_{t+7}^L \oplus z_{t+10}^L \oplus z_{t+14}^L \oplus z_{t+15}^L \oplus z_{t+17}^L\} = 0.
\end{aligned}
$$

式 (8) がマスク値 Γc_j について大きな偏差で成り立つものが発見された場合，上記の distinguisher により識別攻撃が実施できるが，現在，そのようなマスク値は見つかっていない．

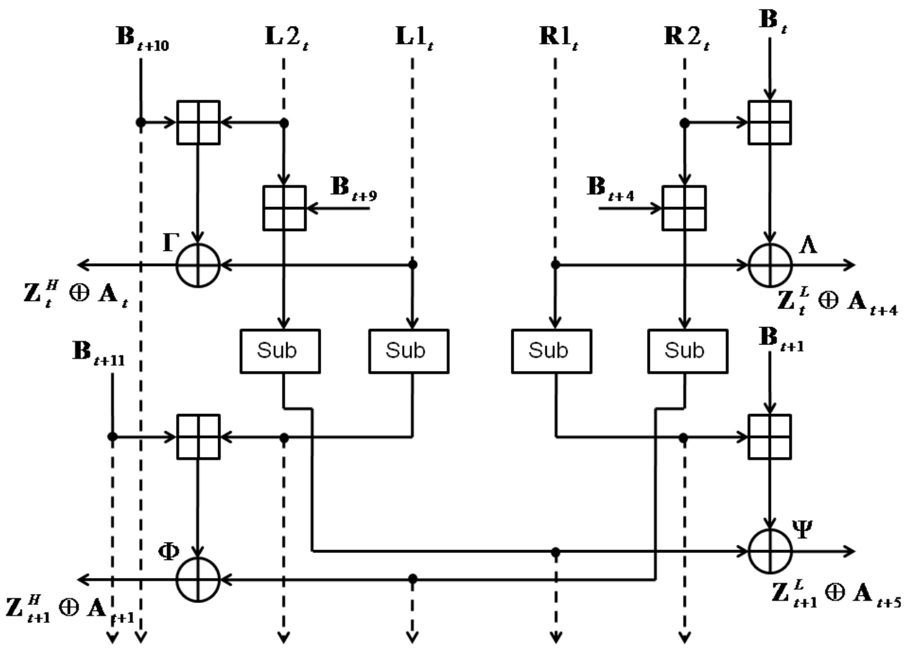

図 4: KCipher-2 の 2 ラウンド出力の線形マスク

以上のように，DFC の出力ビットを $cl1_t = cl2_t = 0$ のように固定した場合において，渡辺らの攻撃 [7, 8] の成立は困難であると提案論文 [2] では述べられている．また，$cl1_t = cl2_t = 0$ のような条件は，確率 2^{-30} で成立するため，このことから，DFC の効果として，識別攻撃の計算量を 2^{60} 倍にすることが述べられている．

3.3 相関攻撃

KCipher-2 の提案論文 [2] では，キーストリームを観測して，遷移する内部状態とキーストリームの相関関係から，内部状態を推定する相関攻撃について，以下のように議論されている．

相関攻撃の基本的なアイデアは Siegenthaler の攻撃 [9]（3.3.1 節）に基づいている．代表的な相関攻撃の一つである Chepyzhov らの攻撃 [11]（3.3.2 節）が挙げられており，これを KCipher-2 に適用した結果（3.3.3 節）が示されている．

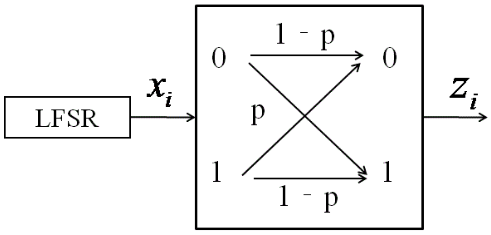

図 5: 相関攻撃のモデル

3.3.1 Siegenthaler の相関攻撃

提案論文 [2] で記されている相関攻撃の基本的なアイデアは，Siegenthaler の相関攻撃 [9] に基づいている．

図 5 のように，LFSR の出力列は 2 元線形 $[N, L]$ 符号の符号語，非線形関数の出力列は誤り確率 $p = 1/2 + \epsilon$ の BSC に符号語を入力したときの出力と見なす．ただし，N を LFSR の出力列の長さ，L を LFSR のフィードバック関数の次数とする．

非線形関数の出力列から LFSR の出力列（LFSR の内部状態）を求める問題は，線形符号の復号問題として扱うことができる．ここでは，復号アルゴリズムの中で復号誤り確率が良好とされる ML(max-imum likelihood)-decoding を適用する．

2 元線形 $[N, L]$ 符号の符号語の集合を $\mathbf{C}$，BSC に入力される送信語を $\bar{\mathbf{x}}$，送信語に対する受信語を $\mathbf{z}$ とおくと，ML-decoding は次のように定義される．

$$\bar{\mathbf{x_0}} \overset{def}{=} min_{\mathbf{x} \in \mathbf{C}} dist(\mathbf{x}, \mathbf{z}),\ dist(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{N} x_i \oplus z_i.$$

また，この復号誤り確率は，次のように定義される．

$$P_e(p) \overset{def}{=} Pr(\bar{\mathbf{x_0}} \neq \bar{\mathbf{x}}).$$

BSC の容量を $C = 1 - H(p)$ とすると，符号化レート $R = L/N$ が $R < C$ を満たすとき，次の式が成立する [9, 10]．ただし，$P_e(p)$ の期待値を $E[P_e(p)]$，ランダムコーディング指数を $\tau(R) > 0$ とする．

$$E[P_e(p)] < 2^{-\tau(R)N}, \tau(R) = \tau(R, p).$$

ML-decoding の復号誤り確率が 0 に近づく，すなわち攻撃が成立するための条件は，次のように書ける．

$$N > n_0, n_0 = \lceil L/C(p) \rceil, C(p) \approx \epsilon^2 2/(ln2).$$

ML-decoding を利用した相関攻撃の計算量は，LFSR の初期値を全数探索することから，$O(2^L L/C(p))$ に従う．

3.3.2 Chepyzhov らの相関攻撃

前節の改良版として，計算量のオーダを $O(2^L L/C(p))$ から $O(2^{\alpha L} L/C(p)), \alpha < 1$ へ減少させる Chepyzhov らの相関攻撃 [11] が示されており，提案論文 [2] ではこれを KCipher-2 に適用している．

LFSR の出力列 $\mathbf{x} = \{x_i\}_{i=1}^N$ とおくと，次のような式が書ける．ただし，LFSR のフィードバック関数を $g(D) = c_0 + c_1 D + \cdots c_L D^L (c_0 = c_L = 1)$ とする．

$$\begin{aligned}
&c_L x_1 + c_{L-1} x_2 + \cdots + c_1 x_L + c_0 x_{L+1} = 0, \\
&c_L x_2 + c_{L-1} x_3 + \cdots + c_1 x_{L+1} + c_0 x_{L+2} = 0, \\
&\qquad \vdots \\
&c_L x_{N-L} + c_{L-1} x_{N-L+1} + \cdots + c_1 x_{N-1} + c_0 x_N = 0.
\end{aligned}$$

次のような $(N-L) \times N$ のパリティ検査行列を定義すると，全ての符号語 $\mathbf{x}$ に関して，$H(x_1, x_2, \cdots, x_N)^T = 0$ が明らかに成り立つ．

$$H = \begin{pmatrix}
c_L & c_{L-1} & c_{L-2} & \cdots & c_0 & 0 & \cdots & 0 \\
0 & c_L & c_{L-1} & \cdots & c_1 & c_0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & 0 & \ldots & c_L & c_{L-1} & \cdots & c_1 & c_0
\end{pmatrix}.$$

さらに，次のような式が書ける．

$$\begin{aligned}
x_{L+1} &= h_{L+1}^1 x_1 + h_{L+1}^2 x_2 + \cdots + h_{L+1}^L x_L, \\
x_{L+2} &= h_{L+2}^1 x_1 + h_{L+2}^2 x_2 + \cdots + h_{L+2}^L x_L, \\
&\vdots \\
x_i &= h_i^1 x_1 + h_i^2 x_2 + \cdots + h_i^L x_L, \\
&\vdots \\
x_N &= h_N^1 x_1 + h_N^2 x_2 + \cdots + h_N^L x_L.
\end{aligned}$$

$h_i(D) = h_i^1 + h_i^2 D + \cdots h_i^L D^{L-1}$ とおくと，次のような式が明らかに成り立つ．

$$h_i(D) = D^{i-1} \mod g(D), i = 1, 2, \cdots N.$$

符号 $\mathbf{C}$ の $L \times N$ 生成行列は次のように書くことができ，左側から $(x_1, x_2, \cdots, x_L)$ を掛けることで，符号 $\mathbf{C}$ の符号語 $\mathbf{x}$ が得られる．

$$G = \begin{pmatrix} h_1^1 & h_2^1 & \cdots & h_N^1 \\ h_1^2 & h_2^2 & \cdots & h_N^2 \\ \vdots & & \cdots & \\ h_1^L & h_2^L & \ldots & h_N^L \end{pmatrix}.$$

ここで，次のような関係を満たす i, j を探索する．ただし $K < L$ とする．

$$h_i^{k+1} = h_j^{k+1}, h_i^{k+2} = h_j^{k+2}, \cdots, h_i^L = h_j^L, 1 \geq i \neq j \geq N.$$

探索した i, j から，次のような関係が得られる．

$$x_i + x_j = (h_i^1 + h_j^1)x_1 + (h_i^2 + h_j^2)x_2 + \cdots + (h_i^K + h_j^K).$$

ペア (i, j) の集合が得られたとき，次のような $[N_2, K]$ 符号 $\mathbf{C_2}$ の符号語 $\mathbf{x_2}$ を定義する．

$$(X_1, X_2, \cdots, X_{N_2}) = (x_{i_1} + x_{j_1}, x_{i_2} + x_{j_2}, \cdots, x_{i_{N_2}} + x_{j_{N_2}}).$$

符号 $\mathbf{C_2}$ の $K \times N_2$ 生成行列は，次のように書くことができ，左側から $(x_1, x_2, \cdots, x_K)$ を掛けることで，符号語 $\mathbf{x_2}$ が得られる．

$$G_2 = \begin{pmatrix} h_{i_1}^1 + h_{j_1}^1 & h_{i_2}^1 + h_{j_2}^1 & \cdots & h_{i_{N_2}}^1 + h_{j_{N_2}}^1 \\ h_{i_1}^2 + h_{j_1}^2 & h_{i_2}^2 + h_{j_2}^2 & \cdots & h_{i_{N_2}}^2 + h_{j_{N_2}}^2 \\ \vdots & & \cdots & \\ h_{i_1}^K + h_{j_1}^K & h_{i_2}^K + h_{j_2}^K & \cdots & h_{i_{N_2}}^K + h_{j_{N_2}}^K \end{pmatrix}.$$

また，符号語 $\mathbf{x_2}$ に対して，次のような受信語 $\mathbf{z_2}$ を定義する．

$$(Z_1, Z_2, \cdots, Z_{N_2}) = (z_{i_1} + z_{j_1}, z_{i_2} + z_{j_2}, \cdots, z_{i_{N_2}} + z_{j_{N_2}}).$$

符号 $\mathbf{C}$ の誤り系列を $(e_1, e_2, \cdots, e_N)$ とすると，符号 $\mathbf{C_2}$ の誤り系列は次のように書ける．

$$\begin{aligned} (E_1, E_2, \cdots, E_{N_2}) &= (e_{i_1} + e_{j_1}, e_{i_2} + e_{j_2}, \cdots, e_{i_{N_2}} + e_{j_{N_2}}), \\ Pr(E_m = 1) &= Pr(e_{i_m} + e_{j_m} = 1) = p_2 = 1/2 - 2\epsilon^2. \end{aligned}$$

このとき，2^K 通りの $(x_1, x_2, \cdots, x_K)$ のパターンに関して，符号語 $\mathbf{x_2}$ を構成して，受信語 $\mathbf{z_2}$ とのハミング距離を計算することで，$O(2^K K/C(p_2))$ で相関攻撃を適用できる．

LFSR の初期値のうち K ビットを推定して，残りの $L-K$ ビットは既存の他の復号アルゴリズムにより求める場合，攻撃が成立するための N_2 の条件は次の式で書ける．

$$N_2 > \lceil K/C(p_2) \rceil, C(p_2) \approx 4\epsilon^4 2/ln2.$$

この攻撃は $h_i^m = h_j^m, K \geq m \geq L, 1 \geq i \neq j \geq N$ を満たす 2 個のパリティ検査式を探索するものであるが，さらに H 個のパリティ検査式を探索するように一般化したものが示されている．このとき，パリティ検査式を探索する前処理の計算量は $N^{\lceil (H-1)/2 \rceil}$，パリティ検査式の個数は $N^H 2^{K-L}/H!$，主処理の計算量は $2^K \cdot N^H 2^{K-L}/H!$，攻撃が成立するための出力列の長さは $N \approx 1/4(2KH!ln2)^{1/H}\epsilon^{-2}2^{(L-K)/H}$ となる．

3.3.3 Chepyzhov らの相関攻撃の KCipher-2 への適用

前節で示した Chepyzhov らの相関攻撃 [11] を KCipher-2 に適用した結果が提案論文 [2] では示されている．

KCipher-2 は，FSR-A の出力により FSR-B のフィードバック関数が制御されていることから，FSR-B の出力に関するフィードバック関数を特定することが難しい．そこで，$cl1_t, cl2_t$ を一定の値で固定して，常時クロックを与える FSR-B の初期値を求める相関攻撃を想定し，また $GF(2^{32})$ 上の演算を $GF(2)$ 上の演算と見なして攻撃を適用する．$N = 2^{64}, H = 9, K = 26$ $, L = 11 \cdot 32 = 352$ とした場合，前処理の計算量は 2^{256}，パリティ検査式の個数は 2^{240}，主処理の計算量は 2^{266} である．常時クロックする FSR-B の出力列と非線形関数の出力列の相関確率は $p = 1/2 + 2^{-13}$ という値が得られており [2]，攻撃が成立するための出力列の長さは $N \approx 2^{62}$ である．

以上のように，FSR-B のフィードバック関数を固定して，$GF(2)$ 上の演算を仮定したとしても，Chepyzhov らの攻撃の計算量は鍵空間の全数探索よりも大きくなり，攻撃が困難であると提案論文 [2] では述べられている．また，FSR-B のフィードバック関数の DFC と $GF(2^{32})$ 上の演算を想定する場合は，相関値の計算量や復号アルゴリズムの計算量が増大し，相関攻撃に用いる線形式を特定できず，さらに攻撃は困難になることが述べられている．

3.4 TMTO 攻撃

KCipher-2 の提案論文 [2] では，キーストリームの部分系列と内部状態の対応表に基づき，キーストリームを観測して内部状態を推定するタイムメモリトレードオフ（TMTO）攻撃について，以下のように議論されている．

TMTO 攻撃の基本的なアイデアは Babbage，Golic の攻撃 [12, 13]（3.4.1 節）に基づいている．代表的な TMTO 攻撃の一つである Biryukov らの攻撃 [14, 15]（3.4.2 節）が挙げられており，これを KCipher-2 に適用した結果（3.4.3 節）が示されている．

3.4.1 Babbage, Golic の TMTO 攻撃

提案論文 [2] で記されている TMTO 攻撃の基本的なアイデアは，Babbage, Golic の攻撃 [12, 13] に基づいている．

内部状態の組み合わせを N，観測したキーストリーム列の個数を D，必要とするメモリの容量を M，前処理の時刻複雑度を P，主処理の時刻複雑度を T とする．

$\log N$ ビットの内部状態を $\log N$ ビットのキーストリームに置き換える関数 f を次のように定義する．

$$f : \log N \text{ bit state}- > \log N \text{ bit keystream},\ M = N/D.$$

ランダムに内部状態 s を選択して，内部状態とキーストリームのペア $(s, f(s))$ から成るテーブルを作る．テーブルは，ペア $(s, f(s))$ のうち $f(s)$ によりソートしておくものとする．

キーストリーム列を D 個だけ観測したとき，それぞれにおいて，ペア $(s, f(s))$ のうち $f(s)$ に一致するものを探す．このとき，バースデイパラドクスより，高い確率で該当のエントリーを見つけることができる．もしも一致するものが見つかった場合，対応する s が攻撃者が求める内部状態となる．

ここで，テーブルのエントリーの個数 M については $M = N/D$ という関係が得られており，テーブルを探索する回数は多くとも D 回であることから時刻複雑度 T については $T = D$ の関係が得られる．したがって，時刻とメモリのトレードオフの式は次のようになる．

$$TM = N,\ P = M,\ 1 \le T \le D.$$

3.4.2 Biryukov らの TMTO 攻撃

前節の改良版として，ブロック暗号に対する攻撃 [16] をストリーム暗号に適用した Biryukov らの攻撃 [14, 15] が示されており，提案論文 [2] ではこれを KCipher-2 に適用している．

$\log N$ ビットの内部状態を $\log N$ ビットのキーストリームに置き換える関数 f, h を定義する．

$$f, h : \log N \text{ bit state} - > \log N \text{ bit keystream},\ N = mt^2.$$

内部状態 $s_{i,0}$ を選択して，次の式を t 回繰り返し，$s_{i,t}$ を計算して，ペア $(s_{i,0}, s_{i,t})$ を得る．

$$s_{i,j+1} = g(s_{i,j}),\ g = h \circ f.$$

関数 h を t 通りランダムに選択し，内部状態 $s_{i,0}$ を m 通りランダムに選択して，ペア $(s_{i,0}, s_{i,t})$ のテーブルを t/D 個作成する．それぞれのテーブルは，ペア $(s_{i,0}, s_{i,t})$ について $s_{i,t}$ でソートしておくものとする．

D 個の出力列 $r_l, l = 0, 1, ..., D-1$ が得られたとき，t/D 個のテーブルの中にあるペア $(s_{i,0}, s_{i,t})$ から，次の関係を満たすものを探す．

$$\begin{aligned} g^x(h(r_i)) &= (g \circ \cdots \circ g)(h(r_i)) \\ &= s_{i,t}. \end{aligned}$$

バースデイパラドクスより，高い確率で該当のエントリーを見つけることができる．この関係が見つかった場合，次のように，出力列 r_i に対する内部状態 $g^{t-x-1}(s_{i,0})$ が得られる．

$$\begin{aligned} (g^x \circ h)(r_i) &= g^x(h(r_i)) \\ &= s_{i,t} \\ &= (g^x \circ h)(s_{i,0}) \\ &= (g^x \circ g \circ g^{t-x-1})(s_{i.0}) \\ &= (g^x \circ h)(f(g^{t-x-1}(s_{i,0}))). \end{aligned}$$

t/D 個のテーブルを用いており，それぞれのテーブルは m 個のエントリーを持つことから，$M = mt/D$ が得られる．D 個の出力列に一致する内部状態を探索する際，t/D 個のテーブルにそれぞれ t 回の繰り返し処理を行うので，$T = D(t/D)t = t^2$ が得られる．したがって，時刻とメモリのトレードオフの式は次のようになる．

$$TM^2D^2 = N^2,\ P = N/D,\ 1 \leq D^2 \leq T.$$

3.4.3 Biryukov らの TMTO 攻撃の KCipher-2 への適用

前節で示した Biryukov らの TMTO 攻撃 [14, 15] を KCipher-2 に適用した結果が提案論文 [2] では示されている．

内部状態のサイズを $s = 512$ とおくと，$N = 2^s = 2^{512}$, $D = 2^{s/4} = 2^{128}$, $P = 2^{3s/4} = 2^{384}$, $T = M = 2^{s/2} = 2^{256}$ のような関係が得られる．

以上のように，鍵サイズを $k = 128$，IV サイズを $v = 128$ とおくと，内部状態サイズ s は $s = 512 \geq 2(k + v) = 512$ を満足しており，十分な空間を有していると提案論文 [2] では述べられている．また，攻撃の計算量は $T = 2^{256}$ であることから，鍵空間の全数探索よりも大きく，攻撃は困難であると述べられている．

3.5 代数攻撃

KCipher-2 の提案論文 [2] では，内部状態とキーストリームに関する関係式を構成し，観測したキーストリームを代入し，これを解くことで内部状態を再構成する代数攻撃について，以下のように議論されている．

代数攻撃の基本的なアイデアは Courtois らの攻撃 [17, 18]（3.5.1 節）に基づいている．一般化した代数攻撃として Courtois の攻撃 [20] とこれを KCipher-2 に適用した結果（3.5.2 節）が示され，また SNOW 2.0 に対する代数攻撃として Billet らの攻撃 [21] とこれを KCipher-2 に適用した結果（3.5.3 節）が示されている．

3.5.1 Courtois らの代数攻撃

提案論文 [2] で記されている代数攻撃の基本的なアイデアは，Courtois らの攻撃 [17, 18] に基づいている．

内部状態を $s_0, s_1, \cdots, s_{n-1}$，キーストリームを $y^0, y^1, \cdots, y^{M-1}$ とおいたとき，次のような内部状態とキーストリームの次数 d の関係式 Q が得られる場合を考える．

$$Q(s_0, s_1, \cdots, s_{n-1}, y^0, y^1, \cdots, y^{M-1}) = 0.$$

ここで，内部状態の線形遷移関数を L，内部状態の初期値を K とおくと，連続する M 個の内部状態について，次のような関係式が成り立つ．

$$Q(L^t(K)_0, L^t(K)_1, \cdots, L^t(K)_{n-1}, y^t, y^{t+1}, \cdots, y^{t+M-1}) = 0.$$

キーストリーム $y^t, y^{t+1}, \cdots, y^{t+M-1}$ を観測して，これらを上記の式に代入すると，内部状態の初期値に関する多次多変数の連立方程式が得られる．連立方程式の解を得る方法としては，Buchberger アルゴリズム，XL [17]，Linearization[18] などが挙げられる．

例えば Linearization では，n ビットの内部状態を想定すると，それぞれの方程式には次数 d 以下の $T \approx \binom{n}{d}, d \leq n/2$ 個の単一項が存在し，これらを変数と見なす．約 $\binom{n}{d} + M$ ビットのキーストリームが得られると，連続する M ビットから $R = \binom{n}{d}, R > T$ 個の方程式が得られ，これにガウスの消去法を適用すると，計算量 $T^\omega, \omega \leq 2.376$[19] で解が得られる．

3.5.2 Courtois の一般化した代数攻撃の KCipher-2 への適用

内部状態の線形遷移関数とメモリ付きコンバイナを有するストリーム暗号を想定した代数攻撃として，Courtois の攻撃 [20] が示されており，提案論文 [2] ではこれを KCipher-2 に適用した結果が述べられている．

線形の内部状態の遷移関数を L とおくとき，時刻 t の内部状態 $s_0^t, s_1^t, \cdots, s_{n-1}^t$ は，次のような式で書ける．

$$\{s_0^t, s_1^t, \cdots, s_{n-1}^t\} = L^t(\{s_0^0, s_1^0, \cdots, s_{n-1}^0\}).$$

時刻 t の内部状態から k ビットを選択して，これらを $x_0^t, x_1^t, \cdots, x_{k-1}^t$ とおくとき，コンバイナ $F = (F_1, F_2)$ を次のように書く．

$$\begin{aligned}(y_0^t, y_1^t, \cdots, y_{m-1}^t) &= F_1(x_0^t, x_1^t, \cdots, x_{k-1}^t, a_0^{t-1}, a_1^{t-1}, \cdots, a_{l-1}^{t-1}),\\ (a_0^t, a_1^t, \cdots, a_{l-1}^t) &= F_2(x_0^t, x_1^t, \cdots, x_{k-1}^t, a_0^{t-1}, a_1^{t-1}, \cdots, a_{l-1}^{t-1}).\end{aligned}$$

F_1 に $x_0^t, x_1^t, \cdots, x_{k-1}^t$ と内部メモリ $a_0^{t-1}, a_1^{t-1}, \cdots, a_{l-1}^{t-1}$ を入力することで，キーストリーム $y_0^t, y_1^t, \cdots, y_{m-1}^t$ が得られる．F_2 に $x_0^t, x_1^t, \cdots, x_{k-1}^t$ と $a_0^{t-1}, a_1^{t-1}, \cdots, a_{l-1}^{t-1}$ を入力して，その出力を次の時刻の内部メモリの値とする．

このような構造に対し，次のような定理 1,2 が示されており，内部状態の初期値を得るための連立方程式の構成，Linearization による解の導出，各手続きの計算量，パラメータ選択について議論されている．

定理 1[20]：F は K ビットの入力，l ビットのメモリ，m ビットの出力のコンバイナとする．d と M は次の不等式を満たす整数とする．

$$2^{Mm} \sum_{i=0}^{d} \binom{Mk}{i} > 2^{Mk+l}.$$

このとき，M 個の連続する手続き/状態 $(t, t+1, \cdots, t+M-1)$ を考えると，x^t に関する次数 d の方程式 R が存在する．

$$R(x_0^t, \cdots, x_{k-1}^t, \cdots, x_0^{t+M-1}, \cdots, x_{k-1}^{t+M-1},\\ y_0^t, \cdots, y_{m-1}^t, \cdots, y_0^{t+M-1}, \cdots, y_{m-1}^{t+M-1}) = 0.$$

定理 2[20]：F は K ビットの入力，l ビットのメモリ，m ビットの出力のコンバイナとする．このとき，$M = \lceil (l+1)/m \rceil$ 個の連続する手続き/状態 $(t, t+1, \cdots, t+M-1)$ を考えると，これら手続き/状態について x^t と y^t のみを含む式が存在し，これは x^t に関して次数 $\lceil kM/2 \rceil = \lceil k \lceil (l+1)/m \rceil /2 \rceil$ を持つ．

連立方程式の構成に要する計算量は，大まかに $2^{\omega(Mk+l)}$ と見積もられており，$T = \binom{n}{d}$ ビットのキーストリームが与えられたとき，Linearization により連立方程式を解く計算量は，$T^\omega \approx 2^{\omega d \log n}$ と見積もられている．

これを KCipher-2 に対して適用すると，$n = 5 \cdot 32 + 11 \cdot 32 = 512, l = 4 \cdot 32 = 128, k = 6 \cdot 32 = 192, m = 2 \cdot 32 = 64$ であり，また定理 2 より，$M = \lceil (128+1)/64 \rceil = 3, d = \lceil kM/2 \rceil = 288$ が得られ，連立方程式を構成する計算量と Linearization の計算量は鍵の全数探索の計算量よりも大きくなる．

3.5.3 Billet らの代数攻撃の KCipher-2 への適用

KCipher-2 は FSR や非線形関数（FSM）の構造が SNOW 2.0[23] に近く，SNOW 2.0 の代数攻撃 [21] と同じアプローチを KCipher-2 に適用することが提案論文 [2] では試みられている．

DFC の出力ビット $cl1_t, cl2_t$ を一定の値で固定して FSR-B に常時クロックを与え，$GF(2^{32})$ 上の加算を XOR で置換えた簡単化した KCipher-2 を想定している．

簡単化した KCipher-2 の構造から，次のような式が得られる．

$$R2_t = R1_t \oplus A_{t+4} \oplus B_t \oplus z_t^L, R1_{t-1} = Sub(R2_{t-2} \oplus B_{t+2}),\\ R1_t = Sub(L1_{t-1} \oplus A_{t-1} \oplus B_{t+8} \oplus B_{t+9} \oplus z_{t-1}^H).$$

上記の式から $Sub()$ を削除できると仮定した場合，次のような線形再

帰が得られる．

$$
\begin{aligned}
R2_t &= R2_{t-2} \oplus A_{t-1} \oplus A_{t+4} \oplus B_t \oplus B_{t+2} \oplus B_{t+8} \\
&\quad \oplus B_{t+9} \oplus z^H_{t-1} \oplus z^L_t, \\
R1_t &= R2_{t-2} \oplus A_{t-1} \oplus A_{t+2} \oplus B_{t-2} \oplus B_{t+2} \oplus B_{t+8} \\
&\quad \oplus B_{t+9} \oplus z^H_{t-1} \oplus z^L_{t-2}.
\end{aligned}
$$

このとき，任意の時刻 t について，キーストリーム，FSR-A および FSR-B のレジスタ，非線形関数の内部メモリを含む，次のような式を定義できる．

$$
\begin{aligned}
R2_t &= R2_0 \bigoplus_{i=0}^{t} \epsilon^i_t z^H_i \bigoplus_{j=0}^{t} \epsilon^j_t z^L_j \bigoplus_{k=0}^{4} \epsilon^k_t A_k \bigoplus^{1} 0_{l=0} \epsilon^l_t B_l, \\
R1_t &= R1_0 \bigoplus_{i=0}^{t} \epsilon^i_t z^H_i \bigoplus_{j=0}^{t} \epsilon^j_t z^L_j \bigoplus_{k=0}^{4} \epsilon^k_t A_k \bigoplus^{1} 0_{l=0} \epsilon^l_t B_l,
\end{aligned}
$$

$R2_t = Sub(R1_{t-1})$ の関係を用いることで 2 次方程式を構成することができるが，上記のような $Sub()$ の削除を仮定しなければ，このような式が得られないので，ここで想定する KCipher-2 への攻撃適用は困難である．

フルバージョンの KCipher-2 では，攻撃者はさらに各サイクルにおいて $cl1_t, cl2_t$ を推測する必要があり，M を連立方程式の中の非定数の単一項の個数，N を 1 サイクルの出力毎に得られる式の個数とすると，攻撃の計算量が $2^2(\lceil M/N \rceil - 1)$ 倍になる．

以上のように，Courtois の代数攻撃 [20] の適用は，計算量の観点から鍵の全数探索よりも困難であり，また Billet らの代数攻撃 [21] の適用は，KCipher-2 の非線形関数の構造から適用が困難であると提案論文 [2] では述べられている．

3.6 GD 攻撃

KCipher-2 の提案論文 [2] では，キーストリームを観測して，内部状態の推測と決定を繰り返し，再構成する推測決定（GD）攻撃について，以下のように議論されている．

提案者らが示す一般化した GD 攻撃とこれを KCipher-2 に適用した結果（3.6.1 節），および提案者らが示す KCipher-2 の構造に注目した GD 攻撃とこれを KCipher-2 に適用した結果（3.6.2 節）が示されている．

3.6.1 提案者らの一般化した GD 攻撃の KCipher-2 への適用

提案論文 [2] では，提案者らの一般化した GD 攻撃とこれを KCipher-2 へ適用した結果が示されている．

ここでは，l ビットの内部状態から m ビットを抽出し，これをもとに n ビットをキーストリームとして出力するストリーム暗号を想定している．

キーストリームの n ビットから内部状態の $m-n$ ビットを推測してその他の n ビットを決定し，また内部状態の未推測，未決定の部分の値について，推測と決定を繰り返していく．

内部状態から一意に m ビットを選択，抽出できることを仮定して，上記の手続きを j 回繰り返したとき，推測または決定したビット数を v_j とすると，推測すべき内部状態のサイズは $(1-\lceil v_j/l \rceil)$ となる．

攻撃における x 回目の手続きにおいて，既に推測または決定されたビット数を $y(x)$ とおくと，次のような式が得られる．ただし，$y(0)=0$ とする．

$$y(x) = (n^2 - mn + lm)(1 - e^{-(m-n)x/l})/(m-n).$$

このとき，$y(\eta) = l$ とすると，推測と決定に要する計算量 C は，次のような式で書くことができる．c は定数とする．ただし，推測と決定を繰り返して得られた内部状態は，そこから生成したキーストリームが，観測したキーストリームと一致するかどうかにより正当性を確認するが，この計算量は除いている．

$$\begin{aligned} C &\approx c \cdot 2^{l-n\eta}, \\ \eta &\approx 1/(m-n) lnm/n. \end{aligned}$$

このような攻撃を KCipher-2 に適用すると，$l = 5 \cdot 32 + 11 \cdot 32 + 4 \cdot 32 = 640, m = 6 \cdot 32 + 2 \cdot 32 = 256, n = 2 \cdot 32 = 64$ であることから，その計算量のオーダは $O(2^{344})$ となる．

以上のように，提案者らが示した一般化した GD 攻撃を KCipher-2 に適用する場合，計算量のオーダは鍵の全数探索よりも大きくなることから，攻撃は成立しないことが提案論文 [2] では述べられている．また，KCipher-2 に対する単純な GD 攻撃として，FSR-A のレジスタと NLF の内部レジスタを全て推測して，FSR-B のレジスタを全て決定するという攻撃があるが，計算量のオーダは $O(2^{288})$ となり，攻撃が成立しないことも述べられている．

3.6.2 提案者らの GD 攻撃の KCipher-2 への適用

提案論文 [2] では，さらに，単純化した KCipher-2 を想定してた GD 攻撃とこれを KCipher-2 へ適用した結果が示されている．

単純化した KCipher-2 は，各フィードバック関数の $\alpha_i, i = 0, 1, 2, 3$ を除いて，$GF(2^{32})$ 上の加算を排他的論理和で置き換えたものを想定している．

NLF の構造に注目すると，次のような式が得られる．

$$\begin{aligned} z_t^L \oplus z_{t+4}^H &= (B_t \oplus Sub(R1_{t-1})) \oplus R1_t \\ &\quad \oplus (B_{t+14} \oplus Sub(L1_{t+3})) \oplus L1_{t+4}. \end{aligned}$$

上記の式の 5 つの要素を推測すると，例えば，B_{t+14} や A_{t+4} の値を決定できる．時刻 t を変化させ，少なくとも 10 個の要素を推測することで，$O(2^{320})$ の計算量のオーダで，FSR の全ての要素を決定することができる．

また，NLF の 4 つのレジスタ $R1, R2, L1, L2$ の関係に注目すると，次のような式が得られる．

$$\begin{aligned} R2_{t+1} &= Sub(R1_t),\ L1_{t+2} = Sub(R2_{t+1} \oplus B_{t+5}), \\ L2_{t+3} &= Sub(L1_{t+2}),\ R1_{t+4} = Sub(L2_{t+3} \oplus B_{t+12}). \end{aligned}$$

上記の式の $R1_t, B_{t+5}, B_{t+12}$ を推測すると，$R2_{t+1},\ L1_{t+2}, L2_{t+3}, R1_{t+4}$ を決定できる．単純化した KCipher-2 から FSR-A を除いた場合，それぞれのサイクルにおいて，$z_t^H \oplus A_t$ と $z_t^L \oplus A_{t+4}$ が得られる．このとき，6 個の要素 $R1_{t+1}, R1_{t+2}, L1_t, L1_{t+1}, B_{t+6},\ B_{t+7}$ を推測することで，計算量のオーダ $O(2^{192})$ で，FSR-B の全ての要素を決定できる．

以上のように，単純化した KCipher-2 に対して，NLF の構造に注目した GD 攻撃では計算量のオーダが $O(2^{320})$ となり，鍵の全数探索よりも大きくなるので攻撃が成立しないことが提案論文 [2] では述べられている．また，NLF の 4 つのレジスタの関係に注目した GD 攻撃では，計算量のオーダが $O(2^{192})$ となるが，フルバージョンの KCipher-2 では FSR-A の要素の他に DFC からの 2 ビットを推測する必要があり，GD 攻撃の適用は困難であると述べられている．

3.7 関連鍵/選択 IV 攻撃

KCipher-2 の提案論文 [2] では，一部だけ異なるような鍵/IV を用いてキーストリームを生成したときの内部状態の差分に基づいて鍵を推定す

る関連鍵/選択 IV 攻撃について，以下のように議論されている．

KCipher-2 では，初期化処理において，AES の鍵スケジューリングアルゴリズムを用いてラウンド鍵を作成して，これと IV を内部状態に設定し，24 サイクルだけ内部状態を更新してラウンド鍵と IV を含む内部状態を撹拌している．

13 サイクル更新すると，ラウンド鍵と IV が内部状態全体に広がり，さらに 11 サイクル更新されると，ラウンド鍵と IV が攪拌され，鍵や IV に関する差分を観測できないため，関連鍵/選択 IV 攻撃に対して脆弱でないと述べられている．

鍵または IV の差分パスを探索するとき，攻撃者はキーストリームを計算する任意のラウンドにおいて内部状態の差分を観測できる必要があるが，以上のように，初期化処理が十分に実施された場合，内部状態の差分の観測が困難になるため攻撃を適用できないことが主張されている．

3.8 統計的性質

KCipher-2 の提案論文 [2] および第三者評価書 [3] では，キーストリームの統計的性質について，NIST 乱数検定（SP800-22）[22] を適用した結果が述べられている．

NIST 乱数検定では，任意の長さのビット列の乱数性を確認するための次のような項目の検定が含まれている．

- The Frequency Test
- Frequency Test Within a Block
- The Runs Test
- Test for the Longest-Run-of-Ones in a Block
- The Binary Matrix Rank Test
- The Discrete Fourier Transform (Spectral) Test
- The Non-overlapping Template Matching Test
- The Overlapping Template Matching Test
- Maurer's Universal Statistical Test

- The Lempel-Ziv Compression Test
- The Linear Complexity Test
- The Serial Test
- The Approximate Entropy Test
- The Cumulative Sums Test
- The Random Excursions Test
- The Random Excursions Variant Tests

KCipher-2 の最大 2^{32} ビットのキーストリームに対して，NIST 乱数検定の全項目の検定をパスしたことが述べられている．

2008 年 12 月の改訂版で，The Lempel-Ziv Compression Test の除外，The Discrete Fourier Transform (Spectral) Test の閾値の変更，評価ソフトおよび評価方法の変更があったが，これについては言及されていない．

4　まとめ

本報告では，KCipher-2 の安全性について，KCipher-2 の提案論文 [2] および第三者評価書 [3, 4] をもとに，DFC 機構により制御される FSR の出力の周期，線形複雑度と，KCipher-2 の識別攻撃，相関攻撃，タイムメモリトレードオフ（TMTO）攻撃，代数攻撃，推測決定（GD）攻撃，関連鍵/選択 IV 攻撃，統計的性質について確認した．

DFC 機構により制御される FSR（FSR-B）の出力の周期と線形複雑度は，それぞれ上限が $(2^{352}-1)(2^{160}-1)$，$352(2^{160}-1)$ であることを確認した．

DFC 機構の出力ビットを $cl1_t = cl2_t = 0$ のように固定した KCipher-2 に対して，識別攻撃 [7, 8] を適用した場合，大きな偏差のマスク値は発見されず，distinguisher の構成が困難であることを確認した．

DFC 機構の出力ビット $cl1_t, cl2_t$ を固定した KCipher-2 に対して，FSR-B の初期値を得る相関攻撃 [11] を適用した場合，前処理の計算量は 2^{256}，主処理の計算量は 2^{266} となり，攻撃は成立しないことを確認した．

KCipher-2 に対してタイムメモリトレードオフ攻撃 [14, 15] を適用した場合，内部状態は十分な空間を有しており，また計算量は 2^{256} であることから，攻撃は成立しないことを確認した．

KCipher-2 に対して一般化された代数攻撃 [20] を適用した場合，連立方程式を構成する計算量と Linearization の計算量は鍵の全数探索の計算量よりも大きく，攻撃は成立しないことを確認した．また，DFC 機構の出力ビット $cl1_t, cl2_t$ を一定の値で固定して，$GF(2^{32})$ 上の加算を XOR で置換えた KCipher-2 に対して，代数攻撃 [21] を適用した場合，NLF の構造により方程式を構成できず攻撃は成立しないことを確認した．

提案論文 [2] において示された一般化した GD 攻撃を KCipher-2 に適用すると計算量のオーダが $O(2^{344})$ となり，鍵の全数探索よりも大きくなり攻撃が成立しないことを確認した．また，単純な GD 攻撃（FSR-A のレジスタと NLF の内部レジスタを全て推測して，FSR-B のレジスタを全て決定する）を KCipher-2 に適用すると，計算量のオーダが $O(2^{288})$ となり，同様に，攻撃が成立しないことを確認した．各フィードバック関数の $\alpha_i, i = 0, 1, 2, 3$ を除いて，$GF(2^{32})$ 上の加算を排他的論理和で置き換えた KCipher-2 に対して，提案論文 [2] にある GD 攻撃を適用した場合，計算量のオーダが $O(2^{192})$ となることを確認した．

KCipher-2 に対して関連鍵/選択 IV 攻撃を適用することに関しては，初期化処理において，AES の鍵スケジューリングアルゴリズムを用いてラウンド鍵を作成して，これと IV を内部状態に設定し，24 サイクルだけ内部状態を更新してラウンド鍵と IV を含む内部状態を撹拌しているため，攻撃は困難であることを確認した．

NIST 乱数検定（SP800-22）[22] を KCipher-2 のキーストリームに対して適用した場合，全ての検定項目をパスすることが述べられている．

参考文献

[1] S. Kiyomoto, T. Tanaka, and K. Sakurai, “A Word-Oriented Stream Cipher Using Clock Control,” Proc. of SASC2007, pp.260-273, 2007.

[2] S. Kiyomoto, T. Tanaka, and K. Sakurai, “K2: A Stream Cipher Algorithm Using Dynamic Feedback Control,” Proc. of SECRYPT2007, pp.204-213, 2007.

[3] Royal Holloway Enterprises Ltd., "Provisional Study of the Properties of Dynamic Linear Feedback Shift Registers-I,II," third-party eval. report, 2009.

[4] General Secretary, Cryptology Research Society of India, "Evaluation of the Word-Oriented Stream Cipher: K2," third-party eval. report, 2009.

[5] G. Gone, "Theory and Applications of q-any Interleaved Sequences," IEEE Trans. on Information Theory, vol.41, no.2, pp.400-411, 1995.

[6] D. Coppersmith, S. Halevi, and C. Jutla, "Cryptanalysis of stream ciphers with linear masking," Advances in Cryptology, CRYPT2002, LNCS2442, pp.515-532, 2002.

[7] D. Watanabe, A. Biryukov, and C.D. Canniere, "A Distinguishing Attack of SNOW 2.0 with Linear Masking Method," In Proc. of SAC2003, LNCS3006, pp.222-233, 2004.

[8] K. Nyberg and J. Wallen, "Improved linear distinguishers for SNOW 2.0," In Proc. of FSE2006, LNCS4047, pp.144-162, 2006.

[9] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," IEEE Trans. Comput., vol.C-34, no.1, pp.81-85, 1985.

[10] R.G. Gallager, "Information Theory and Reliable Communications," JohnWiley and Sons, Inc. New York, London, Sydney, Toronto, 1968.

[11] V.V. Chepyzhov, T. Johansson, and B. Smeets, "A simple algorithm for fast correlation attacks on stream ciphers," FSE2000, LNCS1978, pp.181-195, 2001.

[12] S.H. Babbage, "Improved exhaustive search attacks on stream ciphers," European Convention on Security and Detection, IEE Conference publication, no.408, pp.161-166, 1995.

[13] J.Dj. Golic, "Cryptanalysis of alleged A5 stream cipher," Eurocrypt'97, LNCS1233, pp.239-255, 1997.

[14] A. Biryukov and A. Shamir, "Cryptanalytic time/memory/data tradeoffs for stream ciphers," Asiacrypt2000, LNCS1976, pp.1-13, 2000.

[15] J. Hong and P. Sarker, "Rediscovery of time memory tradeoffs," IACR ePrint Archive, Report 2005/090, 2005.

[16] M.E. Hellman, "A cryptanalytic time-memory trade-off," IEEE Trans. on Infor. Theory, pp.401-406, 1980.

[17] N. Courtois, "Higher Order Correlation Attacks, XL Algorithm and Cryptanalysis of Toyocrypt," ICISC2002, LNCS2587, pp.182-199, 2002.

[18] N. Courtois and W. Meiter, "Algebraic Attacks on Stream Ciphers with Linear Feedback," Eurocrypt2003, LNCS2656, pp.345-359, 2003.

[19] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," J. Symbolic Computation, pp.251-280, 1990.

[20] N. Courtois, "Algebraic attacks on combiners with memory and several outputs," In Proc. of ICISC2004, LNCS3506, pp.3-20, 2005.

[21] O. Billet and H. Gilbert, "Resistance of SNOW 2.0 against algebraic attacks," In Proc. of CT-RSA2005, LNCS3376, pp19-28, 2005.

[22] NIST, Statistical Tests, Cryptographic Toolkit, available at http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software .html

[23] P. Ekdahl and T. Johansson, "A new version of the stream cipher SNOW," Proc. of SAC2002, LNCS 2595, pp.47-61, 2002.

[24] J. Daemen and Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Springer, 2002.