Computer Security Symposium 2021
26 - 29 October 2021

Modular差分を用いたストリーム暗号ZUC-256の解析

堀部 佳吾[1,a)]　Fukang Liu[1]　五十部 孝典[1,2,3]

概要：ZUC-256 は 5G アプリケーションのために設計されたストリーム暗号で，現在，5G モバイル通信における標準的なアルゴリズムの評価が進められている．ZUC-256 では，LFSR(Linear Feedback Shift Register) は $GF(2^{31}-1)$ 上で，FSM(Finite State Machine) は $GF(2^{32})$ 上で定義されている．既存の解析結果では，Babbage と Maximov により初期化フェーズの 28 段に対する XOR 差分を用いた識別攻撃が提案されている．本論文では，初期化フェーズに対して，Modular 差分を用いることで，この攻撃が改良できることを示す．ZUC-256 内部には Modular 加算があるため，Modular 差分で解析することで，差分の広がりを XOR 差分と比較して抑えることが可能である．また，差分を入れる位置を適切に選ぶことで，確率的に差分のキャンセルイベントを発生させ，差分の伝搬を制限する．その結果，28 ラウンドの ZUC-256 に対して Babbage と Maximov の $2^{-10.46}$ よりも大きな $2^{-4.39}$ のバイアスを得ることができた．

キーワード：ストリーム暗号，ZUC-256，差分攻撃, modular 差分

Analysis of the Stream Cipher ZUC-256 by Modular Difference

Keigo Horibe[1,a)]　Fukang Liu[1]　Takanori Isobe[1,2,3]

Abstract: ZUC-256 is a stream cipher designed for 5G applications and is currently being under evaluation for standardized algorithms in 5G mobile telecommunications. A feature of ZUC-256 is that the LFSR (Linear Feedback Shift Register) is defined in $GF(2^{31}-1)$, while the FSM (Finite State Machine) is defined in $GF(2^{32})$. Recently, Babbage and Maximov proposed a distinguishing attack on 28-round ZUC-256 with the XOR difference. We show that Babbage-Maximov's attack can be improved with modular differences. This is because in the round update function of ZUC-256, many additions modulo $2^{31}-1$ are involved and the modular difference linearly propagates through the modular addition. Moreover, by properly selecting the input modular difference, we are able to slow down the propagation of the difference. Consequently, we obtain a linear relation with a bias of $2^{-4.39}$, which is in terms of the state word in LFSR after 28 rounds. This improves the bias $2^{-10.46}$ in Babbage-Maximov's attack.

Keywords: stream cipher, ZUC-256, differential attack, modular difference

1. はじめに

ZUC-128 は，LTE ネットワークの 3GPP Confidentiality and Integrity Algorithms UEA3 & UIA3 の標準のストリーム暗号である．2010 年に UEA3 & UIA3 の候補として提案され，ZUC ワークショップを経て，最終的に 3GPP SA3 において LTE 規格に新たに採用された [1,2]．2018 年 1 月，3GPP からの 5G の 256 ビットセキュリティレベルの要求を満たすために，ZUC-128 の 256 ビット版として ZUC-256 が発表された [3]．ZUC-128 と比較して，ZUC-256 の構造は変わらず，初期化フェーズとメッセージ認証コード生成フェーズのみが 256 ビットのセキュリティレベルに合わせて改良された．ZUC-256 の構造は大きく分けて 3 つの処理層に分れており，線形性を持つ LFSR(Linear Feedback Shift Register) 層，非線形性を持つ FSM(Finite

[1] 兵庫県立大学, University of Hyogo, Japan.
[2] 国立研究開発法人 情報通信研究機構, National Institute of Information and Communications Technology, Japan.
[3] 国立研究開発法人 科学技術振興機構, PRESTO, Japan Science and Technology Agency, Japan.
[a)] horibe.u.hyogo@gmail.com

表 1　本論文の結果

研究手法	攻撃ラウンド数	確率
既存研究 [4]	28 Round	$\frac{1}{2}-2^{-10.46}$
本研究	28 Round	$\frac{1}{2}-2^{-4.39}$

State Machine) 層，また，LFSR 層から FSM 層にビットを入れるための BR(Bit Reorganization layer) 層が存在する．一般的に ZUC-256 において，線形性を持つ LFSR と呼ばれる処理層は $GF(2^{31}-1)$ 上で定義されているのに対し，非線形性を持つ FSM 層は $GF(2^{32})$ 上で定義されており異なるため，ストリーム暗号に対する標準的な線形解読法が扱えない．既存の暗号解析では，初期化の全 33 ラウンドのうち 28 ラウンドで $2^{-10.46}$ のバイアスが Babbage と Maximov により見つかっている [4]．Babbage らの攻撃 [4] では，初期化フェーズの攻撃についてブラックボックス攻撃であり，数学的根拠から求めてはいない．

本論文では，初期化フェーズに対し，2 つのアプローチを用いて，Babbage と Maximov の攻撃が改良できることを示す．1 つ目のアプローチは XOR 差分ではなく，Modular 差分を用いることである．Modular 差分は XOR 差分と比較し，Modular 加算という条件下で差分を操作することができる．ZUC-256 内部には Modular 加算があるため，Modular 差分で解析することで，ZUC-256 の内部の差分をコントロールすることができ，差分の広がりを XOR 差分と比較して抑えることが可能である．

2 つ目のアプローチは内部構造の特性を利用して適切な位置に条件にあった差分を入れることである．適切な位置に差分を入れることで確率的に差分のキャンセルイベントを発生させ，有限体が異なる FSM 層への差分の伝搬を防ぐことができる．結果として，表 1 のように 28 ラウンドの ZUC-256 に対して Babbage と Maximov の $2^{-10.46}$ よりも大きな $2^{-4.39}$ のバイアスを発見した．

このバイアスを利用することで，$2^{10.78}$ の計算量とデータ量で初期化フェーズの識別攻撃が可能となる．

本論文の構成は以下の通りである．2 章では ZUC-256 の構造を示す．具体的には，最初に ZUC-256 の概要を説明した上で各層の処理，初期化フェーズ，そして既存の ZUC-256 の解析手法について示す．3 章では提案手法として，Modular 差分と入力差分の適切な選択について示す．4 章では Modular 差分を利用した暗号解析について差分とバイアスの探索方法を示した上で 5 章で結果を示す．最後に 6 章で本稿を締めくくる．

2. ZUC-256 の仕様

論文で使われる表記と ZUC-256 の説明をする．

2.1 表記

本論文で扱う表記について説明する．

- $\boxplus$ は　$\bmod 2^{32}$ の加算である．
- $+$ は $\bmod\ \bmod 2^{31}-1$ の加算である．
- 鍵 K は 256 ビットであり，$K=(K_{31,}K_{30,\cdots}K_{2,}K_{1,}K_0)$ で表記．
- 初期ベクトル IV は 184 ビットで，$IV=(IV_{24},IV_{23},\cdots IV_{17},IV_{16},IV_{15,}\cdots,IV_1,IV_0)$．$IV_i$ について $0\le i\le 16$ は 8 ビット，$17\le i\le 24$ は 6 ビットで構成．
- 定数 d_i は 7 ビットの定数である．
- $\lll$ は 64 ビットオペランドにおける左回転である．
- $||$ はビットの連結を指す．
- $|$ はビット単位の論理和である．
- a_H は整数 a の上位 15 ビットである．
- a_L は整数 a の下位 15 ビットである．
- $Pr(a)$ は条件 a の確率である．
- $\cdot$ は $GF(p)$ の整数乗算である．
- a^{-1} は $a\cdot a^{-1}=1\pmod 2^{31}-1$ の関係がある．
- $X[a:b]$ は X の a ビット目から b ビット目である．
- $a[i]$ は a の i ビット目である．
- Δa は $\Delta a=a\oplus a'$ の関係がある．

2.2 ZUC-256 概要

ZUC-256 は，256 ビットの鍵 K と 128 ビットの初期ベクトル IV を入力として，鍵ストリームと呼ばれるシーケンスを生成する．ZUC-256 の初期化フェーズ全体像を図 1 に示す．ZUC-256 の構造は大きく分けて 3 つの処理層に分れており，線形性を持つ LFSR 層，非線形性を持つ FSM 層，そして，その間に LFSR 層から FSM 層にビットを入れるための BR 層が存在する．

2.2.1 LFSR 層

LFSR 層から順に説明する．LFSR 層内の 16 つのセル $(s_{15},s_{14},\cdots,s_1,s_0)$ の初期設定は以下の式 (1) の通りである．

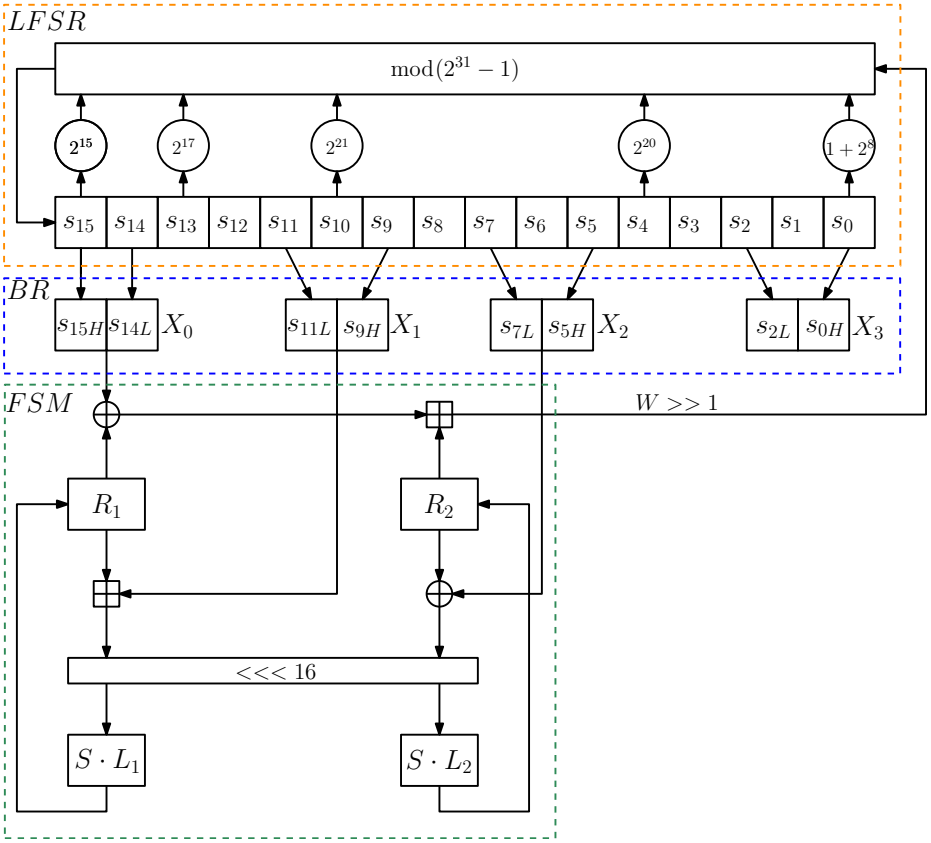

図 1　ZUC-256 の初期化フェーズ全体像

$$
\begin{aligned}
s_0 &= K_0||d_0||K_{21}||K_{16}\\
s_1 &= K_1||d_1||K_{22}||K_{17}\\
s_2 &= K_2||d_2||K_{23}||K_{18}\\
s_3 &= K_3||d_3||K_{24}||K_{19}\\
s_4 &= K_4||d_4||K_{25}||K_{20}\\
s_5 &= IV_0||(d_5|IV_{17})||K_5||K_{26}\\
s_6 &= IV_1||(d_6|IV_{18})||K_6||K_{27}\\
s_7 &= IV_{10}||(d_7|IV_{19})||K_7||IV_2\\
s_8 &= K_8||(d_8|IV_{20})||IV_3||IV_{11}\\
s_9 &= K_9||(d_9|IV_{21})||IV_{12}||IV_4\\
s_{10} &= IV_5||(d_{10}|IV_{22})||K_{10}||K_{28}\\
s_{11} &= K_{11}||(d_{11}|IV_{23})||IV_6||IV_{13}\\
s_{12} &= K_{12}||(d_{12}|IV_{24})||IV_7||IV_{14}\\
s_{13} &= K_{13}||d_{13}||IV_{15}||IV_8\\
s_{14} &= K_{14}||(d_{14}|(K_{31})_H^4)||IV_{16}||IV_9\\
s_{15} &= K_{15}||(d_{15}|(K_{31})_L^4)||K_{30}||K_{29}
\end{aligned}
\tag{1}
$$

ここでの $(K_{31})_H^4$ はセル K_{31} において，上位 4 ビットであり，逆に $(K_{31})_L^4$ は下位 4 ビットである．LFSR 層内では，496 ビットを 16 つのセルに分けて $(s_{15}, s_{14}, \cdots, s_1, s_0)$ 代入している． 時間 $t = 0, 1, \cdots$ についてのセルを s_i^t とすると，$t+1$ と t の各セルの関係は以下の式 (2) ようになる．

$$
\begin{aligned}
W &= (X_0 \oplus R_1^{(t)}) + R_2^{(t)}\\
s_{15}^{(t+1)} &= (W \gg 1) \boxplus 2^{15} \cdot s_{15}^{(t)} + 2^{17} \cdot s_{13}^{(t)}\\
&\quad + 2^{21} \cdot s_{10}^{(t)} + 2^{20} \cdot s_7^{(t)}\\
&\quad + \left(1 + 2^8\right) \cdot s_0^{(t)}
\end{aligned}
\tag{2}
$$

$s_{15}^{(t+1)}$ について，仮に 0 である場合，値は p となる．その他のセルは以下の式 (3) となる

$$
\begin{aligned}
&(s_{14}^{(t+1)}, s_{13}^{(t+1)}, \cdots s_1^{(t+1)}, s_0^{(t+1)})\\
&\qquad = (s_{15}^{(t)}, s_{14}^{(t)}, \cdots s_2^{(t)}, s_1^{(t)})
\end{aligned}
\tag{3}
$$

2.2.2　BR 層

BR 層は，LFSR 層と FSM 層の接続層である．LFSR 層から 128 ビットを取り出し，4 つの 32 ビットワードである X_0，X_1，X_2，X_3 ををを形成する．式は以下の式 (4) である．

$$
\begin{aligned}
X_0 &= s_{15H}||s_{14L}\\
X_1 &= s_{11L}||s_{9H}\\
X_2 &= s_{7L}||s_{5H}\\
X_3 &= s_{2L}||s_{0H}
\end{aligned}
\tag{4}
$$

2.2.3　FSM 層

メモリーとして，2 つの 32 ビットである R_1，R_2 を X_1，X_2 から生成する．

$$
\begin{aligned}
W_1 &= R_1^{(t)} \boxplus X_1 \\
W_2 &= R_2^{(t)} \boxplus X_2 \\
R_1^{(t+1)} &= S(L_1(W_{1L}||W_{2H})) \\
R_2^{(t+1)} &= S(L_2(W_{2L}||W_{1H}))
\end{aligned} \tag{5}
$$

ここで，S は，$S = (S_0, S_1, S_2, S_3)$ の 4 つの並列 8 ビットの S-box であり，L_1，L_2 は 2 つの $GF(2^{32})$ の線形変換である．

2.2.4 初期化フェーズの処理

初期化フェーズは $32+1=33$ ラウンドあり，$0 \le t \le 31$ の 32 ラウンドの各層の更新は以下のようになる．

- LFSR 層

$$
\begin{aligned}
&W = (X_0 \oplus R_1^{(t)}) \boxplus R_2^{(t)} \\
&s_{15}^{(t+1)} = (W \gg 1) \boxplus 2^{15} \cdot s_{15}^{(t)} + 2^{17} \cdot s_{13}^{(t)} \\
&\qquad + 2^{21} \cdot s_{10}^{(t)} + 2^{20} \cdot s_7^{(t)} + \left(1 + 2^8\right) \cdot s_0^{(t)} \\
&(s_{14}^{(t+1)}, s_{13}^{(t+1)}, \cdots s_1^{(t+1)}, s_0^{(t+1)}) = (s_{15}^{(t)}, s_{14}^{(t)}, \cdots s_2^{(t)}, s_1^{(t)})
\end{aligned}
$$

- BR 層

$$
\begin{aligned}
X_0 &= s_{15H}^t || s_{14L}^t \\
X_1 &= s_{11L}^t || s_{9H}^t \\
X_2 &= s_{7L}^t || {s_{5H}}^t \\
X_3 &= s_{2L}^t || s_{0H}^t
\end{aligned}
$$

- FSM 層

$$
\begin{aligned}
W_1 &= R_1^{(t)} \boxplus X_1 \\
W_2 &= R_2^{(t)} \boxplus X_2 \\
R_1^{(t+1)} &= S(L_1(W_{1L}||W_{2H})) \\
R_2^{(t+1)} &= S(L_2(W_{2L}||W_{1H}))
\end{aligned}
$$

$t = 32$ の各層の更新は以下のようになる．

- LFSR 層

$$
\begin{aligned}
&s_{15}^{(t+1)} = 2^{15} \cdot s_{15}^{(t)} + 2^{17} \cdot s_{13}^{(t)} \\
&\qquad + 2^{21} \cdot s_{10}^{(t)} + 2^{20} \cdot s_7^{(t)} + \left(1 + 2^8\right) \cdot s_0^{(t)} \\
&(s_{14}^{(t+1)}, s_{13}^{(t+1)}, \cdots s_1^{(t+1)}, s_0^{(t+1)}) = (s_{15}^{(t)}, s_{14}^{(t)}, \cdots s_2^{(t)}, s_1^{(t)})
\end{aligned}
$$

- BR 層

$$
\begin{aligned}
X_0 &= s_{15H}^t || s_{14L}^t \\
X_1 &= s_{11L}^t || s_{9H}^t \\
X_2 &= s_{7L}^t || s_{5H}^t \\
X_3 &= s_{2L}^t || s_{0H}^t
\end{aligned}
$$

- FSM 層

$$
\begin{aligned}
W_1 &= R_1^{(t)} \boxplus X_1 \\
W_2 &= R_2^{(t)} \boxplus X_2 \\
R_1^{(t+1)} &= S(L_1(W_{1L}||W_{2H})) \\
R_2^{(t+1)} &= S(L_2(W_{2L}||W_{1H}))
\end{aligned}
$$

このように最終ラウンドは他のラウンドと比べ，LFSR 層での W の処理がされない．

2.3 既存の ZUC-256 の解析手法

既存の解析結果としては，Babbage と Maximov により初期化フェーズの 28 段に対する XOR 差分を用いた識別攻撃が提案されている．この攻撃では，初期化フェーズをブラックボックスにして，固定された XOR 差分を入力とし，WHT(Walsh-Hadamard Transform) を利用して網羅的に良い近似バイアスを見つけていく手法である．

初期化フェーズにおいて，式 (3) より，t ラウンド後に LFSR 層の s_{15} に大きな差分が見つかった後に，その差分は $t+15$ ラウンド後の s_0 にも現れる．よって，s_{15} について調べ，できるだけ多くのラウンドでそこにあるバイアスを見る．

1 から 5 ビットの Key ビットに XOR 差分を固定し，(Key, IV) を用意して，$t = 1, 2, \cdots$ ラウンドの ZUC-256 の初期化フェーズをシミュレーションし，$\Delta s_{15}^{(t)} = s_{15}^{(t)} \oplus {s_{15}^{(t)}}'$ の 31 ビットの多次元分布 $\Delta D^{(t)}$ を構築する．しかし，すべての (key, IV) を使うのは時間がかかるため不可能である．そこで，以下の式ように 31 ビットの多次元分布 $\Delta D^{(t)}$ を使って，良いバイナリ近似を探索することができる．

$$
Pr\{L \cdot \Delta(s_{15}^{(t)} = s_{15}^{(t)} \oplus {s_{15}^{(t)}}') = 1\} = (1 - \mathcal{W}(\triangle D^{(t)})_L)/2
$$

ここで，L は 31 ビットの線形マスク，$\mathcal{W}(\triangle D^{(t)})_L$ は L における分布 $\Delta D^{(t)}$ の WHT の値である．良い線形マスク L を見つけるためには，分布 $\Delta D^{(t)}$ の WHT を計算量 $O(31 \cdot 2^{31})$ で取得し，$O(2^{31})$ でスペクトルをループして，絶対値が最大となる非ゼロの L を見つければよい．バイアスが 2^{-q} の場合，以下の式として書くことができる．

$$
Pr\{L \cdot \Delta(s_{15}^{(t)} = s_{15}^{(t)} \oplus {s_{15}^{(t)}}') = 1\} \approx \frac{1}{2} \pm 2^{-q}
$$

また，以下の式を満たせば，バイアスが検出されたと判断する．

$$
N \ge 2^{2q+4}
$$

バイアスが検出されたものについて，より多くのサンプルを収集して検証を行う．その結果，全 33 ラウンドのうち 28 ラウンドで $2^{-10.46}$ のバイアスが発見されている．

3. 提案手法

本章では，2 つのアプローチとして，まず Modular 差分について，XOR 差分との違いを示す．その後，内部構造

の特性を利用して適切な位置に条件にあった差分の選択について示す．

3.1 Modular 差分の導入

Modular 差分と XOR 差分の違いを説明する．具体的には，$a+b$ という計算について考える．a の XOR 差分を $\Delta a = 0\text{x}1$，$\Delta b = 0\text{x}1$ とする．ここで，以下の式は確率的に成り立つ．

$$a+b = (a \oplus \Delta a) + (b \oplus \Delta b)$$

しかし，ここで $(a[0]=0, b[0]=1)$ と限定すると，加算は $\bmod 2^{31}-1$ であるため，以下の式が成り立つ

$$(a \oplus 0\text{x}1) - \text{a} = 2^0$$
$$(b \oplus 0\text{x}1) - \text{b} = 2^{31} - 1 - 2^0$$

これは，$(a[0], b[0])$ の条件で $(a \oplus \Delta a) + (b \oplus \Delta b) = a + 2^0 + (b + 2^{31} - 1 - 2^0) = a + b$ であるため，$a+b = (a \oplus \Delta a) + (b \oplus \Delta b)$ が確率 1 で成立する．

実際，上記の例は Modular 差分に関係している．ここで，$GF(2^{31}-1)$ の Modular 差分について，(a', a) 間の Modular 差分を δa とする．$a+b$ という計算を考えると，$\delta a + \delta b = 0$ のとき，$(a+\delta a) + (b+\delta b) = a+b$ が必ず存在する．

このことから，XOR 差分は $a+b$ の差が必ず打ち消されるとは限らず，キャンセルできるかどうかは (a, b) の実際の値に依存するのに対し，Modular 差分は a と b に差があれば，$a+b$ の差は確率 1 でキャンセルすることができる．

3.2 入力差分の適切な選択

Modular 差分（以降，差分とする）を入力する場所について，時間を考慮し，FSM 層の影響がないような場所にすることで差分に広がりを抑える．図 1 より，s_7 から s_{15} に差分を入れると，数ラウンドで FSM 層に差分の影響を与える．よって，s_{6L} に差分を入れることで FSM 層による差分の処理を避けることができる．しかし，このままでは 3 ラウンド後に s_{15} に差分がでてきる．その差分を打ち消すために s_2 に差分を入れる．以上のことから，s_{6L} と s_2 に差分を入れることにする．

各時間の LFSR 層内の差分について考える．時間 t において，s_2 と s_6 に差分があるとする．その時，s_2 と s_6 の差分 $diff_{s_2}(t)$，$diff_{s_6}(t)$ は式 (3) より 2 ラウンド後での差分は以下のようになる．

$$diff_{s_0}(t+2) = diff_{s_2}(t)$$
$$diff_{s_4}(t+2) = diff_{s_6}(t)$$

また，次のラウンドは式 (2) より以下の関係式がある．

$$diff_{s_{15}}(t+3) = 2^{20} \cdot diff_{s_4}(t+2) + (1+2^8) \cdot diff_{s_0}(t+2) \quad (6)$$

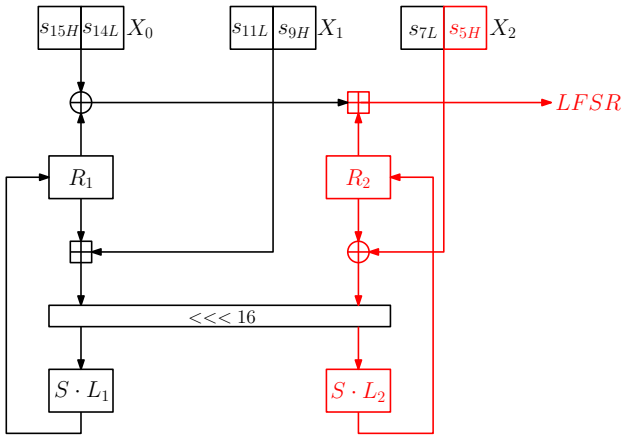

図 2 FSM 層の差分分布

まとめると，以下の式 (7) になる．

$$diff_{s_{15}}(t+3) = 2^{20} \cdot diff_{s_6}(t) + (1+2^8) \cdot diff_{s_2}(t) \quad (7)$$

ここで，$diff_{s_{15}}(t+3)$ の値が 0 になれば，式 (3) よりセルが移動していくため，LFSR 層での s_{15}^{t+3} の差分を削減することができる．

4. 探索方法

本章では，差分の探索方法について探索の条件を示す．その後，条件にあった差分を入力とし，バイアスを探索する．その探索方法について示す．

4.1 差分の探索方法

差分の探索方法について探索の条件について具体的に考える．条件として，前章で示したように $diff_{s_{15}}(t+3)$ の値が 0 になれば良い．これからどのように値を 0 にするか具体的に説明していく．前章での式 (7) について，$diff_{s_2}(t)$，$diff_{s_6}(t)$ をぞれぞれ X，2^α とすると，以下の関係式 (8) を満たす α を見つける．

$$2^{20} \cdot 2^\alpha + (1+2^8) \cdot X = 0 \quad (8)$$

α の探索については，$0 \le \alpha \le 14$ の範囲で行う．ここで，FSM 層の差分分布について図 2 に示す．赤色の箇所が差分がある場所である．図 2 では，FSM 層の R_2 に差分を入れないために，式 (4) の X_2 の式の s_{5H} には差分を入れないようにしなければ行けない．ここで，式 (3) より $s_5^{t+1} = s_6^t$ であるので，差分のある s_6 について s_{6H} ではなく，s_{6L} に差分が入れればよい．ゆえに $0 \le \alpha \le 14$ の範囲でなければならない．式 (8) は $GF(p)$ であるので，式 (9) に置き換えることができる．

$$X = \{p - (2^{20} \cdot 2^\alpha)\} \cdot (1+2^8)^{-1} \quad (9)$$

ここで，X の 16 ビット目から 22 ビット目である $X[16:22]$ に注目する．式 (1) での s_2 において，16 ビット目から 22

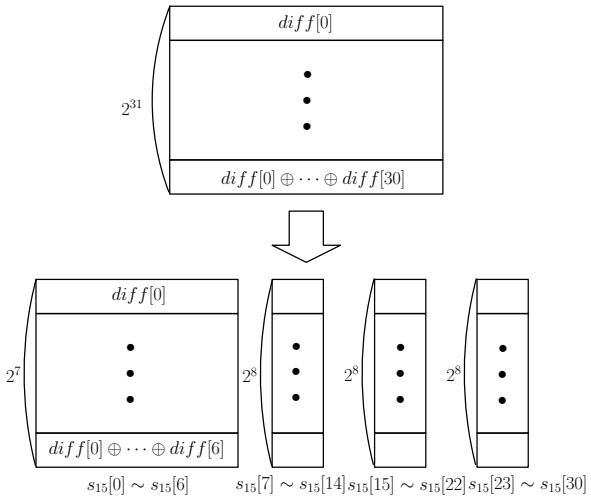

図 3 計算量を考慮した探索

ビット目箇所は d であり，定数であるために差分は必ず存在しない．ゆえに $X[16:22]$ の値は 0x00 か 0x7F になる．

$$X[16:22] = 000\ 0000\ or\ 111\ 1111 \qquad (10)$$

この式 (10) の条件も考慮して α を決定する．そして，X，2^α について，式 (1) での s_2，s_6 から鍵の入力差分を決定する．

4.2 バイアスの探索方法

複数回で見つけた差分を固定入力として，(Key, IV) の 2 つを用意し，13 ラウンドの s_{15} の差分の確率を求める．そこからバイアスを求める．また，13 ラウンドの s_{15} の差分の確率を調べる際の探索方法についても示す．

探索回数を N とした時，1 回ごとに差分がある 2 つの鍵と初期ベクトルを入力とする．初期化フェーズで $s_{15}^{(n)}$,$s_{15}^{(n)\prime}$ の差分 $diff_n$ の 31 ビットを見ていく．式は以下のようになる．

$$\begin{aligned} diff_0 =& s_{15}^{(0)\prime} - s_{15}^{(0)} \\ diff_1 =& s_{15}^{(1)\prime} - s_{15}^{(1)} \\ &\vdots \\ diff_N =& s_{15}^{(N)\prime} - s_{15}^{(N)} \end{aligned}$$

そして，差分 $diff_n$ の 31 ビットについて各ビットの組み合わせを考え，組み合わせの XOR が 0 になる個数を調べ確率を求める．例えば，9 ビットと 10 ビットの場合であれば，以下のような式となる．

$$Pr\left(diff\left[9\right] \oplus diff\left[10\right] = 0\right) = \frac{counter}{N} = \frac{1}{2} \pm 2^A$$

ここでの $counter$ は 0 になる個数であり，A が大きい数値ほどバイアスが大きく良い結果となる．また，各ビットの組み合わせを考える時，図 3 のように 1 つのテーブルでは計算量が WHT[4] を使うと，$O(31 \times 2^{31})$ で非常に時間がかかる．よって，テーブルを 4 つにする．これより，式 (5) の S-box について考慮する S-box の数を削減することができる．

表 2 α と X の関係

α	X	$X[16:22]$
0	0x700FE01F	000 1111
1	0x601FC03F	001 1111
2	0x403F807F	011 1111
3	0x007F00FF	111 1111
4	0x00FE01FE	111 1110
5	0x01FC03FC	111 1100
6	0x03F807F8	111 1000
7	0x07F00FF0	111 0000
8	0x0FE01FE0	110 0000
9	0x1FC03FC0	100 0000
10	0x3F807F80	000 0000
11	0x7F00FF00	000 0000
12	0x7E01FE01	000 0001
13	0x7C03FC03	000 0011
14	0x7807F807	000 0111

5. 結果

5.1 差分の探索結果

前章で求めた件にあった差分について α ごとに求めていく．差分の探索で求めた α と X と X の 16 ビットから 22 ビットである $X[16:22]$ を表 2 に示す．表 2 について，X の 16 ビットから 22 ビットが 0x00 か 0x7F である α は 3，10，11 の時である．ここから各 α について，鍵の入力差分を式 (1) から計算する．

(1) $\alpha = 3$ の時

$$\begin{aligned} X =& (2^{22} + \cdots + 2^{16}) + (2^7 + \cdots + 2^0) \\ =& (2^{23} - 2^{16}) + (2^8 - 2^0) \\ =& (2^{23} - 2^{16}) + (2^{16} - \cdots - 2^8) - 2^0 \\ =& 2^{23} + (2^{15} - \cdots - 2^8) - 2^0 \\ \leftrightarrow& \begin{cases} K_2[0] = 0, \Delta K_2 = 0\text{x}01 \\ K_{23} = 0\text{xFF}, \Delta K_{23} = 0\text{xFF} \\ K_{18}[0] = 1, \Delta K_{18} = 0\text{x}01 \end{cases} \\ 2^\alpha =& 2^3 \\ \leftrightarrow& K_{27}[3] = 0, \Delta K_{27} = 0\text{x}08 \end{aligned}$$

(2) $\alpha = 10$ の時

$$
\begin{aligned}
X =&(2^{29}+\cdots+2^{23})+(2^{14}+\cdots+2^{7})\\
=&(2^{29}+\cdots+2^{23})+(2^{14}+\cdots+2^{8})+2^{7}\\
\leftrightarrow&\begin{cases}K_2=\text{0x00},\Delta K_2=\text{0x7F}\\ K_{23}=\text{0x00},\Delta K_{23}=\text{0x7F}\\ K_{18}[7]=0,\Delta K_{18}=\text{0x80}\end{cases}\\
2^{\alpha} =&2^{10}\\
\leftrightarrow&K_6[2]=0,\Delta K_6=\text{0x04}
\end{aligned}
$$

(3) $\alpha = 11$ の時

$$
\begin{aligned}
X =&(2^{30}+\cdots+2^{24})+(2^{15}+\cdots+2^{8})\\
\leftrightarrow&\begin{cases}K_2=\text{0x00},\Delta K_2=\text{0xFE}\\ K_{23}=\text{0x00},\Delta K_{23}=\text{0xFF}\end{cases}\\
2^{\alpha} =&2^{11}\\
\leftrightarrow&K_6[3]=0,\Delta K_6=\text{0x08}
\end{aligned}
$$

5.2 バイアスの探索結果

5.1 で示した差分を用いて，$N = 2^{25}$ でバイアスの探索を行い，$\alpha = 3$ の時に最もバイアスが大きくなり，以下の値を得た．

$$Pr\left(diff\left[23\right] \oplus diff\left[24\right] = 0\right) = \frac{1}{2} - 2^{-4.39}$$

既存研究 [4] と比較すると表 1 のようになる．これは [4] での $2^{-10.46}$ のバイアスよりも大きいバイアスとなる．大きいバイアスが出ることによって，Modular 差分を用いることでより大きなバイアスを見つけることができることが分かった．

バイアスが b の時の識別攻撃に必要な計算量とデータ量は $4 \times (b)^2$ であることが知られている [5]．ゆえに，$2^{10.78}$ の計算量とデータ量で識別攻撃可能である．

6. おわりに

本研究では内部構造の特性を利用して適切な位置に条件にあった Modular 差分を入れることで確率的に差分のキャンセルイベントを発生させ，差分の伝搬を制限した．その結果，28 ラウンドの ZUC-256 に対して有効な Modular 差分を見つけることができ，$N = 2^{25}$ の探索範囲で $2^{-4.39}$ という既存の研究の XOR 差分で求めたバイアスである $2^{-10.46}$ より大きなバイアスを発見した．また，このバイアスを利用し，$2^{10.78}$ の計算量とデータ量で識別攻撃が可能である．

謝辞

本研究は JST さきがけ（JPMJPR2031）と科研費基盤（B19H02141）の助成を受けたものである．

参考文献

[1] ETSI/SAGE. Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. document 2: ZUC specification (2011).
[2] ETSI/SAGE. Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 & 128-EIA3. document 4: Design and Evaluation Reprot. (2011).
[3] The ZUC design team. The ZUC-256 Stream Cipher(2018).
[4] Steve Babbage and Alexander Maximov. Differential analysis of the ZUC-256 initialisation *IACR Cryptol. ePrint Arch.*, 2020:1215(2020).
[5] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher, Advances in Cryptology - EUROCRYPT '93, Lecture Notes in Computer Science 765, pp.386-pp.397(1993).