Computer Security Symposium 2020
26 - 29 October 2020

ストリーム暗号Salsa20におけるProbabilistic Neutral Bitsの解析

宮下　翔太郎[1,a)]　伊藤　竜馬[2]　宮地　充子[1,3]

概要：ストリーム暗号 Salsa20 は 2005 年に Bernstein によって設計された．Salsa20 の有力な攻撃として，Aumasson らの差分解読法があるが，その鍵回復に利用される概念が Probabilistic Neutral Bits（PNB）である．PNB とは，出力の復号時に影響を及ぼさないキービットである．鍵長が l ビット，PNB が n ビットの時，鍵回復時の全数探索の鍵空間が約 2^{l-n} 個に削減できる．本稿では，Salsa の Quarter round の構造を，ビット位置と ARX 構造が及ぼす影響の関係に着目して詳細に分析し，PNB の出現する確率が高くなる条件を調査する実験を行った．結果として，Quarter round の加法による攪拌の影響が少ないビット位置において，PNB が高い確率で出現することを発見した．また，我々の発見した PNB は，既存研究で攻撃に用いられている PNB と比べ，出現確率が高いことがわかった．

キーワード：ストリーム暗号, Salsa20, 鍵回復攻撃, PNB

Analysis on Probabilistic Neutral Bits in Stream Cipher Salsa20

Shotaro Miyashita[1,a)]　Ryoma Ito[2]　Atsuko Miyaji[1,3]

Abstract: The stream cipher Salsa20 was designed by Bernstein in 2005. A differential cryptanalysis on Salsa20 proposed by Aumasson et al. is based on a new technique called Probabilistic Neutral Bits (PNB), which are key bits not changing outputs. PNB don't affect the decryption effectively. When the key length is l-bit and PNB is n-bit, the exhaustive search size can be reduced to approximately 2^{l-n}. In this paper, we analyze the Salsa20 structure, focusing on the relationship between the output bit position and the structure, and experiment to investigate the conditions in which PNB are likely to occur. As a result, we demonstrate that PNB appears with high probability at positions where the effect of addition is weak in the quarter round. In addition, we clarify that the discovered PNB is more accurate than the PNB used in the existing attacks.

Keywords: Stream cipher, Salsa20, Key recovery attack, PNB

1. はじめに

ストリーム暗号は，高速処理に優れた暗号であり，昨今における通信の高速化や大容量化に伴い，注目が集まっている．そのストリーム暗号の一種である Salsa20 [2] は，2005 年に Bernstein によって設計され，8 ラウンド及び 12 ラウンドの軽量化された変種である Salsa20/8 及び Salsa20/12 が，同時に発表されている．2008 年の eSTREAM プロジェクトでは，3 つの評価フェーズの後，Salsa20/12 が Final portfolio に選ばれた．Salsa20 は発表以降，2006 年の Crowle[4] の差分解読法による Salsa20/5 に対する攻撃を皮切りに，Fischer ら [6] による鍵回復攻撃など，様々な攻撃が行われてきた．しかし，フルラウンドである Salsa20 に対して，脅威となる解読法は，未だに発見されていない．

[1] 大阪大学工学研究科
Graduate School of Engineering, Osaka University
[2] 情報通信研究機構
National Institute of Information and Communications Technology
[3] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
[a)] miyashita@cy2sec.comm.eng.osaka-u.ac.jp

Aumassn ら [1] や Maitra ら [7] が提案した攻撃では，出力差分に特徴を有する入出力差分のペアを決定する．次に，その決定した出力差分（順方向の出力差分）に焦点を当て，Probabilistic Backwards Computation と呼ばれる，逆計算により特徴を探す手法を利用し，Salsa20 の逆関数である Backward round 関数に，キーストリームと初期内部状態を算術減算した値を入力して得られた出力差分（逆方向の出力差分）と比較する．この際，初期内部状態に含まれるキービットの値を反転させても，順方向と逆方向の出力差分が一致する確率について評価することで，確率が高くなるキービットの集合と，そうではないキービットの集合に分別することができる．この鍵空間を 2 つの集合に分別する手法が，Probabilistic Neutral Bits（PNB）と呼ばれる概念であり，全数探索する鍵空間を減らすという目的で用いられている．PNB の集合に属する要素が増えると秘密鍵の全数探索を実行する範囲を減らせるため，鍵回復攻撃にかかる全体の計算量を減らすことができる．それゆえに，PNB について詳細に分析することは重要である．しかしながら，これまで，Salsa20 における差分攻撃においては，高いバイアスを示す入力と出力差分の探索に焦点が当てられていた [4] [6] [1] [7]．しかしながら，差分攻撃の鍵回復における計算量では，PNB に相当する鍵が全数探索の鍵計算量に大きな影響を与えるので，この全数探索の計算量を下げることが攻撃全体の効率に大きな影響を与えるだろう．

本研究では，Salsa20 に対する鍵回復攻撃で用いられる PNB に関して焦点を当て，PNB の出現確率を示すパラメータと出力差分位置の関係性を調査し，Backward round 関数の構造と合わせて分析した．実験の結果，Salsa20 の内部構造の構成体であるワード単位で PNB の出現確率が大きく変化することがわかった．また，4 ワード単位で処理される Backward round 関数への入力ワード位置が同じであれば，PNB の出現確率はほとんど同じ値になることが分かった．さらに，各ワードの中で，特に出現確率が大きくなるビット位置があることを観測できた．これらの実験結果を踏まえ，その細部を考察した結果，次のようなことが明らかになった．

- PNB の出現確率は，Backward round 関数の構造の影響を強く受け，キービット位置の影響でなく出力差分位置に影響を受ける．
- 既存研究で用いられている出力差分位置より，PNB が出現しやすい出力差分位置が存在する．したがって，鍵回復攻撃にかかる全体の計算量を削減させる入出力差分ペアが既存研究で用いられている入出力差分ペアの他に存在する可能性がある．
- PNB を用いて鍵回復攻撃を行う際に，Backward round 関数を用いることができるラウンド数に上界が存在する．

本稿では，まず，第 2 章で Salsa20 の構造を，第 3 章で，関連研究で用いられている差分攻撃及び鍵回復攻撃の手法を説明する．次に，第 4 章で本研究で実施した PNB に対する解析の内容とその実験結果を示し，第 5 章で PNB と Backward round 関数の構造との関係性，そして PNB を鍵回復攻撃に利用可能なラウンド数の上界について考察する．最後に，第 6 章で本稿を締めくくる．

2. Salsa20

Salsa20 [2] は，2005 年に Bernstein によって設計されたストリーム暗号であり，32 ビット算術加算（Addition），ビットローテーション（Rotation），排他的論理和（XOR）で構成された ARX 構造により，擬似乱数系列（キーストリーム）を生成する方式である．

Salsa20 の内部状態は，32 ビットを 1 ワードとして 16 ワードで構成され，8 ワードの秘密鍵 $k = (k_0, \ldots, k_7)$，2 ワードの Nonce $v = (v_0, v_1)$，2 ワードのブロックカウンタ $t = (t_0, t_1)$，4 ワードの定数 $\sigma = (\sigma_0, \ldots, \sigma_3)$ を入力して内部状態を初期化する．通常，秘密鍵は 128 ビット又は 256 ビットから選択することができ，鍵長として 128 ビットを選択した場合には，8 ワードの秘密鍵を $k = (k_0, \ldots, k_7) = (k_0, \ldots, k_3, k_0, \ldots, k_3)$ とする．本稿では，鍵長を 256 ビットとして扱う．また，Salsa20 の内部状態 X は，4×4 の行列で表記され，初期内部状態は次のとおりとなる．

$$X = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} = \begin{pmatrix} \sigma_0 & k_0 & k_1 & k_2 \\ k_3 & \sigma_1 & v_0 & v_1 \\ t_0 & t_1 & \sigma_2 & k_4 \\ k_5 & k_6 & k_7 & \sigma_3 \end{pmatrix}.$$

Salsa20 のラウンド関数は，4 個の Quarter round 関数で構成されている．Quarter round 関数は 4 ワード入出力関数であり，Addition，Rotation，XOR によって内部状態を攪拌する．Quarter round 関数の入力を (x_A, x_B, x_C, x_D)，出力を (y_A, y_B, y_C, y_D) とすると，Quarter round 関数 QR は次のような手順で計算される．

$$\begin{cases} y_B = x_B \oplus ((x_A + x_C) \lll 7) \\ y_C = x_C \oplus ((y_B + x_A) \lll 9) \\ y_D = x_D \oplus ((y_C + y_B) \lll 13) \\ y_A = x_A \oplus ((y_C + y_D) \lll 18) \end{cases}$$

ここで，$+$，$\lll$，$\oplus$ はそれぞれ Addition，左 Rotation，XOR を表している．また，**図 1** は Quarter round 関数の概略図を示している．奇数ラウンド（Odd round）において，Quarter round 関数は内部状態の列ベクトル (x_0, x_4, x_8, x_{12})，(x_5, x_9, x_{13}, x_1)，$(x_{10}, x_{14}, x_2, x_6)$，$(x_{15}, x_3, x_7, x_{11})$ に適用される．一方，偶数ラウンド（Even round）において，Quarter round 関数は内部状態の行ベクトル (x_0, x_1, x_2, x_3)，

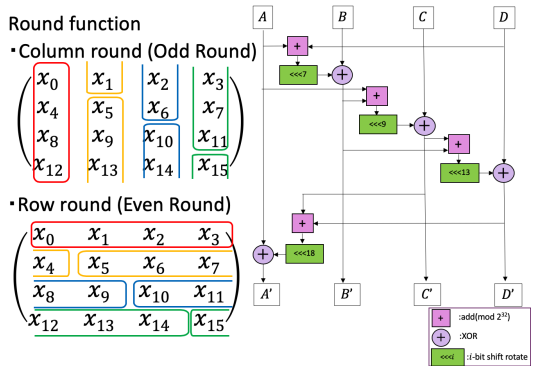

図 1　Quarter round の構造

(x_5, x_6, x_7, x_4), $(x_{10}, x_{11}, x_8, x_9)$, $(x_{15}, x_{12}, x_{13}, x_{14})$ に適用される．

r ラウンド後の内部状態を X^r，Salsa20 のラウンド数を R（Bernstein によって提案されたオリジナルのラウンド数は $R = 20$），R ラウンドバージョンの Salsa20 を Salsa20/R で表す．すると，Salsa20/R におけるキーストリーム Z は，

$$Z = X + X^R$$

として計算される．

3. Salsa20 に対する差分攻撃手法

Salsa20 における差分攻撃では，Crowley[4] や，Fischer ら [6] によって，高いバイアスを持つ入出力差分が計算され，Aumasson ら [1] や，Deepthi ら [5] においては，差分攻撃の鍵回復アルゴリズムを明確に記載することで，解読に掛かる計算量をより厳密に定義した．

3.1 Salsa20 のバイアス

$x_i^r[j]$ を，内部状態 X^r の i 番目のワードの j 番目のビットとする．また，$x'_i[j]$ を，$\Delta_i[j] = x_i[j] \oplus x'_i[j]$ となるビットであると表現する．差分攻撃を行うに際し，初期状態 X に対して，1 ビットの入力差分（$\mathcal{ID}$）$\Delta_i[j] = 1$ を与え，X' を得る．また，r ラウンド後の内部状態 X^r の 1 ビットの出力差分を $\Delta_p^r[q]$ と表現し，出力差分（$\mathcal{OD}$）とする．これらを，$(\Delta_p^r[q] = 1 \mid \Delta_i[j] = 1)$ とする．

ある固定された鍵に対して，バイアス ϵ_d は，次の式で定義される．

$$\Pr(\Delta_p^r[q] = 1 \mid \Delta_i[j] = 1) = \frac{1}{2}(1 + \epsilon_d)$$

一様ランダムに選択した秘密鍵に対して，バイアス ϵ_d の中央値 ϵ_d^* が高くなる $\mathcal{ID} - \mathcal{OD}$ ペアを探す．Aumasson ら [1] や，Deepthi ら [5] においては，$(\Delta_1^4[14] = 1 \mid \Delta_7[31] = 1)$ が用いられており，この時 $|\epsilon_d^*| = 0.131$ である．

3.2 鍵回復攻撃

鍵回復攻撃は，事前準備フェーズと，オンラインフェーズに分かれる．事前準備フェーズは，第 3.2.1 節の Neutral measure を求め，キービットを PNB と non-PNB に分ける部分を指す．オンラインフェーズは，キービットのうち，non-PNB の部分から尤もらしい鍵候補を探し，その後，PNB の部分を全数探索し，鍵を回復していく部分を指す．

3.2.1 Probabilistic neutral bits（PNB）

本節では，Probabilistic Neutral Bits (PNB) と呼ばれる neutral bits の一般的な概念について紹介する．PNB とは，キービットを 2 つのグループに分割するものであり，重要なキービット（m-bit）のグループと，重要でないキービット（n-bit）である．キービットのそれぞれが，出力差分（$\mathcal{OD}$）に与える影響の大きさに焦点を当てており，，その指標として Neutral measure が次のように定義されている．

定義 1 ([1])．出力差分（$\mathcal{OD}$）に関するキービット k_κ の *Neutral measure* は，γ_κ として定義される．この時，$\Pr = \frac{1}{2}(1 + \gamma_\kappa)$ は，キービット k_κ を反転しても出力差分が変化しない確率である．

定義 1 で示される Neutral measure の求め方は，次のとおりである

Step 1. 差分攻撃で利用する $\mathcal{ID}$ ペア X, X' から R ラウンド後の内部状態 X^R, X'^R を計算し，キーストリーム $Z = X + X^R$, $Z' = X' + X'^R$ を生成する．

Step 2. 初期状態 X 及び X' の，キービット位置 κ を反転させた内部状態を，$\overline{X}$ 及び $\overline{X'}$ とする．

Step 3. $Z - \overline{X}$ 及び $Z' - \overline{X'}$ を Backward round 関数への入力とし，$R - r$ ラウンド後の内部状態をそれぞれ Y^r 及び Y'^r とする．

Step 4. 差分攻撃の時に着目していた $\mathcal{OD}$ 位置である p 番目のワードの q 番目のビット差分をとり，$y_p^r[q] \oplus y_p'^r[q] = \Gamma_p^r[q]$ とする．

Step 5. $\Delta_p^r[q]$ と，$\Gamma_p^r[q]$ とが一致する確率のバイアスを Neutral measure とする．

Neutral measure は，以下のように表すことができる．

$$\Pr(\Delta_p^r[q] = \Gamma_p^r[q] \mid \Delta_i[j]) = \frac{1}{2}(1 + \gamma_\kappa)$$

また，Neutral measure が次の値を示す時は，以下のことが言える．

- $\gamma_\kappa = 1 : \kappa$ 番目のキービットは，出力差分（$\mathcal{OD}$）に影響しない．（Neutral bit である）
- $\gamma_\kappa = 0 : \kappa$ 番目のキービットは，出力差分（$\mathcal{OD}$）に影響を及ぼす．（non-PNB である）

キービットを，PNB と non-PNB の 2 つの集合に分別するために，閾値 γ を用意する．

κ 番目のキービットの Neutral measure が $\gamma_\kappa \geq \gamma$ の時，κ 番目のキービットは PNB の集合に属しているものとみなす．反対に，κ 番目のキービットの Neutral measure が $\gamma_\kappa < \gamma$ の時，κ 番目のキービットは non-PNB の集合に属

しているものとみなす．

3.2.2 Probabilistic Backwards Computation

差分攻撃では入力差分ペアを選択して r ラウンドにおける出力差分のバイアスを計算する．つまり，暗号方式における順方向のバイアスを計算していることになる．一方，Salsa20/R では選択した入力差分ペアと対応するキーストリームペアから逆計算により，r ラウンドにおける出力差分のバイアスを計算することも可能である．

逆計算によりバイアスを計算する手法は，Probabilistic Backwards Computation と呼ばれている．

Probabilistic Backwards Computation の過程は，次のとおりである．

Step 1. 差分攻撃で利用する $\mathcal{ID}$ ペア X, X' から R ラウンド後の内部状態 X^R, X'^R を計算し，キーストリーム $Z = X + X^R$, $Z' = X' + X'^R$ を生成する

Step 2. 初期状態 X, X' において，m-bit の non-PNB の値に対して，正しいと思われる値を入れ，n-bit の PNB に対して，固定値（ランダムな値や All-ZERO など）を割り当てるものを $\hat{X}, \hat{X}'$ とする．

Step 3. $Z - \hat{X}, Z' - \hat{X}'$ を計算して Backward round 関数への入力とし，$R - r$ ラウンド後の内部状態 $\hat{Y}^r, \hat{Y'}^r$ を得る．

Step 4. $\hat{Y}^r, \hat{Y'}^r$ を XOR して，$\hat{\Gamma}^r_p[q] = \hat{Y}^r \oplus \hat{Y'}^r$ を得る

Step 5. 確率 $\Pr(\hat{\Gamma}^r_p[q] = \Delta^r_p[q] \mid \Delta_i[j]) = \frac{1}{2}(1 + \epsilon_a)$ を求め，逆計算のバイアスを ϵ_a とする．

R ラウンド後のバイアスは，$\Pr(\hat{\Gamma}^r_p[q] = 1 \mid \Delta_i[j] = 1) = \frac{1}{2}(1 + \epsilon)$ である．これは，Probabilistic Backwards Computation の Step 5. で求めるものを，$\hat{\Gamma}^r_p[q]$ のバイアスとすればよい．$\epsilon \approx \epsilon_a \cdot \epsilon_d$ の時に，鍵が正しいとされる．ϵ^* を，全ての鍵候補の ϵ の中央値とする．

3.2.3 事前準備フェーズ

第 3.2.1 節と第 3.2.2 節で説明した手法に基づき，事前準備フェーズは次のような過程で実行される．

Step 1. ϵ^*_d が高くなる $\mathcal{ID} - \mathcal{OD}$ ペアを探す．

Step 2. 各キービットの Neutral measure である，γ_κ を測る．

Step 3. 閾値 γ を用いて，キービットを PNB と non-PNB の集合に分別する．

3.2.4 オンラインフェーズ

オンラインフェーズは次のような過程で実行される．

Step 1. 入力差分（$\mathcal{ID}$）$\Delta_i[j]$ を満たす初期内部状態ペアを N 組選択する（Nonce とブロックカウンタを一様ランダムに選択する）．

Step 2. non-PNB に属する m 個のキービット（サブキー）に対して，次の過程を実行する．

(1) N 個のキーストリームペアを用いて，ϵ_a を求める．

(2) (1) で，最も高いバイアス ϵ^*_a を尤もらしいサブキーの候補であると識別し，残りの n-bit である PNB の全数探索をする．

(3) 全ての鍵が正しいと識別された場合は，鍵回復攻撃を停止し，回復した鍵を出力する．

3.2.5 鍵回復攻撃における計算量

第 3.2.4 節で示した攻撃は，まず，m-bit の non-PNB に対して，それぞれに対し，サンプル数 N で鍵を推測する．

この時，以下の 2 種類の誤りが想定される．

(1) 第 1 種の誤り：識別したサブキーは正しい鍵ではないが，正しいと判断される．（確率 p_{nd}）

(2) 第 2 種の誤り：識別したサブキーは正しい鍵だが，正しくないと判断される．（確率 p_{fa}）

ネイマン・ピアソンの決定定理により，$p_{nd} = 1.3 \times 10^{-3}$ および，$p_{fa} = 2^{-\alpha}$ として，以下に検証に必要なサンプル数 N を決定することができる．

$$N \approx \left(\frac{\sqrt{\alpha \log 4} + 3\sqrt{1 - \epsilon^2}}{\epsilon} \right)^2$$

最終的に，鍵回復攻撃にかかる計算量は，

$$2^m(N + 2^n p_{fa}) = 2^m N + 2^{256-\alpha}$$

となる．計算量の導出について，細部は Aumasson ら [1] らの論文を参照されたい．

Aumasson ら [1] は，Salsa20/8 に対して，入出力差分 $(\Delta^r_1[14]|\Delta_7[31])$，閾値 $\gamma = 0.12$ を設定することで，$n = 36, |\epsilon_a *| = 0.0011, |\epsilon *| = 0.00015$ を得ることができ，鍵回復攻撃にかかる計算量を 2^{251} と見積もった．

4. PNB の解析

4.1 モチベーション

第 3 章で述べたように，PNB は差分攻撃と組み合わせて Salsa20 の秘密鍵を復元する重要な概念であり，PNB の出現確率を表す Neutral measure の大きさが攻撃可能ラウンド数と攻撃にかかる計算量に影響を及ぼすことがよく知られている．既存攻撃 [1] [9] [7] [3] では，最初に出力差分の高いバイアス ϵ_d を有する $\mathcal{ID}$-$\mathcal{OD}$ ペアを決定した後，その出力差分位置に焦点を当てて Neutral measure γ_κ を評価するという流れで鍵回復攻撃が実施されてきた．しかしながら，この流れで鍵回復攻撃を実施した場合，バイアス ϵ_d と Neutral measure γ_κ の組み合わせが真に最適であるかを示すことができない．

そこで，本稿では Salsa20 における PNB に着目し，Neutral measure γ_κ が高くなる条件を調査する．これまで Salsa20 の PNB について詳細に解析した研究成果は報告されておらず，Neutral measure γ_κ が高くなる条件を明らかにすることができれば，Salsa20 に対する鍵回復攻撃をより厳密に評価できるようになる．

4.2 Neutral measure 測定アルゴリズム

Neutral measure が高くなる条件を調査するため，全ての $\mathcal{ID}$-$\mathcal{OD}$ ペアから Neutral measure を測定する実験を行う．実験方法は次のとおりであり，Neutral measure 測定アルゴリズムは Algorithm 1 のとおりである．

Step 1. 秘密鍵 $k = (k_0, \ldots, k_7)$ を一様ランダムに生成する．

Step 2. 入力差分位置 i, j を決め，入力差分として選択可能な Nonce $v = (v_0, v_1)$ とブロックカウンタ $t = (t_0, t_1)$ を一様ランダムに生成し，初期内部状態 X とその差分ペアである $X' = X \oplus \Delta_i^0[j]$ を生成する．

Step 3. 初期内部状態 X, X' から差分攻撃のターゲットとなる r ラウンド目の内部状態 X^r, X'^r と鍵回復攻撃のターゲットとなる R ラウンド目の内部状態 X^R, X'^R を計算し，キーストリームペア $Z = X + X^R$, $Z' = X' + X'^R$ を生成する．また，r ラウンド目の出力差分 $\Delta^r = X^r \oplus X'^r$ を計算する．

Step 4. 初期内部状態 X, X' のキービット位置 $\kappa \in \{0, \ldots, 255\}$ を 1 ビットのみ反転させた初期内部状態 $\overline{X}$, $\overline{X'}$ を用いて，$(R - r)$ ラウンドの Backward quarter round 関数にて r ラウンド目の内部状態 $Y^r = Salsa^{r-R}(Z - \overline{X})$, $Y'^r = Salsa^{r-R}(Z' - \overline{X'})$ を計算する．また，r ラウンド目の出力差分 $\Gamma^r = Y^r \oplus Y'^r$ を計算する．

Step 5. r ラウンド目の出力差分 Δ^r, Γ^r から，p 番目のワードの q 番目のビット位置において，$\Delta_p^r[q] = \Gamma_p^r[q]$ となる場合にのみカウンタをインクリメントする．この際，出力差分位置とキービット位置ごとに分割してカウンタを準備する．

Step 6. 全ての試行が終了した後，カウンタ値から Neutrak measure γ_κ を計算する．

Algorithm 1 Neutral measure の測定

Input: m : 鍵の数, n : 鍵 1 個当たり各 $\mathcal{ID}$ 毎のサンプル数
Output: 各 $\mathcal{OD}$ $\Delta_p^r[q]$ 毎の neutral measure γ_κ を格納した 3 次元配列 $\gamma[\kappa][p][q]$

for $a = 0$ up to m **do**
  $X \xleftarrow{random} (k_0, \ldots, k_7)$
  for all i, j s.t. $6 \le i \le 9, 0 \le j \le 31$ **do**
    for $b = 0$ up to n **do**
      $X \xleftarrow{random}$ Nonce (v_0, v_1), Counter (t_1, t_2)
      $X' \leftarrow X \oplus \Delta_i^0[j]$
      $X^r \leftarrow Salsa^r(X)$, $X'^r \leftarrow Salsa^r(X')$
      $\Delta^r = X^r \oplus X'^r$
      $X^R \leftarrow Salsa^R(X)$, $X'^R \leftarrow Salsa^R(X')$
      $Z \leftarrow X^R + X$, $Z' \leftarrow X'^R + X'$
      for $\kappa = 0$ up to 255 **do**
        $\hat{X}, \hat{X}' \leftarrow X, X'$ のキービット位置 κ を反転
        $Y^r \leftarrow Salsa^{r-R}(Z - \hat{X})$, $Y'^r \leftarrow Salsa^{r-R}(Z' - \hat{X}')$
        $\Gamma^r \leftarrow Y^r \oplus Y'^r$
        for all p, q s.t. $0 \le p \le 15, 0 \le q \le 31$ **do**
          if $\Delta_p^r[q] = \Gamma_p^r[q]$ **then**
            $sumNM[\kappa][p][q] \leftarrow sumNM[\kappa][p][q] + 1$
          end if
        end for
      end for
    end for
  end for
end for
for all κ, p, q s.t. $0 \le \kappa \le 255, 0 \le p \le 15, 0 \le q \le 31$ **do**
  $NM[\kappa][p][q] \leftarrow sumNM[\kappa][p][q]/(m \cdot n \cdot 4 \cdot 32)$
  $\gamma[\kappa][p][q] \leftarrow 2 \cdot NM[\kappa][p][q] - 1$
end for

4.3 実験結果

本節では，第 4.2 節で示した実験方法により Neutral measure を測定した結果を示す．本実験において，我々は一様ランダムに選択した 2^6 個の秘密鍵を用いて，秘密鍵 1 個当たり 2^{25} 組の入力差分ペア（サンプル数）から Neutral measure を測定した．信頼できるサンプル数であるかを確認するため，Mantin と Shamir によって示された次の定理を使用する [8].

定理 1 ([8], Theorem 2)．ある事象 e が確率 p で起こる分布を $\mathcal{X}$，確率 $p \cdot (1 + q)$ で起こる分布を $\mathcal{Y}$ とする．この時，$\mathcal{X}$ と $\mathcal{Y}$ の分布を一定の成功確率で識別するためには，$\mathcal{O}(\frac{1}{pq^2})$ のサンプル数が必要である．

ここで事象 e は，r ラウンド目の出力差分 Δ^r, Γ^r における i 番目のワードの j 番目のビット位置で $\Delta_i^r[j] = \Gamma_i^r[j]$ を満たす事象を指す．また，一様ランダムな分布を $\mathcal{X}$，Salsa20/R における事象 e の分布を $\mathcal{Y}$ とすると，$p = \frac{1}{2}$，$q = \gamma_\kappa$ となるため，精度の高い Neutral measure γ_κ を得るためには，$\mathcal{O}(\frac{2}{\gamma_\kappa^2})$ 個のサンプル数が必要である．本実験では，サンプル数が 2^{25} 個であるため，測定した Neutral measure が $\gamma_\kappa \ge 2^{-12}$ である事象については，信頼できるサンプル数であると言える．

図 2 は，Neutral measure 測定実験の結果をグラフにまとめたものである．図の縦軸は，各出力差分位置における Neutral measure の平均値，横軸は，出力差分のワード位置を表している．なお，縦軸の補助線はワード位置で区切っている．赤，青，緑，黄色の線は，Backward round 関数での処理ラウンド数 $(R - r)$ がそれぞれ 2, 3, 4, 5 ラウンドの時の Neutral measure の平均値を示しており，差分攻撃のターゲットラウンドは比較を容易にするため $r = 4$ で固定した．

グラフ下部の記号は，ワード位置が Quarter round や Backward round においてどの位置に入るかを示している．丸で表されるものが，Odd round の時の状態を示し，四角で表されるものが，奇数ラウンドの時の状態を示す．同じ色であれば，同じ組み合わせでラウンド処理され，各アルファベットは，その際にどの位置に入るのかを示す．図 2 からみられるとおり，ワードごとで Neutral measure の値が大きく変化することがわかる．また，Quarter round に

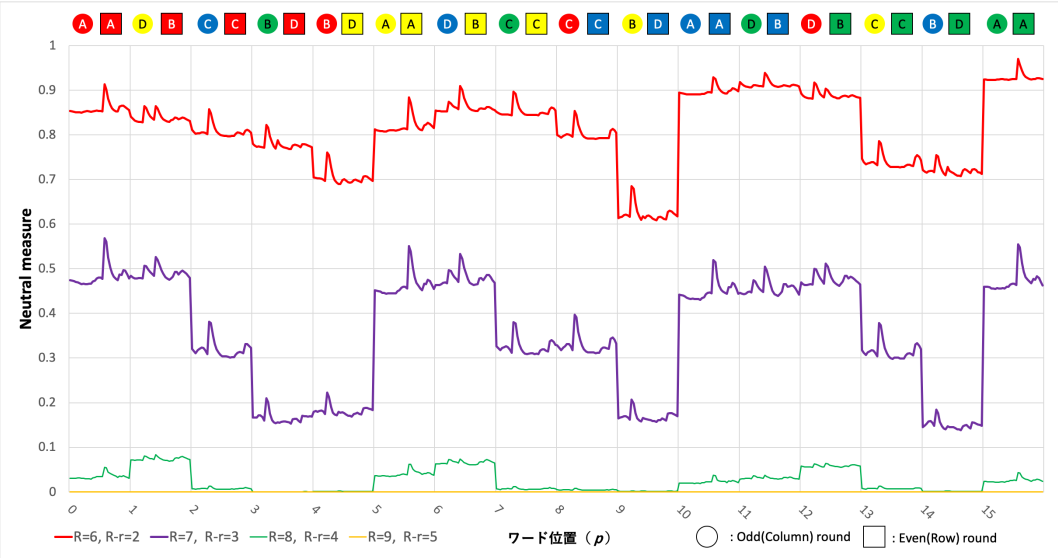

図 2　ラウンド数毎の Neutral measure の推移

入る位置が一緒であれば，同じような値になることが分かる．また，各ワードの中に，特に Neutral measure が大きくなるビット位置があることが観測できる．

5. PNB に関する考察

第 4 章で示した実験結果から，次の 2 点について考察する．1 点目は，PNB の出現確率と Backward round 関数の関係性について，Backward round 関数への入出力ワード位置と Addition の累積実行回数の関係性から考察する．2 点目は，攻撃に利用可能な Backward round 数の上界について，Backward round 数の増加に伴う Neutral measure の推移から考察する．

5.1 PNB の出現確率と Backward round 関数の関係性

PNB の出現確率と Backward round 関数の関係性について考察するために，Backward round 関数への入出力ワード位置と Addition の累積実行回数の関係性を分析する．表 1 は分析した結果をまとめたものである．R の列は，鍵回復攻撃のターゲットラウンド数が奇数（Odd）か偶数（Even）であるかを示している．入力ワード位置の列は，図 1 の入力ワード位置と対応しており，Backward round 関数において入力ワード位置が奇数ラウンド（Odd round）と偶数ラウンド（Even round）でどのように遷移するかを表している．例えば，R が Odd，入力ワード位置の Odd round が B，Even round が D の行については，Backward round 数が 2 の時，Odd round (B) → Even round (D) でワード位置が遷移していくことを意味する．また，Backward round 数が 3 の時，Odd round (B) → Even round (D) → Odd round (B) でワード位置が遷移する．Addition の累積実行回数の列は，Backward round 数が 2, 3 の時に，各入力ワード位置が Backward round 関数によって実行される Addition の累積回数を表している．

表 1　Backward round 関数への入力位置と Addition の累積回数

R	入力ワード位置		Addition の累計回数	
	Odd	Even	2 ラウンド	3 ラウンド
Odd	A	A	3	9
	B	D	4	21
	C	C	5	16
	D	B	7	12
Even	A	A	3	9
	B	D	7	12
	C	C	5	19
	D	B	4	21

表 1 から，Backward round 関数への入出力ワード位置によって，Addition の累積実行回数に差が生じていることがわかる．例えば，R が Odd，Backward round が 2 の時，入力ワード位置が Odd round (A) → Even round (A) において 3 回のみであるのに対し，入力ワード位置が Odd round (D) → Even round (B) においては 7 回も Addition が実行されている．また，R が Odd，Backward round が 3 の時には，入力ワード位置が Odd round (A) → Even round (A) → Odd round (A) において 9 回のみであるのに対し，入力ワード位置が Odd round (B) → Even round (D) → Odd round (B) においては 21 回も Addition が実行されていることを意味する．

ここで，図 2 の実験結果と表 1 の分析結果を比較する．図 2 のグラフをワード単位で区切り，入力ワード位置のパターン（グラフ上部の記号を参照）とその Neutral measure に注目すると，表 1 で示した Addition の累積実行回数が多いほど Neutral measure が低くなることがわかる．

例えば，$R = 7$，Backward round が 3 ラウンドの時，Backward round 関数は Odd round → Even round → Odd round の順に処理が行われる．ワード単位で分析すると，

表 2　各ラウンド数における Neutral measure の最大値と最小値

R	最大値			最小値		
	NM	p	q	NM	p	q
6	$2^{-0.0445}$	15	18	$2^{-0.715}$	9	20
7	$2^{-0.814}$	0	18	$2^{-2.855}$	14	20
8	$2^{-3.585}$	1	13	$2^{-12.072}$	3	24
9	$2^{-13.941}$	0	18	$-2^{-14.515}$	4	15

Neutral measure が最も低い値となった入力ワード位置は $p = 3, 4, 9, 14$ の時であり，これらの入力ワード位置は全て Odd round で B のワード位置，Even round で D のワード位置であることがわかる．また，表 1 から，これらの入力ワード位置のパターンが Addition の累積実行回数が最も多いことがわかる．同様に，入力ワード位置のパターンが同じであれば，Neutral measure もほとんど同じ値を示すことがわかり，これは Addition の累積実行回数の数に依存している．

まとめると，PNB の出現確率（Neutral measure）は，Backward round 関数への入力ワード位置によって差が生じるとともに，その差は Addition の累積実行回数に依存していると言える．

5.2　攻撃に利用可能な Backward round 数の上界

鍵回復攻撃に利用可能な Backward round 数の上界を考察するために，各 Backward round 数における Neutral measure の値について分析する．図 2 から，Backward round 数の増加に伴って，Neutral measure の値が全体的に小さくなっていることがわかる．また，差分攻撃のターゲットラウンド数が $r = 4$ の場合における Neutral measure の最大値と最小値，そしてそれらの出力差分位置について詳細に分析し，表 2 に分析結果をまとめた．表の R の列は，鍵回復攻撃のターゲットラウンド数を表しており，Backward round 数は $R - r$ で計算できる．NM，p，q は，それぞれ Neutral measure，出力差分のワード位置，出力差分のビット位置を表している．

表 2 から，ある Backward round 数における Neutral measure の最大値は，それより少ない Backward round 数における Neutral measure の最小値を上回ることがないことがわかる．例えば，$R = 7$ の時，つまり Backward round 数が $R - r = 3$ における Neutral measure の最大値が $2^{-0.814}$ であるのに対し，$R = 6$ の時，つまり Backward round 数が $R - r = 2$ における Neutral measure の最小値が $2^{-0.716}$ であるため，後者の方が高いことが明らかである．これは第 5.1 節でも考察したように，Addition の累積実行回数が影響していると考えられる．

第 4 章において，本実験で使用したサンプル数（2^{25} 組の入力差分ペア）から測定した Neutral measure が $\gamma_\kappa \geq 2^{-12}$ である事象については，信頼できるサンプル数であると述べた．表 2 から，$R = 6, 7$ の場合における Neutral measure が全て信頼できる値であり，$R = 8$ の場合における Neutral measure が一部を除き信頼できる値であることがわかる（最小値が $2^{-12.072} < 2^{-12}$ であるため）．なお，$R = 9$ の場合における Neutral measure が全て信頼できる値とは言い切れないものの，Neutral measure が $\gamma_\kappa < 2^{-12}$ であることは確実であると言い切れる．第 3 章で示したように，既存研究で実際に利用されている閾値が $\gamma = 0.12 (= 2^{-3.059})$ であることを踏まえると，$R = 9$ の時，つまり Backward round 数が $R - r = 5$ 以上の場合は，鍵回復攻撃を効率的に実行することが困難であると言える．

まとめると，Salsa20 に対する鍵回復攻撃に利用可能な Backward round 数の上界が 4 ラウンドであると言える．

6.　まとめ

本稿では，Salsa20 における PNB に関して，Backward round 関数の構造に着目して詳細に分析し，鍵回復攻撃への適用という観点から，次のように考察した．

まず最初に，PNB の出現確率が高くなる条件について調査するために，全ての $\mathcal{ID} - \mathcal{OD}$ ペアから PNB の出現確率を示す Neutral measure を測定する実験を行った．実験の結果，Neutral measure は出力差分のワード位置ごとに大きな差が生じることが明らかになった．

次に，このような差が生じる原因を明らかにするため，PNB の出現確率と Backward round 関数の関係性について分析した．分析の結果，Backward round 関数への入力差分位置によって，Addition の累積実行回数に差が生じていることが分かった．Addition の累積実行回数が少ないと Neutral measure が高くなるという関係性を導き出し，結果として，Neutral measure に大きな差が生じるのは，Backward round 関数の構造によるものであることを明らかにした．

また，鍵回復攻撃に利用可能な Backward round 数の上界について考察した．Backward round 関数の構造を詳細に解析したところ，Addition の累積実行回数の増加に伴って Neutral measure の値が 0 に収束し，PNB の出現確率が減少することが分かった．解析の結果，Backward round 数の上界が 4 ラウンドであることを示した．

Salsa20 に対する既存の鍵回復攻撃では，高いバイアスを有する入出力差分ペアを利用し，この出力差分に対応する PNB を解析する，という流れで評価されたきた．本研究では，PNB が出現しやすい出力差分位置を明らかにし，鍵回復攻撃にかかる全体の計算量を改善できる可能性があることを示した．

謝辞　本研究の一部は文部科学省「Society5.0 に対応した高度技術人材育成事業成長分野を支える情報技術人材の育成拠点の形成 (enPiT)」さらに文部科学省の平成 30 年度「Society 5.0 実現化研究拠点支援事業」の助成を受けています．

参考文献

[1] Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba. In Kaisa Nyberg, editor, *International Workshop on Fast Software Encryption - FSE 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 470–488. Springer Berlin Heidelberg, 2008.

[2] Daniel J. Bernstein. The Salsa20 Family of Stream Ciphers. In Matthew Robshaw and Olivier Billet, editors, *New Stream Cipher Designs*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer Berlin Heidelberg, 2008.

[3] Arka Rai Choudhuri and Subhamoy Maitra. Significantly Improved Multi-Bit Differentials for Reduced Round Salsa and ChaCha. *IACR Transactions on Symmetric Cryptology*, 2016(2):261–287, 2017.

[4] Paul Crowley. Truncated differential cryptanalysis of five rounds of Salsa20. In *Stream Ciphers Revisited, Workshop Record - SASC2006*, 2006.

[5] Kakumani K. C. Deepthi and Kunwar Singh. Cryptanalysis of Salsa and ChaCha: Revisited. In J. Hu et al., editor, *Mobile Networks and Management - MONAMI 2017*, volume 235 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 324–338. Springer Berlin Heidelberg, 2017.

[6] Simon Fischer, Willi Meier, Côme Berbain, Jean-Fran çois Biasse, and M.J.B. Robshaw. Non-randomness in eSTREAM Candidates Salsa20 and TSC-4. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology - Indocrypt 2006*, volume 4329 of *Lecture Notes in Computer Science*, pages 2–16. Springer Berlin Heidelberg, 2006.

[7] Subhamoy Maitra, Goutam Paul, and Willi Meier. Salsa20 Cryptanalysis: New Moves and Revisiting Old Styles. In *the Ninth International Workshop on Coding and Cryptography - WCC2015*, 2015.

[8] Itsik Mantin and Adi Shamir. Practical Attack on Broadcast RC4. In Mitsuru Matsui, editor, *International Workshop on Fast Software Encryption - FSE 2001*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer Berlin Heidelberg, 2001.

[9] Zhenqing Shi, Bin Zhang, Dengguo Feng, and Wenling Wu. Improved Key Recovery Attacks on Reduced-Round Salsa20 and ChaCha. In T. Kwon, M.-K. Lee, and D. Kwon, editors, *International Conference on Information Security and Cryptology - ICISC2012*, volume 7839 of *Lecture Notes in Computer Science*, pages 337–351. Springer Berlin Heidelberg, 2013.