PNB-focused Differential Cryptanalysis of ChaCha Stream Cipher

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

[1] Osaka University, Suita, Japan
miyashita@cy2sec.comm.eng.osaka-u.ac.jp, miyaji@comm.eng.osaka-u.ac.jp
[2] National Institute of Information and Communications Technology, Koganei, Japan
itorym@nict.go.jp
[3] Japan Advanced Institute of Science and Technology, Nomi, Japan

Abstract. This study focuses on differential cryptanalysis of the ChaCha stream cipher. In the conventional approach, an adversary first searches for an input/output differential pair with the highest differential bias and then analyzes the *probabilistic neutral bits* (PNB) based on the obtained input/output differential pair. However, although the time and data complexities for the attack can be estimated by the differential bias and PNB obtained by this approach, the combination of the differential bias and PNB is not always optimal. In addition, the existing studies have not performed a comprehensive analysis of the PNB; thus, they have not provided an upper bound on the number of rounds required for a differential attack that uses a single-bit truncated differential to be successful. To address these limitations, we propose a *PNB-focused differential attack* on reduced-round ChaCha by first comprehensively analyzing the PNB for all possible single-bit truncated output differences and then searching for the input/output differential pair with the highest differential bias based on the obtained PNB. The best existing attack on ChaCha, proposed by Beierle et al. at CRYPTO 2020, works on up to 7 rounds, whereas the most extended attack we observed works on up to 7.25 rounds using the proposed PNB-focused approach. The time complexity, data complexity, and success probability of the proposed attack are $2^{255.62}$, $2^{48.36}$, and 0.5, respectively. Although the proposed attack is less efficient than a brute force attack, it is the first dedicated attack on the target and provides both a baseline and useful components (*i.e.*, differential bias and PNB) for improved attacks.

Keywords: Stream Cipher · ChaCha · Differential Cryptanalysis · PNB

1 Introduction

ChaCha [4] is a stream cipher designed by Bernstein in January 2008. It was motivated by the ECRYPT Stream Cipher Project (eSTREAM)[1] finalist, Salsa [5], which was proposed by the same designer in April 2005. After the release of

[1] http://www.ecrypt.eu.org/stream

Salsa and ChaCha, several studies performed the security evaluations of both ciphers [1, 3, 6–16, 18]. One of the most relevant of these evaluations is the differential attack based on the concept of *probabilistic neutral bits* (PNB), proposed by Aumasson et al. at FSE 2008 [1]. The PNB concept is to divide secret key bits into two sets – a set of *significant key bits* and a set of *non-significant key bits* – and to use a *neutral measure* as an evaluation indicator to distinguish them. The fewer the elements in the set of significant key bits, the lower the time complexity required for an adversary to recover the unknown secret key; thus, it is crucial to analyze the PNB concept for the differential attacks on Salsa and ChaCha.

Aumasson et al. [1] first searched for the input/output differential pair with the highest differential bias; then, based on this pair, they divided the secret key bits into two sets using the PNB concept; finally, they performed a differential attack on the 7-round version of ChaCha, ChaCha20/7, with time and data complexities of 2^{248} and 2^{27}, respectively. Several researchers later reported improvements to this attack [3, 6–9, 16, 18]. To the best of our knowledge, the best key recovery attack on ChaCha works on up to seven rounds with time and data complexities of $2^{230.86}$ and $2^{48.80}$, respectively, proposed by Beierle et al. at CRYPTO 2020 [3].

The existing studies [1, 3, 6–9, 16, 18] have focused on searching for the input/output differential pair with the highest differential bias; however, no study focusing on PNB analysis has been conducted thus far. For this reason, the combination of differential biases and PNB obtained from the existing attacks may not always be optimal. The theoretical time and data complexities for the attacks can be estimated from the combination of differential biases and PNB. In addition, the differential biases and PNB can be analyzed independently; therefore, focusing on the PNB analysis may help provide an upper bound on the number of rounds required for a differential attack that uses a single-bit truncated differential to be successful. The above suggests that PNB-focused analysis has the potential to improve the existing attacks.

Our Contributions. In this study, we propose a *PNB-focused differential attack*. The proposed attack targets reduced-round ChaCha by first analyzing the PNB for all possible single-bit truncated output differences ($\mathcal{OD}$s) and then searching for the input difference ($\mathcal{ID}$) bit position with the highest differential bias in the obtained $\mathcal{OD}$ bit position. The primary aims of the proposed attack are to identify the best combination of the differential bias and PNB through PNB-focused analysis and to provide an upper bound on the number of rounds required for a differential attack that uses a single-bit truncated differential to be successful. Our contributions can be summarized as follows.

Comprehensive Analysis of PNB. By focusing on PNB analysis, we first clarify the distribution of the number of non-significant key bits in each round. Furthermore, we demonstrate that the number of non-significant key bits varies significantly depending on the $\mathcal{OD}$ bit position. In particular, all 0-th single-bits (*i.e.*, all the least significant bits) of each word in all intermediate rounds of reduced-round ChaCha are $\mathcal{OD}$ bit positions with a large number of non-significant key bits.

Upper Bound on the Number of Rounds for the Attacks. Based on the comprehensive analysis of the PNB, we examine the values of the average neutral measure for each round of the inverse round function. Consequently, we determine that the PNB-focused differential attack on reduced-round ChaCha should work on up to 7.25 rounds. In addition, our investigation suggests that the number of intermediate rounds must be at least 3.5 to improve the existing attacks [1, 3, 6–9, 16, 18].

Best Combinations of Differential Bias and PNB. Let $\Delta_i^{(r)}[j]$ be a single-bit difference for the j-th bit of the i-th word in the r-round internal state. By analyzing the differential biases at the obtained $\mathcal{OD}$ bit positions (*i.e.*, all 0-th single-bit positions of each word in 3.5 intermediate rounds), we report the $\mathcal{ID}$-$\mathcal{OD}$ pairs with a high differential bias to use in the attack, such as $(\Delta_{15}^{(0)}[6], \Delta_0^{(3.5)}[0])$, $(\Delta_{12}^{(0)}[6], \Delta_1^{(3.5)}[0])$, $(\Delta_{13}^{(0)}[6], \Delta_2^{(3.5)}[0])$, and $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$. Our investigation suggests that at least one of these $\mathcal{ID}$-$\mathcal{OD}$ pairs should yield the best combination of the differential bias and PNB.

Differential Attacks on Reduced-Round ChaCha. Based on the combinations of the differential bias and PNB, we present a differential attack on ChaCha20/7 with a time complexity of $2^{231.63}$, data complexity of $2^{49.58}$, and success probability of 0.5 using the $\mathcal{ID}$-$\mathcal{OD}$ pair of $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$. Furthermore, by using the $\mathcal{ID}$-$\mathcal{OD}$ pair of $(\Delta_{15}^{(0)}[6], \Delta_0^{(3.5)}[0])$, we present a differential attack on ChaCha20/7.25 with a time complexity of $2^{255.62}$, data complexity of $2^{48.36}$, and success probability of 0.5.

Table 1 summarizes our proposed attack as well as existing attacks on reduced-round ChaCha[2]. As illustrated in this table, our attack does not offer an improvement over the best existing attack on ChaCha20/7. However, we demonstrate that the PNB-focused differential attack on reduced-round ChaCha should work on up to 7.25 rounds. There have been no studies focusing on attacks on ChaCha20/7.25 thus far. It is crucial to thoroughly analyze the security evaluations of symmetric-key ciphers while gradually increasing the nonlinear operations, such as S-boxes and modular additions. In other words, it is important to thoroughly analyze the security of reduced-round ChaCha for each 0.25 round since the round function in ChaCha adds four wordwise modular additions every 0.25 round.

In conventional attacks on ChaCha, if the time complexity for the attack is beyond that of an exhaustive search for the unknown secret key, cryptanalysts

[2] According to [8], Coutinho and Neto stated that their initial results presented at EUROCRYPT 2021 [9] were erronous. That is, a differential attack on ChaCha20/7 with time and data complexities of $2^{228.51}$ and $2^{80.51}$, respectively, is infeasible. Furthermore, Coutinho and Neto presented a differential attack on ChaCha20/7 with time and data complexities of 2^{224} and 2^{224}, respectively [8]. This was similar to the best attacks on ChaCha20/7; however, verification is beyond the scope of this study because this was a distinguishing attack, not a key recovery attack.

Table 1. Summary of the proposed and existing key recovery attacks.

Target	Time	Data	Reference
ChaCha20/6	2^{139}	2^{30}	[1]
	2^{136}	2^{28}	[18]
	$2^{127.5}$	$2^{27.5}$	[6]
	$2^{102.2}$	2^{56}	[7]
	$2^{77.4}$	2^{58}	[3]
ChaCha20/7	2^{248}	2^{27}	[1]
	$2^{246.5}$	2^{27}	[18]
	$2^{242.59}$	$2^{69.58}$	[8]
	$2^{238.9}$	2^{96}	[16]
	$2^{237.7}$	2^{96}	[6]
	$2^{231.9}$	2^{50}	[7]
	$\mathbf{2^{231.63}}$	$\mathbf{2^{49.58}}$	**This work**
	$2^{230.86}$	$2^{48.8}$	[3]
ChaCha20/7.25	$\mathbf{2^{255.62}}$	$\mathbf{2^{48.36}}$	**This work**

utilize an approach that reduces the number of target rounds for the attack or selects an $\mathcal{ID}$-$\mathcal{OD}$ pair with a higher differential bias. In our approach, we focus on the fact that the PNB concept has a strong influence on the theoretical time complexity. We demonstrate the relevance of the comprehensive analysis of PNB for ChaCha for the first time and conclude that it is crucial to analyze not only differential biases but also PNB.

Organization. The rest of this paper is organized as follows. In Sect. 2, we briefly describe the ChaCha specification. In Sect. 3, we review generic techniques for the existing attack based on the PNB concept. In Sect. 4, we present and discuss the experimental results of the comprehensive analysis of PNB. In Sect. 5, we examine the differential bias at the $\mathcal{OD}$ bit position obtained in Sect. 4 and perform a differential attack on ChaCha20/7, ChaCha20/7.25, and ChaCha20/7.5. Finally, we summarize related works in Sect. 6 and conclude this study in Sect. 7.

2 Specification of ChaCha

ChaCha [4] performs the following three steps to generate a keystream block of 16 words, where the size of each word is 32 bits:

Step 1. The initial state matrix $X^{(0)}$ of order 4×4 is initialized from a 256-bit secret key $k = (k_0, k_1, \dots, k_7)$, a 96-bit nonce $v = (v_0, v_1, v_2)$, a 32-bit block counter t_0, and four 32-bit constants $c = (c_0, c_1, c_2, c_3)$, such as $c_0 = \mathrm{0x}61707865$, $c_1 = \mathrm{0x}3320646e$, $c_2 = \mathrm{0x}79622d32$, and $c_3 = \mathrm{0x}6b206574$.

After initialization, the following initial state matrix is obtained:

$$X^{(0)} = \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & x_2^{(0)} & x_3^{(0)} \\ x_4^{(0)} & x_5^{(0)} & x_6^{(0)} & x_7^{(0)} \\ x_8^{(0)} & x_9^{(0)} & x_{10}^{(0)} & x_{11}^{(0)} \\ x_{12}^{(0)} & x_{13}^{(0)} & x_{14}^{(0)} & x_{15}^{(0)} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & v_0 & v_1 & v_2 \end{pmatrix}.$$

Step 2. The round function of ChaCha comprises four simultaneous computations of the quarterround function. According to the procedure, a vector $(x_a^{(r)}, x_b^{(r)}, x_c^{(r)}, x_d^{(r)})$ in the internal state matrix $X^{(r)}$ is updated by sequentially computing the following:

$$\begin{cases} x_{a'}^{(r)} = x_a^{(r)} + x_b^{(r)};\ x_{d'}^{(r)} = x_d^{(r)} \oplus x_{a'}^{(r)};\ x_{d''}^{(r)} = x_{d'}^{(r)} \lll 16; \\ x_{c'}^{(r)} = x_c^{(r)} + x_{d''}^{(r)};\ x_{b'}^{(r)} = x_b^{(r)} \oplus x_{c'}^{(r)};\ x_{b''}^{(r)} = x_{b'}^{(r)} \lll 12; \\ x_a^{(r+1)} = x_{a'}^{(r)} + x_{b''}^{(r)};\ x_{d'''}^{(r)} = x_{d''}^{(r)} \oplus x_a^{(r+1)};\ x_d^{(r+1)} = x_{d'''}^{(r)} \lll 8; \\ x_c^{(r+1)} = x_{c'}^{(r)} + x_d^{(r+1)};\ x_{b'''}^{(r)} = x_{b''}^{(r)} \oplus x_c^{(r+1)};\ x_b^{(r+1)} = x_{b'''}^{(r)} \lll 7; \end{cases}$$

where the symbols "+", "$\oplus$", and "$\lll$" represent wordwise modular addition, bitwise XOR, and bitwise left rotation, respectively. For odd-numbered rounds, which are called columnrounds, the quarterround function is applied to the following four column vectors: $(x_0^{(r)}, x_4^{(r)}, x_8^{(r)}, x_{12}^{(r)})$, $(x_1^{(r)}, x_5^{(r)}, x_9^{(r)}, x_{13}^{(r)})$, $(x_2^{(r)}, x_6^{(r)}, x_{10}^{(r)}, x_{14}^{(r)})$, and $(x_3^{(r)}, x_7^{(r)}, x_{11}^{(r)}, x_{15}^{(r)})$. For even-numbered rounds, which are called diagonalrounds, the quarterround function is applied to the following four diagonal vectors: $(x_0^{(r)}, x_5^{(r)}, x_{10}^{(r)}, x_{15}^{(r)})$, $(x_1^{(r)}, x_6^{(r)}, x_{11}^{(r)}, x_{12}^{(r)})$, $(x_2^{(r)}, x_7^{(r)}, x_8^{(r)}, x_{13}^{(r)})$, and $(x_3^{(r)}, x_4^{(r)}, x_9^{(r)}, x_{14}^{(r)})$.

Step 3. A 512-bit keystream block is computed as $Z = X^{(0)} + X^{(R)}$, where R is the final round. The original version of ChaCha has $R = 20$ rounds, and the reduced-round version of ChaCha is denoted as ChaCha20/R.

The round function of ChaCha is reversible. In other words, an input vector $(x_a^{(r+1)}, x_b^{(r+1)}, x_c^{(r+1)}, x_d^{(r+1)})$ in the internal state matrix $X^{(r+1)}$ is backdated by sequentially computing the following:

$$\begin{cases} x_{b'''}^{(r)} = x_b^{(r+1)} \lll 25;\ x_{b''}^{(r)} = x_{b'''}^{(r)} \oplus x_c^{(r+1)};\ x_{c'}^{(r)} = x_c^{(r+1)} - x_d^{(r+1)}; \\ x_{d'''}^{(r)} = x_d^{(r+1)} \lll 24;\ x_{d''}^{(r)} = x_{d'''}^{(r)} \oplus x_a^{(r+1)};\ x_{a'}^{(r)} = x_a^{(r+1)} - x_{b''}^{(r)}; \\ x_{b'}^{(r)} = x_{b''}^{(r)} \lll 20;\ x_b^{(r)} = x_{b'}^{(r)} \oplus x_{c'}^{(r)};\ x_c^{(r)} = x_{c'}^{(r)} - x_{d''}^{(r)}; \\ x_{d'}^{(r)} = x_{d''}^{(r)} \lll 16;\ x_d^{(r)} = x_{d'}^{(r)} \oplus x_{a'}^{(r)};\ x_a^{(r)} = x_{a'}^{(r)} - x_b^{(r)}; \end{cases}$$

where the symbol "$-$" represents wordwise modular subtraction.

For a more accurate analysis of the round function, we further divide it into four rounds: 0.25, 0.5, 0.75, and 1 round. For example, the 0.25 round signifies that all quarterround functions in the round function have 0.25 round.

The 0.25-round quarterround function comprises one wordwise modular addition, one bitwise XOR, and one bitwise left rotation; thus, the ChaCha round function adds four wordwise modular additions every 0.25 round.

3 Differential Cryptanalysis of ChaCha

In this section, we review generic techniques for a differential attack based on the PNB concept, proposed by Aumasson et al. at FSE 2008 [1]. This attack comprises precomputation and online phases. In the precomputation phase, we examine single-bit differential biases and PNB and perform a probabilistic backward computation (PBC). Subsequently, we execute the online phase to recover the unknown key.

3.1 Precomputation Phase

Single-Bit Differential Biases. Let $x_i^{(r)}[j]$ be the j-th bit of the i-th word in the r-round internal state matrix $X^{(r)}$ for $0 \leq i \leq 15$ and $0 \leq j \leq 31$, and let $x_i'^{(r)}[j]$ be an associated bit with the difference $\varDelta_i^{(r)}[j] = x_i^{(r)}[j] \oplus x_i'^{(r)}[j]$. Based on the difference $\varDelta_i^{(0)}[j] = 1$ to the initial state matrix $X^{(0)}$, which is called the *input difference* or $\mathcal{ID}$, we obtain the corresponding initial state matrix $X'^{(0)}$. Then, we execute the round function of ChaCha using these initial state matrices $X^{(0)}$ and $X'^{(0)}$ as inputs and obtain $\varDelta_p^{(r)}[q] = x_p^{(r)}[q] \oplus x_p'^{(r)}[q]$ from the r-round output internal state matrices $X^{(r)}$ and $X'^{(r)}$, which is called the *output difference* or $\mathcal{OD}$. For a fixed key and all possible choices of nonces and block counters, the single-bit differential probability is defined as

$$\Pr\bigl(\varDelta_p^{(r)}[q] = 1 \mid \varDelta_i^{(0)}[j] = 1\bigr) = \frac{1}{2}(1 + \epsilon_d), \tag{1}$$

where ϵ_d denotes the $\mathcal{OD}$ bias. Note that we use the specified 1-bit $\mathcal{ID}$ and then obtain the truncated 1-bit $\mathcal{OD}$.

To estimate the number of samples to distinguish two distributions of random bit strings, we use the following theorem provided by Baignères et al. at ASIACRYPT 2004 [2].

Theorem 1 ([2, Theorem 6]). *Let $Z_1, \ldots, Z_n$ be independent and identically distributed random variables over the set $\mathcal{Z}$ of distribution D, D_0 and D_1 be two distributions of same support which are close to each other, and n be the number of samples of the best distinguisher between $\mathsf{D} = \mathsf{D}_0$ or $\mathsf{D} = \mathsf{D}_1$. Let d be a real number such that*

$$n = \frac{d}{\sum_{z \in \mathcal{Z}} \frac{\epsilon_z^2}{p_z}}, \tag{2}$$

where p_z and $p_z + \epsilon_z$ are probabilities of a random variable z following D_0 and D_1, respectively. Then, the overall probability of error is $P_e \approx \varPhi(-\sqrt{d}/2)$, where $\varPhi(\cdot)$ is the distribution function of the standard normal distribution.

Let D_0 and D_1 be the uniform distribution and a distribution of the truncated $\mathcal{OD}$ bit strings obtained from the internal state of ChaCha, respectively. In this case, the target event occurs in D_0 and D_1 with probabilities of $\frac{1}{2}$ and $\frac{1}{2}\cdot(1+\epsilon_d)$, respectively ($i.e.$, $p_0 = p_1 = \frac{1}{2}$ and $|\epsilon_0| = |\epsilon_1| = \frac{\epsilon_d}{2}$). Based on this, the number of samples of the best distinguisher between $\mathsf{D} = \mathsf{D}_0$ and $\mathsf{D} = \mathsf{D}_1$ can be estimated as $\frac{4}{\epsilon_d^2}$ with an overall probability of error of $P_e \approx \Phi(-\sqrt{4}/2) = \Phi(-1)$.

PNB. The PNB divides secret key bits into sets of m-bit significant and n-bit non-significant key bits. To differentiate between the sets, Aumasson et al. [1] focused on the degree of influence of each secret key bit on the $\mathcal{OD}$. The degree of influence is called the *neutral measure* and is defined as follows:

Definition 1 ([1, Definition 1]). *The neutral measure of the key bit position κ with respect to the $\mathcal{OD}$ is defined as γ_κ, where $\frac{1}{2}(1+\gamma_\kappa)$ is the probability that complementing the key bit κ does not change the $\mathcal{OD}$.*

For example, we have the following singular cases of neutral measure:

- $\gamma_i = 1$: $\mathcal{OD}$ does not depend on the i-th key bit ($i.e.$, it is non-significant).
- $\gamma_i = 0$: $\mathcal{OD}$ is statistically independent of the i-th key bit ($i.e.$, it is significant).
- $\gamma_i = -1$: $\mathcal{OD}$ linearly depends on the i-th key bit.

By performing the following steps, we compute the neutral measure and divide the secret key bits into two sets – a set of m-bit significant key bits and a set of n-bit non-significant key bits:

Step 1. Compute the R-round internal state matrix pair $(X^{(R)}, X'^{(R)})$ corresponding to the input pair $(X^{(0)}, X'^{(0)})$ with $\Delta_i^{(0)}[j] = 1$, and derive the keystream blocks $Z = X^{(0)} + X^{(R)}$ and $Z' = X'^{(0)} + X'^{(R)}$, respectively.

Step 2. Prepare a new input pair $(\overline{X}^{(0)}, \overline{X'}^{(0)})$ with the key bit position κ_i of the original input pair $(X^{(0)}, X'^{(0)})$ flipped by one bit.

Step 3. Compute the r-round internal state matrix pair $(Y^{(r)}, Y'^{(r)})$ for $r < R$ with $Z - \overline{X}^{(0)}$ and $Z' - \overline{X'}^{(0)}$ as inputs to the inverse round function of ChaCha.

Step 4. Compute $\Gamma_p^{(r)}[q] = y_p^{(r)}[q] \oplus y_p'^{(r)}[q]$ for the fixed $\mathcal{OD}$ bit, where $y_p^{(r)}[q]$ and $y_p'^{(r)}[q]$ denote the q-th bit of the p-th word of $Y^{(r)}$ and $Y'^{(r)}$, respectively.

Step 5. Repeat Steps 1–4 using different initial state matrices with the same $\Delta_i^{(0)}[j] = 1$, and compute the neutral measure as $\Pr(\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q] \mid \Delta_i^{(0)}[j] = 1) = \frac{1}{2}(1+\gamma_i)$, where $\Delta_p^{(r)}[q]$ is the $\mathcal{OD}$ obtained when searching for single-bit differential biases.

Step 6. Set a threshold γ and place all key bits with $\gamma_\kappa < \gamma$ into the set of m-bit significant key bits and those with $\gamma_\kappa \geq \gamma$ into the set of n-bit non-significant key bits.

PBC. As explained at the beginning of this subsection, we obtain r-round single-bit differential biases from the initial state matrices with the selected $\mathcal{ID}$, indicating that these biases can be obtained by performing forward computation in the target cipher. Moreover, we can obtain the r-round single-bit differential biases for ChaCha20/R from the obtained keystream by performing the following backward computation, which is called *PBC*:

Step 1. Compute the R-round internal state matrix pair $(X^{(R)}, X'^{(R)})$ corresponding to the input pair $(X^{(0)}, X'^{(0)})$ with $\Delta_i^{(0)}[j] = 1$, and derive the keystream blocks $Z = X^{(0)} + X^{(R)}$ and $Z' = X'^{(0)} + X'^{(R)}$, respectively.

Step 2. Prepare a new input pair $(\hat{X}^{(0)}, \hat{X}'^{(0)})$ with only non-significant key bits reset to a fixed value (*e.g.*, all zeros) from the original input pair $(X^{(0)}, X'^{(0)})$.

Step 3. Compute the r-round internal state matrix pair $(\hat{Y}^{(r)}, \hat{Y}'^{(r)})$ for $r < R$ with $Z - \hat{X}^{(0)}$ and $Z' - \hat{X}'^{(0)}$ as inputs to the inverse round function of ChaCha.

Step 4. Compute $\hat{\Gamma}_p^{(r)}[q] = \hat{y}_p^{(r)}[q] \oplus \hat{y}_p'^{(r)}[q]$ for the fixed $\mathcal{OD}$ bit, where $\hat{y}_p^{(r)}[q]$ and $\hat{y}_p'^{(r)}[q]$ are the q-th bit of the p-th word of $\hat{Y}^{(r)}$ and $\hat{Y}'^{(r)}$, respectively.

Step 5. Repeat Steps 1-4 using different initial state matrices with the same $\Delta_i^{(0)}[j] = 1$. Compute the r-round bias ϵ_a as $\Pr(\Delta_p^{(r)}[q] = \hat{\Gamma}_p^{(r)}[q] \mid \Delta_i^{(0)}[j] = 1) = \frac{1}{2}(1+\epsilon_a)$, where $\Delta_p^{(r)}[q]$ is the $\mathcal{OD}$ obtained when searching for single-bit differential biases.

The bias of $\hat{\Gamma}_p^{(r)}[q]$ is denoted by ϵ, that is, $\Pr(\hat{\Gamma}_p^{(r)}[q] = 1 \mid \Delta_i^{(0)}[j] = 1) = \frac{1}{2}(1+\epsilon)$. According to [1], the bias ϵ is approximated as $\epsilon_d \cdot \epsilon_a$ and is used to compute the overall complexity of the attack on the R-round target cipher.

3.2 Online Phase

After the precomputation phase, we perform the following steps to recover an unknown key:

Step 1. For an unknown key, collect N keystream block pairs where each pair is generated by a random input pair satisfying the relevant $\mathcal{ID}$.

Step 2. For each choice of the subkey (*i.e.*, m-bit significant key bits), the following steps should be performed:

Step 2-1. Derive the r-round single-bit differential biases from the obtained N keystream block pairs by performing backward computation.

Step 2-2. If the optimal distinguisher legitimates the subkeys candidate as (possibly) correct, perform an additional exhaustive search over the n-bit non-significant key bits to confirm the correctness of the filtered subkey and identify the n-bit non-significant key bits.

Step 2-3. Stop if the correct key is reported and output the recovered key.

Complexity Estimation. Given N keystream block pairs and a false alarm probability of $P_{fa} = 2^{-\alpha}$, the time complexity of the attack is

$$2^m(N + 2^n P_{fa}) = 2^m N + 2^{256-\alpha}, \text{ where } N \approx \left(\frac{\sqrt{\alpha \log 4} + 3\sqrt{1-\epsilon^2}}{\epsilon}\right)^2,$$

for a probability of non-detection $P_{nd} = 1.3 \times 10^{-3}$. In practice, α and thus N are selected to minimize the time complexity of the attack. Based on an existing study [1], we use the median bias ϵ in our attack; therefore, we note that our attack has a success probability of approximately 0.5.

4 Analysis of PNB

4.1 Search for PNB with High Neutral Measures

Typically, differential attacks on Salsa and ChaCha first determine the $\mathcal{ID}$-$\mathcal{OD}$ pair with a higher differential bias and then explore neutral measures of the target $\mathcal{OD}$ bit position. The existing studies [1, 3, 6–9, 16, 18] analyzed the differential bias and optimized the combination of the differential bias and PNB, as this combination can be used to determine the time and data complexities for the attack. Optimizing this combination by focusing on PNB analysis may help improve differential attacks on Salsa and ChaCha.

In this section, we perform a comprehensive analysis of the PNB and examine the conditions that produce a large number of non-significant key bits because the size of the PNB directly influences the theoretical time complexity of an attack, as described in Sect. 3.2. No study focusing on analyzing PNB has been conducted. If the conditions that produce a large number of non-significant key bits can be clarified, it can be claimed that existing attacks require improvement.

We perform the following procedure to search for conditions that produce a large number of non-significant key bits:

Step 1. Generate a known key $k = (k_0, \dots, k_7)$ uniformly at random.

Step 2. Select the $\mathcal{ID}$ bit position $\Delta_i^{(0)}[j]$, nonce, and block counter uniformly at random. Then, generate the initial state matrix $X^{(0)}$ and the corresponding initial matrix $X'^{(0)} = X^{(0)} \oplus \Delta_i^{(0)}[j]$.

Step 3. From the input pair $(X^{(0)}, X'^{(0)})$, compute the r-round internal state matrix pair $(X^{(r)}, X'^{(r)})$ and R-round internal state matrix pair $(X^{(R)}, X'^{(R)})$, where R is the target round for the attack on ChaCha20/R.

Step 4. From the r-round internal state matrix pair $(X^{(r)}, X'^{(r)})$, compute the $\mathcal{OD}$ for each bit, such as $\Delta_p^{(r)}[q] = X_p^{(r)}[q] \oplus X_p'^{(r)}[q]$ for all possible choices of p and q.

Step 5. From the R-round internal state matrix pair $(X^{(R)}, X'^{(R)})$, obtain keystream blocks $Z = X^{(0)} + X^{(R)}$ and $Z' = X'^{(0)} + X'^{(R)}$.

Step 6. Complement a particular key bit position κ ($\kappa \in \{0, \dots, 255\}$) to yield states $\overline{X}^{(0)}$ and $\overline{X'}^{(0)}$. Then, compute the r-round internal state matrix pair

$(Y^{(r)}, Y'^{(r)})$ with $Z - \overline{X}^{(0)}$ and $Z' - \overline{X'}^{(0)}$ as inputs to the inverse round function of ChaCha, and derive $\Gamma_p^{(r)}[q] = Y_p^{(r)}[q] \oplus Y_p'^{(r)}[q]$ for all possible choices of p and q.

Step 7. Increase the counter for each p, q, and κ only if $\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q]$.

Step 8. Repeat Steps 2–7 for the required number of samples.

After completing multiple trials with the above steps, we compute the neutral measures γ_κ for each key bit position and then count the number of non-significant key bits for each $\mathcal{OD}$ bit position with a specified threshold value γ. We note that the number of trials represents the number of different keys used in our experiments, while the number of samples represents the number of different initial state matrices generated from a fixed $\mathcal{ID}$ bit in each trial.

4.2 Experimental Results

This subsection presents our experimental results based on the search procedure described in Sect. 4.1. The following is our experimental environment: five Linux machines with 40-core Intel Xeon CPU E5-2660 v3 (2.60 GHz), 128.0 GB of main memory, a gcc 7.2.0 compiler, and the C programming language. We use the Mersenne Twister[3], which is a pseudorandom number generator proposed by Matsumoto and Nishimura [17], to generate the secret keys and samples used in all our experiments, and thus did not reuse secret keys and samples in any of the experiments.

To search for the conditions that produce a large number of non-significant key bits, we conduct experiments with 2^8 trials using 2^{21} samples for each of the possible 2^7 $\mathcal{ID}$s (*i.e.*, 2^{28} total samples). Based on Theorem 1, let D_0 and D_1 be the uniform distribution and a distribution of $\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q]$ obtained from the r-round internal state matrices of ChaCha20/R, respectively. The target event occurs in D_0 and D_1 with probabilities of $\frac{1}{2}$ and $\frac{1}{2} \cdot (1 + \gamma_\kappa)$, respectively; thus, the number of samples of the best distinguisher between $\mathsf{D} = \mathsf{D}_0$ and $\mathsf{D} = \mathsf{D}_1$ can be estimated as $\frac{4}{\gamma_\kappa^2}$. Our results are reliable when the derived neutral measures γ_κ are greater than 2^{-13} (≈ 0.000122), as 2^{28} samples are used.

ChaCha20/7. Fig. 1 presents the number of non-significant key bits for each $\mathcal{OD}$ bit position in ChaCha20/7. In this figure, the vertical axis represents the number of non-significant key bits at each $\mathcal{OD}$ bit position, the horizontal axis represents the $\mathcal{OD}$ bit position, and the auxiliary lines on the vertical axis separate the $\mathcal{OD}$ word positions (*i.e.*, the word positions are $0, 1, \ldots, 15$ in order from left to right). The blue (top), orange (center), and green (bottom) lines represent the number of non-significant key bits when the number of intermediate rounds r is 3, 3.5, and 4, respectively.

[3] The source code is available at `https://github.com/omitakahiro/omitakahiro.github.io/blob/master/random/code/MT.h`

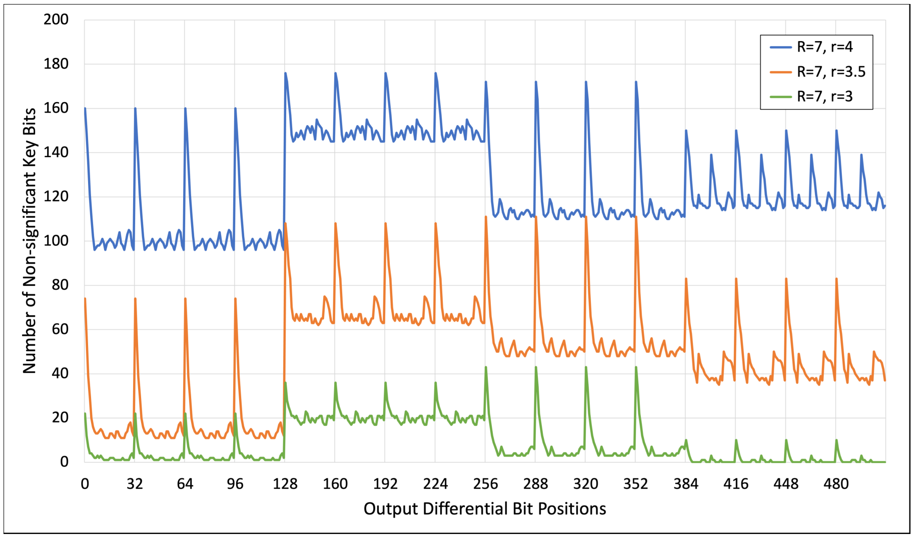

Fig. 1. Number of non-significant key bits for each $\mathcal{OD}$ bit position when the number of intermediate rounds r is 3, 3.5, and 4 in ChaCha20/7. We use $\gamma = 0.35$ as the threshold value.

Fig. 1 indicates that the number of non-significant key bits tends to be larger at all 0-th $\mathcal{OD}$ bit positions (*i.e.*, all least significant $\mathcal{OD}$ bit positions) of each word regardless of the number of intermediate rounds. Therefore, optimizing the combination of the differential bias and PNB by focusing on all 0-th $\mathcal{OD}$ bit positions may improve the differential attack on ChaCha20/7. Referring to the existing studies [1,16,18], the 0-th $\mathcal{OD}$ bit positions with a high average neutral measure were selected in the third round (*i.e.*, $\Delta_{11}^{(3)}[0]$); thus, it is difficult to improve the differential attack on ChaCha20/7 even for 3 intermediate rounds r. This is because the smaller the number of the intermediate rounds r, the smaller the number of non-significant key bits. Therefore, to improve the differential attack on ChaCha20/7, we should focus on more than 3 intermediate rounds.

The PNB analysis in this subsection cannot be directly compared with that in existing studies (*e.g.*, [3,6,11]) because a multi-bit differential or differential-linear technique was employed in the existing studies, whereas we focus solely on the single-bit differential technique. From a computational complexity perspective, we have searched for the number of non-significant key bits for only a single-bit $\mathcal{OD}$ bit position. Similarly, we should search for the number of non-significant key bits for multi-bit $\mathcal{OD}$ bit positions, which is left for future work.

ChaCha20/7.25, ChaCha20/7.5, and ChaCha20/7.75. Fig. 2 presents the number of non-significant key bits for each 3.5-round $\mathcal{OD}$ bit position when the number of target rounds R is 7, 7.25, 7.5, and 7.75. In this figure, the vertical and horizontal axes and the auxiliary lines on the vertical axis are the same as in Fig. 1. The blue (top), orange (second from the top), green (second from the bottom), and yellow (bottom) lines represent the number of non-significant

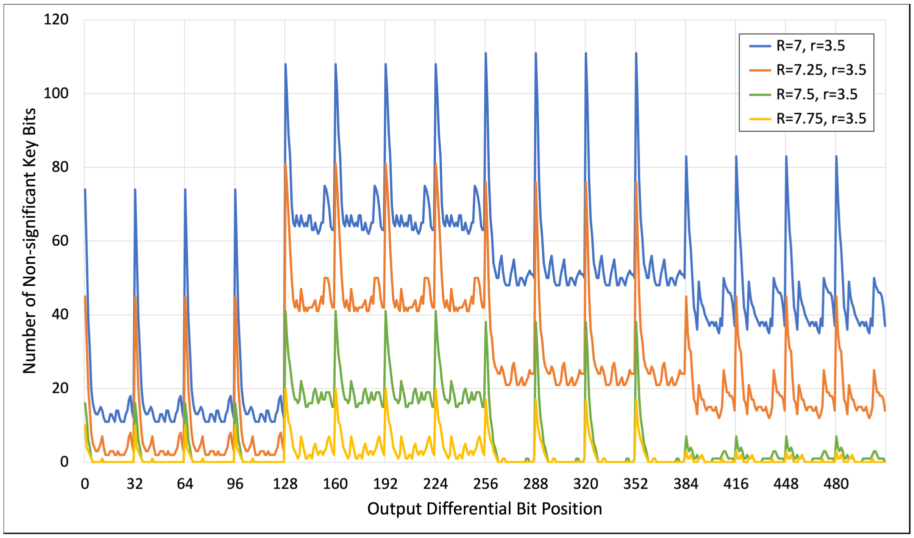

Fig. 2. Number of non-significant key bits for each $\mathcal{OD}$ bit position when the number of intermediate rounds r is 3.5 and number of target rounds R is 7, 7.25, 7.5, and 7.75. We use $\gamma = 0.35$ as the threshold value.

key bits when the number of intermediate rounds r is 3.5 and number of target rounds R is 7, 7.25, 7.5, and 7.75, respectively.

Similar to the experimental results for ChaCha20/7, the number of non-significant key bits tends to be larger at all 0-th $\mathcal{OD}$ bit positions of each word regardless of the number of target rounds. Therefore, optimizing the combination of the differential bias and PNB by focusing on all 0-th $\mathcal{OD}$ bit positions may be effective for performing a differential attack on ChaCha20/7.25, ChaCha20/7.5, and ChaCha20/7.75.

4.3 Discussion

Relationship between PNB and Inverse Round Function. We discuss the relationship between the PNB (or the number of non-significant key bits) and inverse round function of ChaCha. To this end, we investigate the relationship between the input word position to the inverse quarterround function and the cumulative number of wordwise modular subtractions. This is because wordwise modular addition/subtraction plays a crucial role in ensuring the security of ARX ciphers. In our investigation, the cumulative number of wordwise modular subtractions is counted as follows:

Wordwise modular subtraction. The cumulative number of wordwise modular subtractions is counted only when wordwise modular subtraction is executed. Moreover, we calculate the sum of the cumulative number of wordwise modular subtractions in two input words to wordwise modular subtraction. For example, when the wordwise modular subtraction, $A' = A - B$, is executed and the cumulative number of wordwise modular subtractions in the

Table 2. Relationship between the input word position to the inverse quarterround function and the cumulative number of modular subtractions when the number of target rounds R is 7 or 7.5.

Input word position	Cumulative number of modular subtractions for $R - r$ rounds.				
	3 rounds ($r = 4$ or 4.5)	3.25 rounds ($r = 3.75$ or 4.25)	3.5 rounds ($r = 3.5$ or 4)	3.75 rounds ($r = 3.25$ or 3.75)	4 rounds ($r = 3$ or 3.5)
A	70	70	156	156	349
B	37	85	85	192	192
C	48	107	107	236	236
D	58	128	128	128	284

Table 3. Relationship between the input word position to the inverse quarterround function and the cumulative number of modular subtractions when the number of target rounds R is 7.25 or 7.75.

Input word position	Cumulative number of modular subtractions for $R - r$ rounds.				
	3 rounds ($r = 4.25$ or 4.75)	3.25 rounds ($r = 4$ or 4.5)	3.5 rounds ($r = 3.75$ or 4.25)	3.75 rounds ($r = 3.5$ or 4)	4 rounds ($r = 3.25$ or 3.75)
A	48	107	107	236	236
B	58	58	128	128	284
C	70	70	156	156	349
D	37	85	85	192	192

two input words A and B are 70 and 85, respectively, 156 is the cumulative number of wordwise modular subtractions in the output word A'.

Bitwise XOR. We calculate only the sum of the cumulative number of wordwise modular subtractions in two input words to bitwise XOR. For example, when the bitwise XOR operation, $B' = B \oplus C$, is executed and the cumulative number of wordwise modular subtractions in the two input words B and C are 37 and 48, respectively, 85 is the cumulative number of wordwise modular subtractions in the output word B'.

Bitwise left rotation. The cumulative number of wordwise modular subtractions did not change after the execution of bitwise left rotation.

Tables 2 and 3 present the results of examining the cumulative number of wordwise modular subtractions. In Table 2, the number of target rounds R is 7 or 7.5, whereas in Table 3, the number of target rounds R is 7.25 or 7.75. In these tables, the input word position column contains the word positions, such as a vector (A, B, C, D), input to the inverse quarterround function. Note that each input word position always transitions to the same input word position in the next round (refer to Sect. 2 for more details).

Tables 2 and 3 indicate that the cumulative number of wordwise modular subtractions differ depending on the input word position relative to the inverse round function and the number of intermediate rounds r. In particular, the cumulative number of wordwise modular subtractions is smaller in the order of the input word positions B, C, D, and A when the number of intermediate rounds r is 3, 3.5, 4, and 4.5. In contrast, the cumulative number of wordwise modular subtractions is smaller in the order of the input word positions D, A, B, and C when the number of intermediate rounds r is 3.25, 3.75, 4.25, and 4.75.

Table 4. Maximum, minimum, average, and median values of the average neutral measures $\hat{\gamma}_\kappa$ for each target round R when $r = 3.5$, where p and q are the word and bit positions of the $\mathcal{OD}$, respectively (*i.e.*, $\Delta_p^{(r)}[q]$).

R	Maximum			Minimum			Average	Median
	$\hat{\gamma}_\kappa$	p	q	$\hat{\gamma}_\kappa$	p	q		
7	0.382	11	0	0.050	2	13	0.169	0.174
7.25	0.282	6	0	0.018	3	13	0.097	0.087
7.5	0.151	4	0	0.004	0	13	0.034	0.016
7.75	0.075	9	0	0.001	0	13	0.011	0.005

We now compare the experimental results presented in Fig. 2 with the results when $r = 3.5$, as illustrated in Tables 2 and 3. Note that the range of input word positions A, B, C, and D corresponds to the $\mathcal{OD}$ bit positions 0 to 127, 128 to 255, 256 to 383, and 384 to 511, respectively. From Fig. 2, the number of non-significant key bits is larger in the order of the input word positions B, C, D, and A when the number of intermediate rounds r is 3.5 (all 0-th bit positions are exceptions); thus, the smaller the cumulative number of wordwise modular subtractions, the larger the number of non-significant key bits. The 0-th bit position is uninfluenced by the carry-in wordwise modular subtraction (*i.e.*, it is uninfluenced by the $\mathcal{ID}/\mathcal{OD}$). This has been suggested to be a special case.

In summary, the number of non-significant key bits depends on the input word position relative to the inverse round function and is affected by the cumulative number of wordwise modular subtractions. Specifically, the conditions that produce the number of non-significant key bits depend on the $\mathcal{OD}$ bit position, particularly all 0-th $\mathcal{OD}$ bit positions.

Upper Bound on the Number of Rounds for the Attacks. We discuss the upper bound on the number of rounds required for a PNB-focused differential attack that uses a single-bit truncated differential to be successful. To this end, we investigate the value of the average neutral measures $\hat{\gamma}_\kappa$ for each round of the inverse round function. Table 4 presents the maximum, minimum, average, and median values of the average neutral measures $\hat{\gamma}_\kappa$ for each target round R when the number of intermediate rounds r is 3.5[4]. These findings can be obtained by a detailed analysis of the experimental results described in Sect. 4.2. The R column in Table 4 lists the number of target rounds for our attack, and the number of rounds of the inverse round function can be calculated as $R - r$.

Our experimental results are reliable when the derived average neutral measures $\hat{\gamma}_\kappa$ are greater than 2^{-13} (≈ 0.000122), as 2^{28} samples are used. As illustrated in Table 4, all values of $\hat{\gamma}_\kappa$ are reliable when the number of target rounds R is 7, 7.25, 7.5, and 7.75; thus, the upper bound on the number of rounds

[4] The latest study presented by Coutinho and Neto at EUROCRYPT 2021 [9] used $\Delta_5^{(3.5)}[0]$ ($= \Delta_5^{(4)}[7] \oplus \Delta_{10}^{(4)}[0]$) as the $\mathcal{OD}$ to perform a differential attack on ChaCha20/7. Accordingly, we focused solely on $r = 3.5$.

required for a PNB-focused differential attack that uses a single-bit truncated differential to be successful is at most 7.75 rounds. However, given that the threshold γ used in the existing attacks, such as [3,6,9], was $\gamma = 0.27$ or 0.35, it is practically difficult to perform a differential attack when the number of target rounds R is 7.5 or 7.75 because $\hat{\gamma}_\kappa$ is too small; thus, our results suggest that a PNB-focused differential attack on reduced-round ChaCha should work on up to 7.25 rounds. To verify this claim, we perform a PNB-focused differential attack on reduced-round ChaCha with target rounds of 7, 7.25, and 7.5.

5 PNB-focused Differential Attack

In this section, we describe a PNB-focused differential attack on reduced-round ChaCha. First, based on the PNB analysis described in Sect. 4, we determine the target $\mathcal{OD}$ bit position for the proposed attack. Next, we analyze the differential biases at the target $\mathcal{OD}$ bit positions and then obtain the $\mathcal{ID}$ bit position with the best differential bias at the target $\mathcal{OD}$ bit positions. Finally, we estimate the time and data complexities for our attack using the combination of the differential bias and PNB.

5.1 Analysis of Single-Bit Differential Biases

In Sect. 4, we comprehensively analyze the PNB for all possible single-bit truncated $\mathcal{OD}$s. Accordingly, by analyzing the $\mathcal{ID}$ bit position with the highest differential bias at the target $\mathcal{OD}$ bit position, we can determine the $\mathcal{ID}$-$\mathcal{OD}$ pair to use for our attack.

To identify the $\mathcal{ID}$ bit position with the highest differential bias $|\epsilon_d|$ at the target $\mathcal{OD}$ bit positions, we conduct experiments with 2^6 trials using 2^{28} samples for a fixed $\mathcal{ID}$; thus, the results are reliable when the derived differential biases $|\epsilon_d|$ are greater than 2^{-13} ($\approx$ 0.000122), as 2^{28} samples are used. In our experiments, the target $\mathcal{OD}$s are $\varDelta_0^{(3.5)}[0]$, $\varDelta_1^{(3.5)}[0]$, $\varDelta_2^{(3.5)}[0]$, $\varDelta_3^{(3.5)}[0]$, $\varDelta_{12}^{(3.5)}[0]$, $\varDelta_{13}^{(3.5)}[0]$, $\varDelta_{14}^{(3.5)}[0]$, and $\varDelta_{15}^{(3.5)}[0]$. Consequently, our results are reliable at $\varDelta_0^{(3.5)}[0]$, $\varDelta_1^{(3.5)}[0]$, $\varDelta_2^{(3.5)}[0]$, and $\varDelta_3^{(3.5)}[0]$ because these absolute biases are at least 0.000430, but not at $\varDelta_{12}^{(3.5)}[0]$, $\varDelta_{13}^{(3.5)}[0]$, $\varDelta_{14}^{(3.5)}[0]$, and $\varDelta_{15}^{(3.5)}[0]$ because these absolute biases are at most 0.000028. Moreover, these results lead to unreliable at other 0-th $\mathcal{OD}$ bit positions, such as $\varDelta_4^{(3.5)}[0]$, $\varDelta_5^{(3.5)}[0]$, $\varDelta_6^{(3.5)}[0]$, $\varDelta_7^{(3.5)}[0]$, $\varDelta_8^{(3.5)}[0]$, $\varDelta_9^{(3.5)}[0]$, $\varDelta_{10}^{(3.5)}[0]$, and $\varDelta_{11}^{(3.5)}[0]$, because the results are affected by the unreliable results at $\varDelta_{12}^{(3.5)}[0]$, $\varDelta_{13}^{(3.5)}[0]$, $\varDelta_{14}^{(3.5)}[0]$, and $\varDelta_{15}^{(3.5)}[0]$ according to the computations of the quarterround function (see Sect. 2 for details). Consequently, we determine the following $\mathcal{ID}$-$\mathcal{OD}$ pairs to use for our attack: $(\varDelta_{15}^{(0)}[6], \varDelta_0^{(3.5)}[0])$, $(\varDelta_{12}^{(0)}[6], \varDelta_1^{(3.5)}[0])$, $(\varDelta_{13}^{(0)}[6], \varDelta_2^{(3.5)}[0])$, and $(\varDelta_{14}^{(0)}[6], \varDelta_3^{(3.5)}[0])$.

To obtain more precise single-bit differential biases for the derived $\mathcal{ID}$-$\mathcal{OD}$ pairs, we conduct additional experiments with 2^8 trials using 2^{34} samples for a fixed $\mathcal{ID}$; thus, the results are reliable when the derived differential biases $|\epsilon_d|$ are greater than 2^{-16} ($\approx$ 0.000015), as 2^{34} samples are used. Table 5 lists the

Table 5. Best single-bit differential biases $|\epsilon_d|$ at the 0-th $\mathcal{OD}$ bit positions of each word for 3.5 rounds of ChaCha. Experiments are conducted with 2^8 trials using 2^{34} samples for a fixed $\mathcal{ID}$; thus, the results are reliable when the derived differential biases $|\epsilon_d|$ are greater than 2^{-16} (≈ 0.000015), as 2^{34} samples are used.

$\mathcal{ID}$	$\mathcal{OD}$	$\lvert\epsilon_d\rvert$
$\Delta_{15}^{(0)}[6]$	$\Delta_{0}^{(3.5)}[0]$	0.000469
$\Delta_{12}^{(0)}[6]$	$\Delta_{1}^{(3.5)}[0]$	0.000478
$\Delta_{13}^{(0)}[6]$	$\Delta_{2}^{(3.5)}[0]$	0.000504
$\Delta_{14}^{(0)}[6]$	$\Delta_{3}^{(3.5)}[0]$	0.000478

additional experimental results of the best differential biases $|\epsilon_d|$ at the target $\mathcal{OD}$ bit positions: $\Delta_0^{(3.5)}[0]$, $\Delta_1^{(3.5)}[0]$, $\Delta_2^{(3.5)}[0]$, and $\Delta_3^{(3.5)}[0]$. As displayed in this table, we can obtain reliable results at the target positions; then, we use the listed biases $|\epsilon_d|$ to estimate the time and data complexities for our attack.

5.2 Complexity Estimation

To estimate the time and data complexities for the PNB-focused differential attack on the target rounds of ChaCha (*i.e.*, 7, 7.25, and 7.5 rounds), the remaining steps should be performed as follows (see Sect. 3 for details):

Step 1. Recalculate the neutral measures corresponding to the derived $\mathcal{ID}$-$\mathcal{OD}$ pairs and divide the secret key bits into two sets – a set of m-bit significant and a set of n-bit non-significant key bits.

Step 2. By performing PBC, obtain the biases $|\epsilon_a|$ for each threshold γ from the obtained keystream and approximate the overall bias $\epsilon \approx \epsilon_d \cdot \epsilon_a$ for the attack on the target rounds of ChaCha.

Step 3. Perform the online phase and estimate the time and data complexities to recover the unknown key, as described in Sect. 3.2.

To perform the above-mentioned steps, we conduct experiments with 2^8 trials using 2^{30} samples for the fixed $\mathcal{ID}$; thus, the results are reliable when the derived biases $|\epsilon_a|$ are greater than 2^{-14} (≈ 0.000061), as 2^{30} samples are used.

ChaCha20/7. Table 6 presents the best parameters for each target $\mathcal{ID}$-$\mathcal{OD}$ pair to estimate the time and data complexities for our attack on ChaCha20/7. The threshold γ is set from 0.10 to 0.95 at intervals of 0.05 (*i.e.*, total 18 patterns), n represents the number of non-significant key bits, $|\epsilon_d|$ is derived from Table 5, $|\epsilon_a|$ is obtained by performing PBC for each threshold γ, and α is selected to minimize the time complexity of our attack.

Consequently, we can perform our attack on ChaCha20/7 with time and data complexities of $2^{231.63}$ and $2^{49.58}$, respectively, using the best parameters, where the $\mathcal{ID}$-$\mathcal{OD}$ pair is $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$, γ is 0.35, n is 74, α is 29, and the list of PNB is {6, 7, 8, 9, 10, 11, 12, 13, 14, 19, 27, 28, 29, 30, 31, 34, 35, 36, 37, 46,

Table 6. Best parameters for the proposed attack on ChaCha20/7.

$\mathcal{ID}$	$\mathcal{OD}$	γ	n	$\lvert\epsilon_d\rvert$	$\lvert\epsilon_a\rvert$	α	Time	Data
$\varDelta_{15}^{(0)}[6]$	$\varDelta_{0}^{(3.5)}[0]$	0.35	74	0.000469	0.000662	29	$2^{231.74}$	$2^{49.68}$
$\varDelta_{12}^{(0)}[6]$	$\varDelta_{1}^{(3.5)}[0]$	0.35	74	0.000478	0.000556	29	$2^{232.17}$	$2^{50.13}$
$\varDelta_{13}^{(0)}[6]$	$\varDelta_{2}^{(3.5)}[0]$	0.35	74	0.000504	0.000615	29	$2^{231.74}$	$2^{49.69}$
$\varDelta_{14}^{(0)}[6]$	$\varDelta_{3}^{(3.5)}[0]$	0.35	74	0.000478	0.000674	29	$2^{231.63}$	$2^{49.58}$

Table 7. Best parameters for the proposed attack on ChaCha20/7.25.

$\mathcal{ID}$	$\mathcal{OD}$	γ	n	$\lvert\epsilon_d\rvert$	$\lvert\epsilon_a\rvert$	α	Time	Data
$\varDelta_{15}^{(0)}[6]$	$\varDelta_{0}^{(3.5)}[0]$	0.30	49	0.000469	0.000564	3	$2^{255.62}$	$2^{48.36}$
$\varDelta_{12}^{(0)}[6]$	$\varDelta_{1}^{(3.5)}[0]$	0.35	45	0.000478	0.002200	3	$2^{255.64}$	$2^{44.38}$
$\varDelta_{13}^{(0)}[6]$	$\varDelta_{2}^{(3.5)}[0]$	0.35	45	0.000504	0.001783	2	$2^{256.02}$	$2^{44.61}$
$\varDelta_{14}^{(0)}[6]$	$\varDelta_{3}^{(3.5)}[0]$	0.35	45	0.000478	0.002186	3	$2^{255.65}$	$2^{44.40}$

71, 79, 80, 83, 98, 99, 100, 101, 102, 103, 104, 105, 106, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 122, 123, 127, 128, 129, 130, 148, 149, 150, 159, 187, 188, 189, 190, 191, 200, 223, 224, 225, 231, 232, 239, 240, 243, 244, 251, 252, 253, 254, 255}.

ChaCha20/7.25 and ChaCha20/7.5. Similar to the complexity estimation for ChaCha20/7, we present the best parameters for each target $\mathcal{ID}$-$\mathcal{OD}$ pair to estimate the time and data complexities for our attack on ChaCha20/7.25 and ChaCha20/7.5 in Tables 7 and 8, respectively.

As illustrated in Table 7, our attack on ChaCha20/7.25 can be performed with time and data complexities of $2^{255.62}$ and $2^{48.36}$, respectively, using the best parameters, where the $\mathcal{ID}$-$\mathcal{OD}$ pair is $(\varDelta_{15}^{(0)}[6], \varDelta_{0}^{(3.5)}[0])$, γ is 0.30, n is 49, α is 3, and the list of PNB is {2, 3, 10, 13, 14, 19, 20, 26, 27, 31, 40, 44, 45, 46, 51, 59, 60, 61, 62, 63, 128, 129, 130, 135, 136, 143, 144, 147, 148, 155, 156, 157, 158, 159, 160, 161, 162, 180, 181, 182, 191, 219, 220, 221, 222, 223, 224, 232, 255}. ChaCha provides a 256-bit security level against key recovery attacks. Given that the success probability is approximately 0.5, our attack on ChaCha20/7.25 is slightly less efficient than a brute force attack; however, it is the first dedicated attack on the target to be reported. It provides both a baseline and useful components (*i.e.*, differential bias and PNB) for improved attacks.

In addition, as displayed in Table 8, our attack on ChaCha20/7.5 can be performed with time and data complexities of $2^{273.49}$ and $2^{37.49}$, respectively, using the best parameters, where the $\mathcal{ID}$-$\mathcal{OD}$ pair is $(\varDelta_{15}^{(0)}[6], \varDelta_{0}^{(3.5)}[0])$, γ is 0.30, n is 20, α is 1, and the list of PNB is {6, 7, 14, 22, 25, 31, 39, 40, 41, 42, 56, 57, 58, 63, 191, 219, 220, 221, 222, 223}. Thus, our attack on ChaCha20/7.5 is inefficient because this is beyond the security level of ChaCha.

Table 8. Best parameters for the proposed attack on ChaCha20/7.5.

$\mathcal{ID}$	$\mathcal{OD}$	γ	n	$\lvert\epsilon_d\rvert$	$\lvert\epsilon_a\rvert$	α	Time	Data
$\Delta_{15}^{(0)}[6]$	$\Delta_{0}^{(3.5)}[0]$	0.30	20	0.000469	0.020269	1	$2^{273.49}$	$2^{37.49}$
$\Delta_{12}^{(0)}[6]$	$\Delta_{1}^{(3.5)}[0]$	0.30	20	0.000478	0.014840	1	$2^{274.33}$	$2^{38.33}$
$\Delta_{13}^{(0)}[6]$	$\Delta_{2}^{(3.5)}[0]$	0.30	20	0.000504	0.017594	1	$2^{273.69}$	$2^{37.69}$
$\Delta_{14}^{(0)}[6]$	$\Delta_{3}^{(3.5)}[0]$	0.30	20	0.000478	0.018693	1	$2^{273.67}$	$2^{37.67}$

6 Related Works

Aumasson et al. [1] proposed a framework for a differential attack based on the PNB concept and applied it to reduced-round Salsa, ChaCha, and Rumba. They first obtained an $\mathcal{ID}$-$\mathcal{OD}$ pair, $(\Delta_{13}^{(0)}[13], \Delta_{11}^{(3)}[0])$, with a high differential bias using a single-bit differential technique. Then, they determined the PNB at the target $\mathcal{OD}$ bit position and estimated the time and data complexities for their attack on ChaCha20/7. Their attack can be performed with time and data complexities of 2^{248} and 2^{27}, respectively.

Shi et al. [18] proposed new techniques, called the column chaining distinguisher (CCD) and probabilistic neutral vector (PNV) concept, to improve Aumasson et al.'s attack. They used the same $\mathcal{ID}$-$\mathcal{OD}$ pair, $(\Delta_{13}^{(0)}[13], \Delta_{11}^{(3)}[0])$, obtained by Aumasson et al., constructed a 4-step CCD, determined the PNV at the target $\mathcal{OD}$ bit position, and estimated the time and data complexities as well as the success probability for their attack on ChaCha20/7. Their attack can be performed with time and data complexities of $2^{246.5}$ and 2^{27}, respectively, and a success probability of approximately 0.43.

Maitra [16] further improved Aumasson et al.'s attack by using the chosen-IV technique. Maitra used the same $\mathcal{ID}$-$\mathcal{OD}$ pair, $(\Delta_{13}^{(0)}[13], \Delta_{11}^{(3)}[0])$, obtained by Aumasson et al. and explored how to appropriately select IVs corresponding to the secret keys, given the target $\mathcal{ID}$, $\Delta_{13}^{(0)}[13]$. This attack can be performed on ChaCha20/7 with the time and data complexities of $2^{238.94}$ and $2^{23.89}$, respectively.

Choudhuri and Maitra [6] used a differential-linear technique to extend the existing 3-round single-bit differential, $(\Delta_{13}^{(0)}[13], \Delta_{11}^{(3)}[0])$, to 4-, 4.5-, and 5-round multi-bit differentials, such that the 4.5-round $\mathcal{OD}$ is $\Delta_{0}^{(4.5)}[0] \oplus \Delta_{0}^{(4.5)}[8] \oplus \Delta_{1}^{(4.5)}[0] \oplus \Delta_{5}^{(4.5)}[12] \oplus \Delta_{11}^{(4.5)}[0] \oplus \Delta_{9}^{(4.5)}[0] \oplus \Delta_{15}^{(4.5)}[0] \oplus \Delta_{12}^{(4.5)}[16] \oplus \Delta_{12}^{(4.5)}[24]$. Using such multi-bit differentials, their attack on ChaCha20/7 can be performed with time and data complexities of $2^{237.65}$ and $2^{31.6}$, respectively.

Beierle et al. [3] presented a generic framework for differential-linear attacks with a special focus on ARX ciphers. Then, they applied this framework to ChaCha20/7 and improved the best existing attacks. To perform a differential-linear attack on ChaCha20/7, the target cipher is divided into a differential part covering 1 round, a middle part covering 2.5 rounds, a linear part covering 2.5 rounds, and a key guessing part covering 1 round. As a result, their attack can be performed on ChaCha20/7 with time and data complexities of $2^{230.86}$ and $2^{48.83}$, respectively.

As summarized above, the best existing attack on reduced-round ChaCha works on up to 7 rounds with time and data complexities of $2^{230.86}$ and $2^{48.83}$, respectively. Our attack has the time and data complexities of $2^{231.63}$ and $2^{49.58}$, respectively; thus, it is not an improvement over the best existing attack on ChaCha20/7. However, our analysis suggests that a PNB-focused differential attack on reduced-round ChaCha should work on up to 7.25 rounds. No study focusing on attacks on ChaCha20/7.25 has been conducted until now. Although the proposed attack on ChaCha20/7.25 is less efficient than a brute force attack, it is the first dedicated attack on the target. It provides both a baseline and useful components (*i.e.*, differential bias and PNB) for improved attacks.

7 Conclusion

In this study, we have proposed a new approach for differential cryptanalysis against the ChaCha stream cipher. Our approach focuses on analyzing PNB rather than searching for differential biases; therefore, we refer to the proposed approach as a *PNB-focused differential attack*. The proposed approach allows us to perform the most effective differential attack on the 7.25-round ChaCha (*i.e.*, ChaCha20/7.25) with a time complexity of $2^{255.62}$, a data complexity of $2^{48.36}$, and a success probability of 0.5. Although this attack is less efficient than a brute force attack, it is the first dedicated attack on the target. It provides both a baseline and useful components (*i.e.*, differential bias and PNB) for improved attacks.

Our work can be extended in the following directions in the future. First, in this study, we have focused solely on the truncated single-bit differential technique. However, it may be possible to improve the proposed attack by employing multi-bit differential or differential-linear techniques, especially in the framework proposed by Beierle et al. [3]. In addition, our analysis have not fully considered both the differential bias and PNB to obtain the best combination because these characteristics can be analyzed independently. The next step is thus to consider these characteristics together to obtain stricter evaluation results. Finally, the PNB-focused differential attack can be used to improve existing differential attacks on the Salsa stream cipher.

Acknowledgment

We would like to thank the reviewers for their valuable feedback that helped improve the quality of our paper. This work is partially supported by JSPS KAKENHI Grant Number JP21H03443, and Innovation Platform for Society 5.0 at MEXT.

References

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 *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 470–488. Springer, 2008.
2. Thomas Baignères, Pascal Junod, and Serge Vaudenay. How Far Can We Go Beyond Linear Cryptanalysis? In *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004.
3. Christof Beierle, Gregor Leander, and Yosuke Todo. Improved Differential-Linear Attacks with Applications to ARX Ciphers. In *CRYPTO (3)*, volume 12172 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.
4. Daniel J. Bernstein. ChaCha, A Variant of Salsa20. In *Workshop Record of SASC*, volume 8, 2008.
5. Daniel J. Bernstein. The Salsa20 Family of Stream Ciphers. In *The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.
6. Arka Rai Choudhuri and Subhamoy Maitra. Significantly Improved Multi-bit Differentials for Reduced Round Salsa and ChaCha. *IACR Trans. Symmetric Cryptol.*, 2016(2):261–287, 2016.
7. Murilo Coutinho and T. C. Souza Neto. New Multi-bit Differentials to Improve Attacks Against ChaCha. *IACR Cryptol. ePrint Arch.*, page 350, 2020.
8. Murilo Coutinho and T. C. Souza Neto. Improved Linear Approximations to ARX Ciphers and Attacks Against ChaCha. *IACR Cryptol. ePrint Arch.*, page 224, 2021.
9. Murilo Coutinho and Tertuliano C. Souza Neto. Improved Linear Approximations to ARX Ciphers and Attacks Against ChaCha. In *EUROCRYPT (1)*, volume 12696 of *Lecture Notes in Computer Science*, pages 711–740. Springer, 2021.
10. Kakumani K. C. Deepthi and Kunwar Singh. Cryptanalysis of Salsa and ChaCha: Revisited. In *MONAMI*, volume 235 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 324–338. Springer, 2017.
11. Sabyasachi Dey and Santanu Sarkar. Improved analysis for reduced round Salsa and Chacha. *Discret. Appl. Math.*, 227:58–69, 2017.
12. Sabyasachi Dey and Santanu Sarkar. Proving the biases of Salsa and ChaCha in differential attack. *Des. Codes Cryptogr.*, 88(9):1827–1856, 2020.
13. Sabyasachi Dey and Santanu Sarkar. A theoretical investigation on the distinguishers of Salsa and ChaCha. *Discret. Appl. Math.*, 302:147–162, 2021.
14. Tsukasa Ishiguro, Shinsaku Kiyomoto, and Yutaka Miyake. Latin Dances Revisited: New Analytic Results of Salsa20 and ChaCha. In *ICICS*, volume 7043 of *Lecture Notes in Computer Science*, pages 255–266. Springer, 2011.
15. Ryoma Ito. Rotational Cryptanalysis of Salsa Core Function. In *ISC*, volume 12472 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2020.
16. Subhamoy Maitra. Chosen IV cryptanalysis on reduced round ChaCha and Salsa. *Discret. Appl. Math.*, 208:88–97, 2016.
17. Makoto Matsumoto and Takuji Nishimura. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.
18. Zhenqing Shi, Bin Zhang, Dengguo Feng, and Wenling Wu. Improved Key Recovery Attacks on Reduced-Round Salsa20 and ChaCha. In *ICISC*, volume 7839 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2012.