PNB-based Differential Cryptanalysis of ChaCha Stream Cipher

Shotaro Miyashita¶, Ryoma Ito§, and Atsuko Miyaji¶

¶Osaka University, Japan.
§National Institute of Information and Communications Technology (NICT), Japan.

November 18, 2021

Abstract

In this study, we focus on the differential cryptanalysis of the ChaCha stream cipher. In the conventional approach, an adversary first searches for the input/output differential pair with the best differential bias and then analyzes the *probabilistic neutral bits* (PNB) in detail based on the obtained input/output differential pair. However, although time and data complexities for the attack can be estimated by the differential bias and PNB obtained in this approach, their combination does not always represent the best. In addition, a comprehensive analysis of the PNB was not provided in existing studies; they have not clarified the upper bounds of the number of rounds required for the differential attack based on the PNB to be successful. To solve these problems, we proposed a *PNB-based differential attack* on the reduced-round ChaCha by first comprehensively analyzing the PNB at all output differential bit positions and then searching for the input/output differential pair with the best differential bias based on the obtained PNB. By comprehensively analyzing the PNB, we clarified that an upper bound of the number of rounds required for the PNB-based differential attack to be successful was 7.25 rounds. As a result, the proposed attack can work on the 7.25-round ChaCha with time and data complexities of $2^{255.62}$ and $2^{37.49}$, respectively. Further, using the existing differential bias presented by Coutinho and Neto at EUROCRYPT 2021, we further improved the attack on the 7.25-round ChaCha with time and data complexities of $2^{244.22}$ and $2^{69.14}$, respectively. The best existing attack on ChaCha, proposed by Coutinho and Neto at EUROCRYPT 2021, works on up to 7 rounds with time and data complexities of $2^{228.51}$ and $2^{80.51}$, respectively. Therefore, we improved the best existing attack on the reduced-round ChaCha. We believe that this study will be the first step towards an attack on more rounds of ChaCha, e.g., the 8-round ChaCha.

1 Introduction

1.1 Background

Salsa, which was designed by Bernstein in April 2005 [4], is a stream cipher having a 256-bit security level against key recovery attacks. He submitted Salsa20, a 20-round Salsa, to the ECRYPT Stream Cipher Project, eSTREAM[1], as a candidate stream cipher for software applications with high throughput requirements and hardware applications with restricted resources. The eSTREAM portfolio was completed in September 2008; eventually, Salsa20/12, a 12-round Salsa20, was selected as a finalist for the eSTREAM software portfolio. ChaCha, which a variant of Salsa, was proposed by Bernstein in January 2008 [3] to provide better diffusion and higher resistance of cryptanalysis than Salsa. ChaCha has a 256-bit security level against key recovery attacks.

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

After releasing the Salsa and ChaCha algorithms, several studies reported the security evaluations for both ciphers [1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 19]. The most relevant of these is the differential attack based on the *probabilistic neutral bits* (PNB) concept, proposed by Aumasson et al. at FSE 2008 [1]. The PNB concept is to divide secret key bits into two sets: one of *significant key bits* and another of *nonsignificant key bits*, and a *neutral measure* is used as an evaluation indicator to discriminate them. The fewer elements in a set of significant key bits, the less the time complexity required for an adversary to recover an unknown secret key; thus, it is crucial to analyze the PNB concept in the differential attacks on Salsa and ChaCha. In fact, Aumasson et al. [1] first searched for the input/output differential pair having the best differential bias; then, based on the obtained input/output differential pair, they divided secret key bits into two sets using the PNB concept; finally, they performed a differential attack on ChaCha20/7, the 7-round version of ChaCha, with time and data complexities of 2^{248} and 2^{27}, respectively. Then, several researchers reported improvements of their proposed attack [2, 5, 6, 7, 16, 19]. 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 $2^{228.51}$ and $2^{80.51}$, respectively, proposed by Coutinho and Neto at EUROCRYPT 2021 [7].

As mentioned above, existing differential attacks on Salsa and ChaCha have focused on searching for the input/output differential pair with the best differential bias. However, the differential biases and PNB obtained from the existing attacks are not always the best combination. Time and data complexities for attacks can be estimated by their combination. Furthermore, a comprehensive analysis of the PNB was not performed in existing studies [1, 2, 5, 6, 7, 16, 19]; the upper bounds of the number of rounds required for the differential attack based on the PNB to be successful have not been clarified. This indicates that a comprehensive analysis of the PNB should have room for improvement of existing attacks.

1.2 Our Contributions

In this study, we propose a *PNB-based differential attack*, which first analyzes the PNB and then the differential bias. To summarize, the proposed attack works on a reduced-round ChaCha by first comprehensively analyzing the output differential ($\mathcal{OD}$) bit position with high neutral measures and then searching for the input differential ($\mathcal{ID}$) bit position with the best 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 a comprehensive analysis of PNB and clarify the upper bounds of the number of rounds required for the differential attack based on the PNB to be successful. Our contributions in this study can be summarized as follows.

- By comprehensively analyzing the PNB, we clarified the distribution of neutral measures for each round. Furthermore, we demonstrated that the value of the neutral measure varied significantly depending on the $\mathcal{OD}$ bit position. In particular, we reported that all 0th single-bits of each word in all intermediate rounds of the reduced-round ChaCha20 were $\mathcal{OD}$ bit positions having a high neutral measure. In fact, these $\mathcal{OD}$ bit positions were used for the proposed attack.

- Based on the comprehensive analysis of the PNB, we examined the value of neutral measures for each round of the inversed round function. Consequently, we speculated that the upper bound of the number of rounds required for the PNB-based differential attack to be successful is 7.25 rounds.

- Let $\Delta_i^{(r)}[j]$ be a 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, we reported the $\mathcal{ID}$-$\mathcal{OD}$ pairs

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}	[19]
	2^{130}	2^{25}	[5]
	$2^{127.5}$	$2^{27.5}$	[5]
	$2^{102.2}$	2^{56}	[6]
	$2^{77.4}$	2^{58}	[2]
ChaCha20/7	2^{248}	2^{27}	[1]
	$2^{246.5}$	2^{27}	[19]
	$2^{238.9}$	2^{96}	[16]
	$2^{237.7}$	2^{96}	[5]
	$2^{231.9}$	2^{50}	[6]
	$\mathbf{2^{231.63}}$	$\mathbf{2^{49.58}}$	**This work**
	$2^{230.86}$	$2^{48.8}$	[2]
	$2^{228.51}$	$2^{80.51}$	[7]
ChaCha20/7.25	$\mathbf{2^{255.62}}$	$\mathbf{2^{37.49}}$	**This work**
	$\mathbf{2^{244.22}}$	$\mathbf{2^{69.14}}$	**This work**

with a high differential bias to use for our 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 $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$. We believe that at least one of these $\mathcal{ID}$-$\mathcal{OD}$ pairs should yield the best combination of the differential bias and PNB.

- Based on the combination of the differential biases and PNB, we demonstrated a differential attack on ChaCha20/7 with time and data complexities of $2^{231.63}$ and $2^{49.58}$, respectively, using the $\mathcal{ID}$-$\mathcal{OD}$ pair of $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$. Furthermore, we present a differential attack on ChaCha20/7.25 with time and data complexities of $2^{255.62}$ and $2^{37.49}$, respectively, using the $\mathcal{ID}$-$\mathcal{OD}$ pair of $(\Delta_{15}^{(0)}[6], \Delta_0^{(3.5)}[0])$.

- For the existing best attack on ChaCha20/7, Coutinho and Neto [7] used $\Delta_5^{(3.5)}[0]$ $(= \Delta_5^{(4)}[7] \oplus \Delta_{10}^{(4)}[0])$ as the $\mathcal{OD}$. Because all 0th single-bits of each word in all intermediate rounds of the reduced-round ChaCha20 are the $\mathcal{OD}$ bit positions with a high neutral measure, we consider that it should have the possibility to achieve differential attacks on ChaCha20/7.25, ChaCha20/7.5. In fact, we demonstrate the differential attack on ChaCha20/7.25 with time and data complexities of $2^{244.22}$ and $2^{69.14}$, respectively, using $\mathcal{ID}$-$\mathcal{OD}$ pair presented by Coutinho and Neto [7].

Table 1 summarizes the existing attacks and our attack on the reduced-round ChaCha[2]. As shown in the table, our attack could not reach the improvement of the best existing attack on ChaCha20/7.

[2]According to [7, Table 1], Coutinho and Neto presented two differential attacks on ChaCha20/7 with time complexities of 2^{218} and 2^{224} and data complexities of 2^{218} and 2^{224}, respectively. These seemed similar to the best attacks on ChaCha20/7; however, the verification is beyond the scope of this study because they are distinguishing attacks, and not key recovery attacks.

However, no study that focused on the attack on ChaCha20/7.25 has been conducted; thus, our attack is the best differential attack on the reduced-round ChaCha20, particularly on ChaCha20/7.25.

In the conventional attacks on ChaCha20, if a time complexity for the attack is beyond the exhaustive search for an unknown secret key, cryptanalysts select an approach that reduces the number of target rounds for the attack or changes an $\mathcal{ID}$-$\mathcal{OD}$ pair with a better forward bias. Furthermore, we focused on the fact that the PNB concept has a strong influence on the theoretical time complexity. Consequently, we revealed that even if the number of target rounds for the attack is increased, it may be possible to suppress the increase in the theoretical time complexity. We believe that this study will be the first step toward an attack on more rounds of ChaCha20, e.g., ChaCha20/8.

1.3 Organization of This Paper

The rest of this paper is organized as follows. In Section 2, we briefly describe the specification of the ChaCha stream cipher. In Section 3, we review generic techniques for the existing differential attack based on the PNB concept. In Section 4, we present experimental results associated with the detailed analysis of the PNB and discuss certain properties. In Section 5, we examine the differential bias at the output differential bit position obtained in Section 4 and then perform the differential attack on ChaCha20/7, ChaCha20/7.25, and ChaCha20/7.5. Finally, Section 6 concludes this study.

2 Specification of ChaCha

ChaCha [3,18] comprises the following three steps to generate a keystream block of 16 words, where each word size is 32 bits:

Step 1. The initial state matrix $X^{(0)}$ of order 4×4 is initialized from a 256-bit secret key $k = (k_0, k_1, \ldots, k_7)$, 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 $c_0 = \text{0x}61707865$, $c_1 = \text{0x}3320646e$, $c_2 = \text{0x}79622d32$, and $c_3 = \text{0x}6b206574$. After initialization, we obtained the following initial state matrix:

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

Step 2. The round function of ChaCha comprises four simultaneous computations of the so-called $\mathsf{quarterround}$ function. The overall structure of the $\mathsf{quarterround}$ function is shown in Fig. 1. As per 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:

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

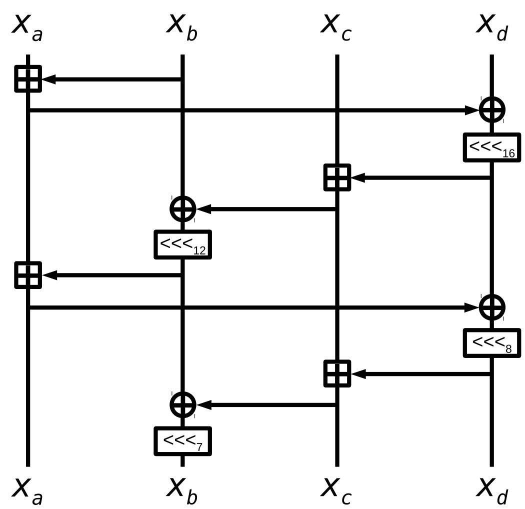

Figure 1: Overall structure of the quarterround function.

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, called ChaCha20, has $R = 20$ rounds, and the reduced-round version of ChaCha20 is denoted as ChaCha20/R.

The round function of ChaCha is reversible, i.e., a 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.

Note that the quarterround function can then be subdivided into four rounds: 0.25, 0.5, 0.75, and 1 round. In the following, the 0.25-round quarterround function comprises one wordwise modular addition, one bitwise XOR, and one bitwise left rotation.

3 Differential Cryptanalysis of ChaCha

The most relevant study on the security analysis of Salsa and ChaCha was presented by Aumasson et al. at FSE 2008 [1]. They proposed a differential attack based on the *probabilistic neutral bits* (PNB) concept and applied it to reduced versions of Salsa and ChaCha. Then, several researchers reported improvements of their proposed attack [2, 5, 6, 7, 10, 14, 16, 19], and it is now possible to attack up to 7 rounds of ChaCha, i.e., ChaCha20/7.

In this section, we review generic techniques for the differential attack based on the PNB concept. This attack comprises the precomputation and online phases. In the precomputation phase, we examine single-bit differential biases and PNB as well as execute a probabilistic backward computation (PBC). Subsequently, we execute the online phase to recover an unknown key.

3.1 Precomputation Phase

3.1.1 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 $0 \le i \le 15$ and $0 \le j \le 31$, and $x_i'^{(r)}[j]$ be an associated bit with the difference $\Delta_i^{(r)}[j] = x_i^{(r)}[j] \oplus x_i'^{(r)}[j]$. Based on a difference $\Delta_i^{(0)}[j] = 1$ 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 $\Delta_p^{(r)}[q] = x_p^{(r)}[q] \oplus x_p'^{(r)}[q]$ 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 by

$$\Pr\big(\Delta_p^{(r)}[q] = 1 \mid \Delta_i^{(0)}[j] = 1\big) = \frac{1}{2}(1 + \epsilon_d), \tag{1}$$

where ϵ_d denotes the $\mathcal{OD}$ bias.

To distinguish between the $\mathcal{OD}$ obtained from true random number sequences and the $\mathcal{OD}$ obtained from the r-round internal state matrices in ChaCha, we use the following theorem proved by Mantin and Shamir at FSE 2001 [17].

Theorem 1 ([17, Theorem 2])**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be two distributions, and suppose that the target event occurs in $\mathcal{X}$ with a probability p and $\mathcal{Y}$ with a probability $p \cdot (1 + q)$. Then, for small p and q, $\mathcal{O}(\frac{1}{p \cdot q^2})$ samples suffice to distinguish $\mathcal{X}$ from $\mathcal{Y}$ with a constant probability of success.*

Let $\mathcal{X}$ be a distribution of the $\mathcal{OD}$ of true random number sequences and $\mathcal{Y}$ be a distribution of the $\mathcal{OD}$ obtained from the r-round internal state matrices in ChaCha. As per Theorem 1 and Eq. (1), the target event occurs in $\mathcal{X}$ and $\mathcal{Y}$ with probabilities $\frac{1}{2}$ and $\frac{1}{2} \cdot (1 + \epsilon_d)$, respectively; thus, the number of samples to distinguish $\mathcal{X}$ and $\mathcal{Y}$ is $\mathcal{O}(\frac{2}{\epsilon_d^2})$, as p and q are equal to $\frac{1}{2}$ and ϵ_d, respectively.

3.1.2 PNB

The PNB divides secret key bits in the sets of m-bit significant and n-bit nonsignificant key bits. To differentiate between the sets, Aumasson et al. focused on the degree of influence of each secret key bit on the $\mathcal{OD}$, and the degree of influence, the *neutral measure*, was 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 nonsignificant.
- $\gamma_i = 0$: $\mathcal{OD}$ 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 in two sets, the m-bit significant and n-bit nonsignificant 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 $\Delta_i^{(0)}[j] = 1$; derive the keystream blocks $Z = X^{(0)} + X^{(R)}$ and $Z' = X'^{(0)} + X'^{(R)}$, respectively.

Step 2. Prepare the 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 inversed round function of ChaCha.

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

Step 5. Repeatedly perform Steps 1-4 using different initial state matrices with the same $\Delta_i^{(0)}[j] = 1$; 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 a set of m-bit significant key bits and those with $\gamma_\kappa \geq \gamma$ into a set of n-bit nonsignificant key bits.

3.1.3 PBC

As explained at the beginning of this subsection, we obtained r-round single-bit differential biases from the initial state matrices with the selected $\mathcal{ID}$, indicating that these biases are obtained by performing the forward computation in the target cipher. Moreover, we could 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 $\Delta_i^{(0)}[j] = 1$; 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 nonsignificant 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 inversed round function of ChaCha.

Step 4. Compute $\hat{\Gamma}_p^{(r)}[q] = \hat{y}_p^{(r)}[q] \oplus \hat{y}_p'^{(r)}[q]$ for all possible choices of p and q, 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.

As per [1], the bias ϵ was approximated as $\epsilon_d \cdot \epsilon_a$ and was considered for computing 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, we 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., the m-bit significant key bits, the following 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, we perform an additional exhaustive search over the n-bit nonsignificant key bits to confirm the correctness of the filtered subkey and identify the n-bit nonsignificant key bits.
- **Step 2-3.** Stop if the correct key is reported and output the recovered key.

3.2.1 Complexity Estimation

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

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

for a probability of nondetection $P_{nd} = 1.3 \times 10^{-3}$. In practice, α (and hence N) is selected to minimize the time complexity of the attack.

4 Analysis of PNB

4.1 Searching for the PNB with High Neutral Measures

Typically, differential attacks on Salsa and ChaCha determine the $\mathcal{ID}$-$\mathcal{OD}$ pair with high differential biases in the beginning, then focus on the $\mathcal{OD}$ bit position, and explore its neutral measures. Expressed differently, certain studies [1, 2, 5, 6, 7, 16, 19] focused on analyzing the differential bias and optimized a combination of the differential bias and PNB as time and data complexities for the attack can be evaluated by their combination. Furthermore, optimizing the combination by focusing on the PNB analysis may be effective for improving the differential attack on ChaCha.

In this section, we focus on a comprehensive analysis of the PNB and examine the conditions that induce high neutral measures because the size of PNB directly influences the time complexity of an attack, as shown in Section 3.2.1. No study focusing on comprehensively analyzing the PNB has been conducted. If conditions that induce high neutral measures can be clarified, we can claim that the existing attacks may require improvement.

We perform the following procedure to comprehensively search for the PNB with high neutral measures:

Step 1. We generate a secret key $k = (k_0, \ldots, k_7)$ uniformly at random.

Step 2. We select the $\mathcal{ID}$ bit position $\Delta_i^{(0)}[j]$, nonce, and uniformly block counter at random. Then, we 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 $(X^{(0)}, X'^{(0)})$, we 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 our attack on ChaCha20/R.

Step 4. From the r-round internal state matrix pair $(X^{(r)}, X'^{(r)})$, we 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 $(X^{(R)}, X'^{(R)})$, we obtain keystream blocks $Z = X^{(0)} + X^{(R)}$ and $Z' = X'^{(0)} + X'^{(R)}$.

Step 6. We complement a particular key bit position κ ($\kappa \in \{0, \ldots, 255\}$) to yield the states $\overline{X}^{(0)}$ and $\overline{X'}^{(0)}$. Then, we 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 inversed round function of ChaCha as well as derive $\Gamma_p^{(r)}[q] = Y_p^{(r)}[q] \oplus Y_p'^{(r)}[q]$ for all possible choices of p and q.

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

Step 8. We repeat Steps 2-7.

After completing our trials with the above steps, we compute the neutral measures γ_κ for each counter.

4.2 Experimental Results

This subsection shows the experimental results based on the PNB searching procedure described in Section 4.1. To search for the PNB with high neutral measures, we conducted experiments with 2^8 trials using 2^{28} $\mathcal{ID}$s (samples) for each key. Based on Theorem 1, let $\mathcal{X}$ be a distribution of $\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q]$ obtained from the r-round internal state matrices in a true random number generator and $\mathcal{Y}$ be a distribution of $\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q]$ obtained from the r-round internal state matrices in ChaCha20/R. The target event occurs in $\mathcal{X}$ and $\mathcal{Y}$ with probabilities $\frac{1}{2}$ and $\frac{1}{2} \cdot (1 + \gamma_\kappa)$, respectively; thus, the number of samples to distinguish $\mathcal{X}$ and $\mathcal{Y}$ is $\mathcal{O}(\frac{2}{\gamma_\kappa^2})$. Our results were reliable when the derived neutral measures γ_κ were greater than $2^{-13.5}$ (≈ 0.000086), as 2^{28} samples were used.

4.2.1 Experimental Results of ChaCha20/7

Fig. 2 shows the average neutral measures $\hat{\gamma}_\kappa$ for each $\mathcal{OD}$ bit position in ChaCha20/7. In this figure, the vertical axis represents the average value of the neutral measures 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 the left). The blue

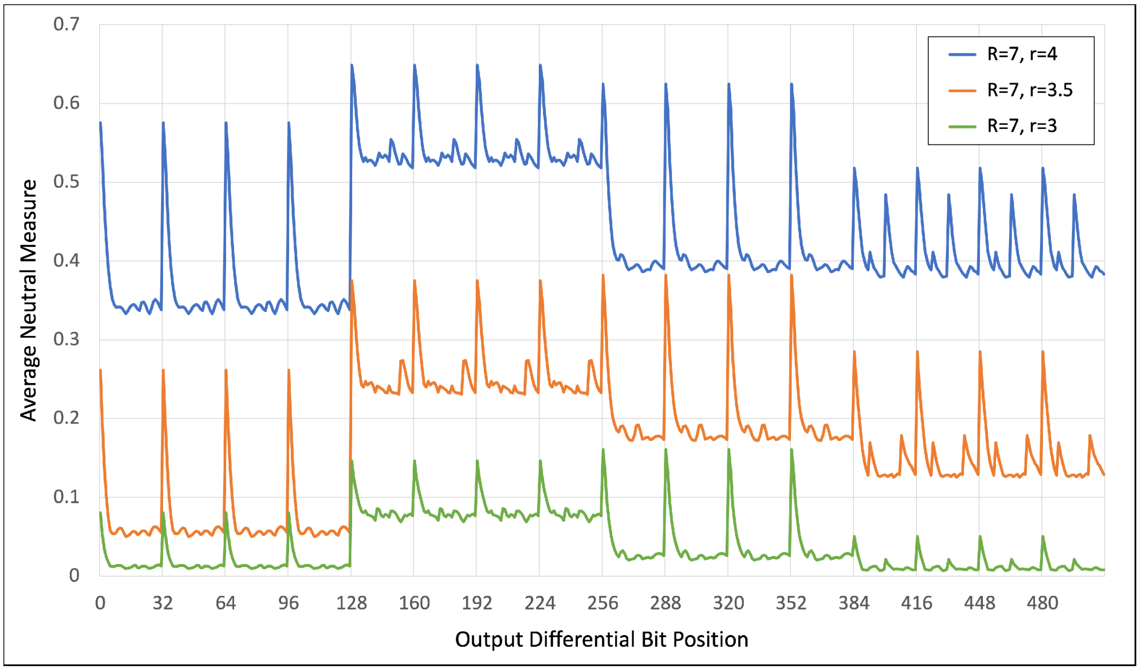

Figure 2: Average neutral measures $\hat{\gamma}_\kappa$ for each $\mathcal{OD}$ bit position when the number of intermediate rounds r is 3, 3.5, and 4 in ChaCha20/7.

(top), orange (center), and green (bottom) lines show the average value of the neutral measures when the number of intermediate rounds r is 3, 3.5, and 4, respectively.

From this figure, the average neutral measures $\hat{\gamma}_\kappa$ tends to be higher at all 0th $\mathcal{OD}$ bit position of each word, regardless of the number of intermediate rounds. Expressed differently, optimizing a combination of the differential bias and PNB by focusing on all 0th $\mathcal{OD}$ bit positions may be effective for improving the differential attack on ChaCha20/7. Focusing on the existing studies [1, 16, 19], the 0th $\mathcal{OD}$ bit positions with a high average neutral measure were selected in the third round, i.e., $\Delta^{(3)}_{11}[0]$; thus, it is difficult to improve the differential attack on ChaCha20/7, even if we focus on when the number of intermediate rounds r is 3. It is difficult if the number of intermediate rounds r is ¡3 because the less the number of the intermediate rounds r, the lower the average neutral measures. Therefore, we should attempt to improve the differential attack on ChaCha20/7 by focusing on when the number of intermediate rounds r is ¿3, e.g., 3.5 or 4 rounds.

The comprehensive analysis of the PNB in this section cannot be directly compared with those in existing studies, e.g., [2,5,6,7,10] because a multi-bit differential or a differential-linear technique was employed in the existing studies, whereas we only focus on the single-bit differential technique. From a computational complexity perspective, we searched for the PNB with high neutral measures for only a single-bit $\mathcal{OD}$ bit position. Similarly, we should search for the PNB with high neutral measures for multi-bit $\mathcal{OD}$ bit positions, which is left as future work.

4.2.2 Experimental Results of ChaCha20/7.25, ChaCha20/7.5, and ChaCha20/7.75

In this study, we performed the differential attack on not only ChaCha20/7 but also ChaCha20/7.25, ChaCha20/7.5, and ChaCha20/7.75. Thus, we searched for the PNB with high neutral measures for the target rounds. Fig. 3 shows the average neutral measures $\hat{\gamma}_\kappa$ 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

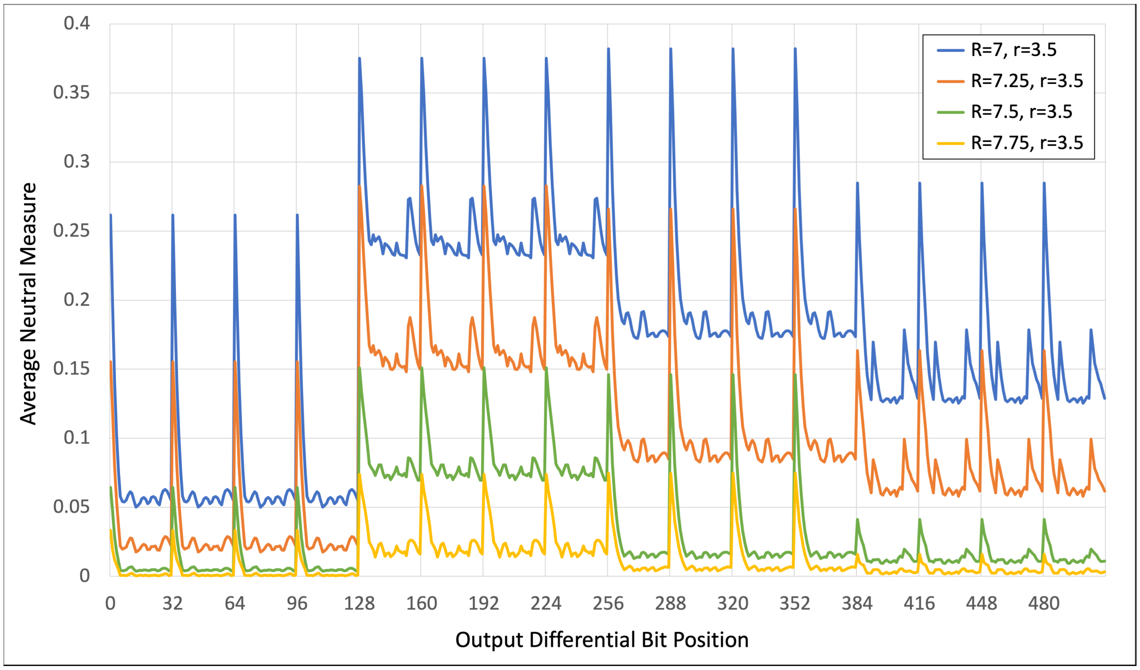

Figure 3: Average neutral measures $\hat{\gamma}_\kappa$ for each $\mathcal{OD}$ bit position when the number of intermediate rounds r is 3.5 and number of target rounds R is 7, 7.25, 7.5, and 7.75.

and horizontal axes and the auxiliary lines on the vertical axis are the same as in Fig. 2. The blue (top), orange (the second from the top), green (the second from the bottom), and yellow (bottom) lines show the average value of the neutral measures 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.

Similar to the experimental results of ChaCha20/7, from this figure, the average neutral measures $\hat{\gamma}_\kappa$ tended to be higher at all 0th $\mathcal{OD}$ bit positions of each word. Therefore, optimizing a combination of the differential bias and PNB by focusing on all 0th $\mathcal{OD}$ bit positions may be effective for performing a differential attack on ChaCha20/7.25, ChaCha20/7.5, and ChaCha20/7.75.

4.3 Discussions

In this subsection, based on experimental results described in Section 4.2, we discuss the PNB from the following two aspects.

- Relationships between the PNB and inversed round function.
- Upper bounds of the number of rounds for analyzing PNB.

4.3.1 Relationships between PNB and Inversed Round Function

We discuss relationships between the PNB (or the average neutral measure) and inversed round function of ChaCha. To this end, we investigated relationships between the input word position to the inversed `quarterround` function and the cumulative number of wordwise modular subtractions, which was 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 was counted as follows:

Table 2: Relationships between the input word position to the inversed 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: Relationships between the input word position to the inversed 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

Wordwise modular subtraction. The cumulative number of wordwise modular subtractions is counted only when wordwise modular subtraction is executed. Moreover, we calculated the sum of the cumulative numbers of wordwise modular subtractions in two input words to the wordwise modular subtraction. For example, when wordwise modular subtraction, $A' = A - B$, was executed and the cumulative numbers of wordwise modular subtractions in the two input words A and B were 70 and 85, respectively, we could obtain 156 as the cumulative number of wordwise modular subtractions in the output word A'.

Bitwise XOR. We calculated only the sum of the cumulative numbers of wordwise modular subtractions in two input words to bitwise XOR. For example, when bitwise XOR, $B' = B \oplus C$, was executed and the cumulative numbers of wordwise modular subtractions in the two input words B and C were 37 and 48, respectively, we could obtain 85 as 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 operation of bitwise left rotation.

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

From these tables, the cumulative number of wordwise modular subtractions differed depending on the input word position relative to the inversed round function and number of intermediate rounds r. In particular, the cumulative number of wordwise modular subtractions was smaller in the order of the input word positions B, C, D, and A when the number of intermediate rounds r was

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

R	Maximum			Minimum			Average	Median
	γ_κ	p	q	γ_κ	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

3, 3.5, 4, and 4.5, whereas the cumulative number of wordwise modular subtractions was smaller in the order of the input word positions D, A, B, and C when the number of intermediate rounds r was 3.25, 3.75, 4.25, and 4.75. We now compare the experimental results shown in Fig. 3 with the investigation results when $r = 3.5$, shown in Tables 2 and 3. Note that the range of input word positions A, B, C, and D corresponded to the output difference bit positions 0 to 127, 128 to 255, 256 to 383, and 384 to 511, respectively. From Fig. 3, the value of the average neutral measure was higher in the order of the input word positions B, C, D, and A when the number of intermediate rounds r was 3.5 (all 0th bit positions are exceptions); thus, the smaller the cumulative number of wordwise modular subtractions, the higher the value of the average neutral measure. After the 0th bit position is uninfluenced by the carry-in wordwise modular subtraction, i.e., it is uninfluenced by the input/output difference, we speculated that it is a special case.

In summary, the value of neutral measures depends on the input word position relative to the inversed round function and is influenced by the cumulative number of wordwise modular subtractions. To summarize, the conditions that induce a high neutral measure depend on the $\mathcal{OD}$ bit position, particularly all 0th $\mathcal{OD}$ bit positions.

4.3.2 Upper Bounds of the Number of Rounds for Analyzing PNB

We discuss the upper bounds of the number of rounds required for the PNB-based differential attack to be successful. To this end, we investigated the value of neutral measures for each round of the inversed round function. Table 4 shows the maximum, minimum, average, and median values of neutral measures γ_κ for each target round R when the number of the intermediate rounds r is 3.5[3]. These findings can be obtained by a detailed analysis of the experimental results described in Section 4.2. The R column in these tables shows the number of target rounds for our attack, and we can compute the number of rounds of the inversed round function as $R - r$.

Our experimental results were reliable when the derived neutral measures γ_κ were greater than $2^{-13.5}$ (≈ 0.000086), as 2^{28} samples were used. From Table 4, all values of neutral measures were reliable when the number of target rounds R was 7, 7.25, 7.5, and 7.75; thus, the upper bounds of the number of rounds required for the PNB-based differential attack to be successful could be at least 7.75 rounds. However, given that the threshold γ used in the existing attacks, such as [2,5,7], was $\gamma = 0.27$ or 0.35, it was practically difficult to perform the differential attack when the number of target rounds R was 7.5 or 7.75; thus, we speculated that the upper bound of the number of rounds required for the PNB-based differential attack to be successful was 7.25 rounds. To verify our speculation, we performed the PNB-based differential attack on the reduced-round ChaCha

[3]In the existing best attack on ChaCha20/7, Coutinho and Neto [7] used $\Delta_5^{(3.5)}[0]$ ($= \Delta_5^{(4)}[7] \oplus \Delta_{10}^{(4)}[0]$) as $\mathcal{OD}$. Accordingly, we focused solely on when $r = 3.5$.

Table 5: Best single-bit differential biases $|\epsilon_d|$ at the 0th $\mathcal{OD}$ bit positions of each word for 3.5 rounds of ChaCha. Our experiments were conducted with 2^6 trials using 2^{28} $\mathcal{ID}$s for each key; thus, our experimental results were reliable when the derived differential biases $|\epsilon_d|$ were greater than $2^{-13.5}$ (≈ 0.000086), as 2^{28} samples were used.

$\mathcal{ID}$	$\mathcal{OD}$	$\|\epsilon_d\|$
$\Delta_{15}^{(0)}[6]$	$\Delta_0^{(3.5)}[0]$	0.000506
$\Delta_{12}^{(0)}[6]$	$\Delta_1^{(3.5)}[0]$	0.000468
$\Delta_{13}^{(0)}[6]$	$\Delta_2^{(3.5)}[0]$	0.000482
$\Delta_{14}^{(0)}[6]$	$\Delta_3^{(3.5)}[0]$	0.000430
$\Delta_{14}^{(0)}[23]$	$\Delta_{12}^{(3.5)}[0]$	0.000023
$\Delta_{13}^{(0)}[19]$	$\Delta_{13}^{(3.5)}[0]$	0.000023
$\Delta_{15}^{(0)}[12]$	$\Delta_{14}^{(3.5)}[0]$	0.000024
$\Delta_{12}^{(0)}[27]$	$\Delta_{15}^{(3.5)}[0]$	0.000028

with target rounds of 7, 7.25, and 7.5.

5 PNB-based Differential Attack

In this section, we describe a PNB-based differential attack on the reduced-round ChaCha. First, we clarified the $\mathcal{OD}$ bit position with high neutral measures, i.e., the 0th $\mathcal{OD}$ bit positions of each word, from the PNB analysis described in Section 4. Then, we analyzed the differential biases at the target $\mathcal{OD}$ bit positions in detail and obtained the $\mathcal{ID}$ bit position with the best differential bias at the target $\mathcal{OD}$ bit positions. Finally, we estimated 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 Section 4, we comprehensively analyzed the $\mathcal{OD}$ bit positions with high neutral measures. Accordingly, by analyzing the $\mathcal{ID}$ bit position with the best differential bias at the target $\mathcal{OD}$ bit positions, we decided the $\mathcal{ID}$-$\mathcal{OD}$ pair to use for our attack.

To identify the $\mathcal{ID}$ bit position with the best differential bias $|\epsilon_d|$ at the target $\mathcal{OD}$ bit positions, we conducted experiments with 2^6 trials using 2^{28} $\mathcal{ID}$s for each key; thus, the results were reliable when the derived differential biases $|\epsilon_d|$ were greater than $2^{-13.5}$ (≈ 0.000086), as 2^{28} samples were used. Table 5 lists the best differential biases $|\epsilon_d|$ at the target $\mathcal{OD}$ bit positions such as $\Delta_0^{(3.5)}[0]$, $\Delta_1^{(3.5)}[0]$, $\Delta_2^{(3.5)}[0]$, $\Delta_3^{(3.5)}[0]$, $\Delta_{12}^{(3.5)}[0]$, $\Delta_{13}^{(3.5)}[0]$, $\Delta_{14}^{(3.5)}[0]$, and $\Delta_{15}^{(3.5)}[0]$. As shown in this table, we could obtain the reliable results at $\Delta_0^{(3.5)}[0]$, $\Delta_1^{(3.5)}[0]$, $\Delta_2^{(3.5)}[0]$, and $\Delta_3^{(3.5)}[0]$, but not at $\Delta_{12}^{(3.5)}[0]$, $\Delta_{13}^{(3.5)}[0]$, $\Delta_{14}^{(3.5)}[0]$, and $\Delta_{15}^{(3.5)}[0]$. Moreover, these led to unreliable results at other 0th $\mathcal{OD}$ bit positions, such as $\Delta_4^{(3.5)}[0]$, $\Delta_5^{(3.5)}[0]$, $\Delta_6^{(3.5)}[0]$, $\Delta_7^{(3.5)}[0]$, $\Delta_8^{(3.5)}[0]$, $\Delta_9^{(3.5)}[0]$, $\Delta_{10}^{(3.5)}[0]$, and $\Delta_{11}^{(3.5)}[0]$, which was because the results were affected by the unreliable results at $\Delta_{12}^{(3.5)}[0]$, $\Delta_{13}^{(3.5)}[0]$, $\Delta_{14}^{(3.5)}[0]$, and $\Delta_{15}^{(3.5)}[0]$, according to the computations of the quarterround function (see Section 2 for details). Consequently, we decided the $\mathcal{ID}$-$\mathcal{OD}$ pairs to use for our attack: $(\Delta_{15}^{(0)}[6], \Delta_0^{(3.5)}[0])$, $(\Delta_{12}^{(0)}[6], \Delta_1^{(3.5)}[0])$, $(\Delta_{13}^{(0)}[6], \Delta_2^{(3.5)}[0])$, and $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$.

Table 6: Best single-bit differential biases $|\epsilon_d|$ at the 0th $\mathcal{OD}$ bit positions of each word for 3.5 rounds of ChaCha. Experiments were conducted with 2^8 trials using 2^{34} $\mathcal{ID}$s for each key; thus, the results were reliable when the derived differential biases $|\epsilon_d|$ were greater than $2^{-16.5}$ (≈ 0.000011), as 2^{34} samples were 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

To obtain additional precise single-bit differential biases for the decided $\mathcal{ID}$-$\mathcal{OD}$ pairs, we conducted additional experiments with 2^8 trials using 2^{34} $\mathcal{ID}$s for each key; thus, the results were reliable when the derived differential biases $|\epsilon_d|$ were greater than $2^{-16.5}$ (≈ 0.000011), as 2^{34} samples were used. Table 6 lists the 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 shown in this table, we could obtain reliable results at the target positions; then, we used the listed biases $|\epsilon_d|$ to estimate time and data complexities for our attack.

5.2 Complexity Estimation

To estimate time and data complexities for the PNB-based 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 Section 3 for details):

Step 1. We recalculate neutral measures corresponding to the decided $\mathcal{ID}$-$\mathcal{OD}$ pairs and divide the secret key bits in two sets: m-bit significant and n-bit nonsignificant key bits.

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

Step 3. We perform the online phase and estimate time and data complexities to recover the unknown key, as described in Eq.(2).

To perform the abovementioned steps, we conducted experiments with 2^8 trials using 2^{30} $\mathcal{ID}$s for each key; thus, the results were reliable when the derived biases $|\epsilon_a|$ were greater than $2^{-14.5}$ (≈ 0.000043), as 2^{30} samples were used.

5.2.1 Complexity Estimation for ChaCha20/7

Table 7 shows the best parameters for each target $\mathcal{ID}$-$\mathcal{OD}$ pair to estimate time and data complexities for our attack on ChaCha20/7. The threshold γ was in total 18 patterns, from 0.10 to 0.95 at an interval of 0.05, n represented the number of nonsignificant key bits, $|\epsilon_d|$ was derived from Table 6, $|\epsilon_a|$ was obtained by performing PBC for each threshold γ, and α was selected to minimize the time complexity of our attack.

Consequently, we could perform our attack on ChaCha20/7 with time and data complexities of $2^{231.63}$ and $2^{49.58}$, respectively, using the best parameters, such that $\mathcal{ID}$-$\mathcal{OD}$ pair was $(\Delta_{14}^{(0)}[6], \Delta_3^{(3.5)}[0])$, γ was 0.35, n was 74, α was 29, and the list of PNB was {6, 7, 8, 9, 10,

Table 7: Best parameters for our attack on ChaCha20/7.

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

Table 8: Best parameters for our attack on ChaCha20/7.25.

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

11, 12, 13, 14, 19, 27, 28, 29, 30, 31, 34, 35, 36, 37, 46, 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}.

5.2.2 Complexity Estimation for ChaCha20/7.25 and ChaCha20/7.5

Similar to the complexity estimation for ChaCha20/7, we show the best parameters for each target $\mathcal{ID}$-$\mathcal{OD}$ pair to estimate time and data complexities for our attack on ChaCha20/7.25 and ChaCha20/7.5 in Tables 8 and 9, respectively.

As shown in Table 8, we could perform our attack on ChaCha20/7.25 with time and data complexities of $2^{255.62}$ and $2^{48.36}$, respectively, using the best parameters, such that $\mathcal{ID}$-$\mathcal{OD}$ pair was $(\Delta_{15}^{(0)}[6], \Delta_{0}^{(3.5)}[0])$, γ was 0.30, n was 49, α was 3, and the list of PNB was {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}. ChaCha20 provides a 256-bit security level against key recovery attacks; thus, our attack on ChaCha20/7.25 is more efficient than the exhaustive search for an unknown secret key.

Moreover, as shown in Table 9, we performed our attack on ChaCha20/7.5 with time and data complexities of $2^{273.49}$ and $2^{37.49}$, respectively, using the best parameters, such that $\mathcal{ID}$-$\mathcal{OD}$ pair was $(\Delta_{15}^{(0)}[6], \Delta_{0}^{(3.5)}[0])$, γ was 0.30, n wass 20, α was 1, and the list of PNB was {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 was inefficient because this is beyond the security level of ChaCha20.

5.3 Discussions

5.3.1 Related Works

As described in Section 3, Aumasson et al. [1] proposed a framework of the differential attack based on the PNB concept and applied it to the reduced-round Salsa and ChaCha. They first obtained an

Table 9: Best parameters for our 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	$2^{273.67}$	$2^{37.67}$

$\mathcal{ID}$-$\mathcal{OD}$ pair, $(\Delta_{13}^{(0)}[13], \Delta_{11}^{(3)}[0])$, with a high differential bias using a single-bit differential technique. Then, they observed the PNB at the target $\mathcal{OD}$ bit position and finally estimated 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. [19] proposed new techniques, called a column chaining distinguisher (CCD) and a 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 4-step CCD, observed the PNV at the target $\mathcal{OD}$ bit position, and finally estimated time and data complexities as well as a 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 around 0.43.

Maitra [16] further improved Aumasson et al.'s attack to use a chosen-IV technique. He 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 select IVs corresponding to the secret keys properly, given the target $\mathcal{ID}$, $\Delta_{13}^{(0)}[13]$. His attack can be performed on ChaCha20/7 with time and data complexities of $2^{238.94}$ and $2^{23.89}$, respectively.

Choudhuri and Maitra [5] 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 $\mathcal{OD}$ was $\Delta_0^{(4.5)}[0] \oplus \Delta_0^{(4.5)}[8] \oplus \Delta_1^{(4.5)}[0] \oplus \Delta_5^{(4.5)}[12] \oplus \Delta_{11}^{(4.5)}[0] \oplus \Delta_9^{(4.5)}[0] \oplus \Delta_{15}^{(4.5)}[0] \oplus \Delta_{12}^{(4.5)}[16] \oplus \Delta_{12}^{(4.5)}[24]$. Using such multi-bit differentials, they presented the attack on ChaCha20/7 with time and data complexities of $2^{237.65}$ and $2^{31.6}$, respectively.

Beierle et al. [2] presented a generic framework of differential-linear attacks with a special focus on ARX ciphers, applied it to ChaCha20/7, and then improved upon the best existing attacks. Their attack can be performed on ChaCha20/7 with time and data complexities of $2^{230.86}$ and $2^{48.83}$, respectively.

Coutinho and Neto [7] developed Beierle et al.'s differential-linear attack and found a new $\mathcal{ID}$-$\mathcal{OD}$ pair, $(\Delta_{15}^{(0)}[6], \Delta_5^{(4)}[7] \oplus \Delta_{10}^{(4)}[0])$, to improve upon the best existing attacks. Their attack can be performed on ChaCha20/7 with time and data complexities of $2^{228.51}$ and $2^{80.51}$, respectively. To the best of our knowledge, this is the best attack on the reduced-round version of ChaCha, i.e., ChaCha20/7.

As summarized above, the best existing attack on the reduced-round ChaCha works on up to 7 rounds with time and data complexities of $2^{228.51}$ and $2^{80.51}$, respectively, although our attack had time and data complexities of $2^{231.63}$ and $2^{49.58}$, respectively; thus, our attack could not reach the improvement of the best existing attack on ChaCha20/7. As mentioned in Section 4, in this study, we solely focused on the single-bit differential technique; therefore, it might be possible to improve the best existing attack on ChaCha20/7 by focusing on the multi-bit differential or differential-linear technique, which is left as future work.

5.3.2 Further Improvement for the Attack on ChaCha20/7.25

Focusing on the best existing attack on ChaCha20/7, Coutinho and Neto [7] prepared 8 NVIDIA GPUs (RTX 2080ti) as their experimental environment and performed experiments with 2^{50} samples to comprehensively search for forward biases; thus, their experimental results were reliable when the derived differential biases $|\epsilon_d|$ were greater than $2^{-24.5}$ (≈ 0.000000042), as 2^{50} samples were used. Consequently, they reported the $\mathcal{OD}$ to use for their key recovery attack, such as $\Delta_5^{(3.5)}[0]$ ($= \Delta_5^{(4)}[7] \oplus \Delta_{10}^{(4)}[0]$), with the forward bias of $|\epsilon_d| = 0.0000002489$ (> 0.000000042); thus, the experimental results were reliable. Moreover, the following is our experimental environment: five Linux machines with 40-core Intel(R) Xeon(R) CPU E5-2660 v3 (2.60GHz), 128.0 GB of main memory, a gcc 7.2.0 compiler, and the C programming language. As described in Section 5.1, we obtained the unreliable forward biases for $\Delta_5^{(3.5)}[0]$ in our experimental environment. Therefore, we consider that the difference between Coutinho et al.'s and our attack significantly depend on the experimental environment.

Moreover, as discussed in Section 4.3, we speculated that the upper bounds of the number of rounds required for the PNB-based differential attack to be successful were 7.25 rounds, but no study focusing on the attack on ChaCha20/7.25 has been conducted. Regarding the security evaluations of symmetric-key ciphers, it is crucial to thoroughly analyze while gradually increasing the nonlinear operations such as S-boxes and modular additions. Expressed differently, we consider that it is meaningful to thoroughly analyze the security of the reduced-round ChaCha20 for each 0.25 round after the 0.25 quarter-round function in ChaCha20 includes one wordwise modular addition as the nonlinear operation. In summary, we improved on the best existing attack on the reduced-round ChaCha20, i.e., ChaCha20/7.25.

In conventional attacks on ChaCha20, if a time complexity for an attack is beyond the exhaustive search for an unknown secret key, cryptanalysts select an approach that reduces the number of target rounds for the attack or changes an $\mathcal{ID}$-$\mathcal{OD}$ pair with a better forward bias. Moreover, in this study, we focused on the fact that the PNB concept has a strong influence on the theoretical time complexity. Consequently, this study revealed that even if the number of target rounds for an attack is increased, it may be possible to suppress the increase in the theoretical time complexity.

In the best existing attack on ChaCha20/7, Coutinho and Neto [7] used $\Delta_5^{(3.5)}[0]$ ($= \Delta_5^{(4)}[7] \oplus \Delta_{10}^{(4)}[0]$) as the $\mathcal{OD}$. Because all 0th single-bits of each word in all intermediate rounds of the reduced-round ChaCha20 are the $\mathcal{OD}$ bit positions with a high neutral measure, we consider that it should have the possibility to achieve the differential attacks on ChaCha20/7.25, ChaCha20/7.5, and more. In fact, using $\mathcal{ID}$-$\mathcal{OD}$ pair presented by Coutinho and Neto [7], we performed experiments in the same procedure as described in Section 5.2 to estimate time and data complexities for the attack on ChaCha20/7.25 and ChaCha20/7.5. Consequently, we could perform the differential attack on ChaCha20/7.25 with time and data complexities of $2^{244.22}$ and $2^{69.14}$, respectively, using the best parameters such that $\mathcal{ID}$-$\mathcal{OD}$ pair was $(\Delta_{15}^{(0)}[6], \Delta_5^{(3.5)}[0])$, $|\epsilon_d| = 0.0000002489$, $|\epsilon_a| = 0.001215$, γ was 0.35, n was 81, α was 16, and the list of PNB was {2, 10, 11, 12, 19, 20, 26, 27, 28, 29, 30, 31, 32, 39, 40, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 66, 128, 129, 130, 135, 136, 143, 144, 147, 148, 149, 150, 151, 155, 156, 157, 158, 159, 160, 161, 162, 163, 168, 169, 170, 173, 174, 175, 176, 179, 180, 181, 182, 185, 186, 191, 199, 200, 201, 219, 220, 221, 222, 223, 232, 255}. Because our attack on ChaCha20/7.25 described in Section 5.2 had time and data complexities of $2^{255.62}$ and $2^{48.36}$, respectively, we demonstrated a further improvement for the attack on ChaCha20/7.25. Moreover, we performed the differential attack on ChaCha20/7.5 with time and data complexities of $2^{274.01}$ and $2^{64.01}$, respectively, using the best parameters, such that $\mathcal{ID}$-$\mathcal{OD}$ pair was $(\Delta_{15}^{(0)}[6], \Delta_5^{(3.5)}[0])$, $|\epsilon_d| = 0.0000002489$, $|\epsilon_a| = 0.003894$, γ

was 0.25, n was 46, α was 1, and the list of PNB was {0, 6, 7, 8, 9, 10, 14, 22, 23, 24, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 51, 52, 56, 57, 58, 59, 60, 61, 62, 63, 78, 162, 170, 179, 186, 191, 199, 219, 220, 221, 222, 223}; thus, the attack on ChaCha20/7.5 was inefficient because this is beyond the security level of ChaCha20. In summary, we clarified that using $\mathcal{ID}$-$\mathcal{OD}$ pair presented by Coutinho and Neto [7] could improve the attack on ChaCha20/7.25 but not that on ChaCha20/7.5.

We conclude that it is crucial to comprehensively analyze not only forward biases but also backward biases, i.e., the PNB. Moreover, this study shows the relevance of a comprehensive analysis of backward biases for ChaCha20 for the first time, and we are convinced that this study is relevant from such a viewpoint. We believe that our study will be the first step toward an attack on more rounds of ChaCha20, e.g., ChaCha20/8.

6 Conclusion

In this study, we 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 the *PNB-based differential attack*. The proposed approach allowed us to perform the most effective differential attack on the 7.25-round ChaCha, i.e., ChaCha20/7.25, with time and data complexities of $2^{255.62}$ and $2^{37.49}$, respectively. Moreover, using $\mathcal{ID}$-$\mathcal{OD}$ pair presented in the best existing method by Coutinho and Neto [7], we further improved the attack on ChaCha20/7.25, with time and data complexities of $2^{244.22}$ and $2^{69.14}$, respectively. To the best of our knowledge, this is the best attack on the reduced-round version of ChaCha.

In this study, we focus solely on the single-bit differential technique; therefore, it may be possible to improve the proposed attack by focusing on multi-bit differential or differential-linear techniques. Moreover, the PNB-based differential attack may contribute to improving existing differential attacks on the Salsa stream cipher. These are left as relevant future works.

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 Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, volume 5086 of *Lecture Notes in Computer Science*, pages 470–488. Springer, 2008.

[2] Christof Beierle, Gregor Leander, and Yosuke Todo. Improved Differential-Linear Attacks with Applications to ARX Ciphers. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.

[3] Daniel J. Bernstein. ChaCha, A Variant of Salsa20. In *Workshop Record of SASC*, volume 8, 2008.

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

[5] 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.

[6] Murilo Coutinho and T. C. Souza Neto. New multi-bit differentials to improve attacks against chacha. *IACR Cryptol. ePrint Arch.*, 2020:350, 2020.

[7] Murilo Coutinho and Tertuliano C. Souza Neto. Improved linear approximations to ARX ciphers and attacks against chacha. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 711–740. Springer, 2021.

[8] Paul Crowley. Truncated differential cryptanalysis of five rounds of Salsa20. *IACR Cryptol. ePrint Arch.*, 2005:375, 2005.

[9] Kakumani K. C. Deepthi and Kunwar Singh. Cryptanalysis of Salsa and ChaCha: Revisited. In Jiankun Hu, Ibrahim Khalil, Zahir Tari, and Sheng Wen, editors, *Mobile Networks and Management - 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings*, volume 235 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 324–338. Springer, 2017.

[10] Sabyasachi Dey and Santanu Sarkar. Improved analysis for reduced round Salsa and Chacha. *Discret. Appl. Math.*, 227:58–69, 2017.

[11] Sabyasachi Dey and Santanu Sarkar. Proving the biases of Salsa and ChaCha in differential attack. *Des. Codes Cryptogr.*, 88(9):1827–1856, 2020.

[12] Sabyasachi Dey and Santanu Sarkar. A theoretical investigation on the distinguishers of Salsa and ChaCha. *Discret. Appl. Math.*, 302:147–162, 2021.

[13] Simon Fischer, Willi Meier, Côme Berbain, Jean-François Biasse, and Matthew J. B. Robshaw. Non-randomness in estream candidates salsa20 and TSC-4. In Rana Barua and Tanja Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 2–16. Springer, 2006.

[14] Tsukasa Ishiguro, Shinsaku Kiyomoto, and Yutaka Miyake. Latin dances revisited: New analytic results of salsa20 and chacha. In Sihan Qing, Willy Susilo, Guilin Wang, and Dongmei Liu, editors, *Information and Communications Security - 13th International Conference, ICICS 2011, Beijing, China, November 23-26, 2011. Proceedings*, volume 7043 of *Lecture Notes in Computer Science*, pages 255–266. Springer, 2011.

[15] Ryoma Ito. Rotational Cryptanalysis of Salsa Core Function. In Willy Susilo, Robert H. Deng, Fuchun Guo, Yannan Li, and Rolly Intan, editors, *Information Security - 23rd International Conference, ISC 2020, Bali, Indonesia, December 16-18, 2020, Proceedings*, 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] Itsik Mantin and Adi Shamir. A Practical Attack on Broadcast RC4. In Mitsuru Matsui, editor, *Fast Software Encryption, 8th International Workshop, FSE 2001 Yokohama, Japan, April 2-4, 2001, Revised Papers*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer, 2001.

[18] Yoav Nir and Adam Langley. Chacha20 and poly1305 for IETF protocols. *RFC*, 8439:1–46, 2018.

[19] Zhenqing Shi, Bin Zhang, Dengguo Feng, and Wenling Wu. Improved key recovery attacks on reduced-round salsa20 and chacha. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*, volume 7839 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2012.