New Multi-bit Differentials to Improve Attacks Against ChaCha

Murilo Coutinho
murilo9988@gmail.com

T. C. Souza Neto
tsouzaneto@gmail.com

March 2020

Abstract

The stream cipher ChaCha is an ARX type algorithm developed by Daniel Bernstein in 2008. Since its development, ChaCha has received a lot of attention and is currently being used in several systems. The most powerful cryptanalysis of reduced versions of this cipher was presented by Choudhuri and Maitra on FSE 2017 by using differential-linear cryptanalysis. In their work they show that is possible to obtain linear relations between bits from different rounds with high probability and use the proposed equations to create multi-bit differentials and improve previous attacks. In this work, we provide new linear approximations that can be used in a similar fashion but with increased efficiency. Therefore, we show that using these new equations is possible to improve the attacks against 6 and 7 rounds of ChaCha.

1 Introduction

In 2005, Bernstein proposed the stream cipher Salsa20 [4] as a contender to the eSTREAM [19], the ECRYPT Stream Cipher Project. As outlined by the author, Salsa20 is an ARX type family of algorithms which can be ran with several number of rounds, including the well known Salsa20/12 and Salsa20/8 versions.

In 2008, Bernstein proposed some modifications to Salsa20 in order to provide better diffusion per round and higher resistance to cryptanalysis. These changes originated a new stream cipher, a variant which he called ChaCha [3]. Although Salsa20 was one of the winners of the eSTREAM competition, ChaCha has received much more attention through the years. Nowadays, we see the usage of this cipher in several projects and applications.

ChaCha, along Poly1305 [2], is in one of the cipher suits of the new TLS 1.3 [14], which is actually used by Google on both Chrome and Android. ChaCha is used not only in TLS but in many other protocols such as SSH, Noise, and S/MIME 4.0. In addition, the RFC 7634 proposes the use of ChaCha in IKE and IPsec. ChaCha is used not only for encryption, but also as a pseudo-random number generator. For example, in any operating system running Linux kernel 4.8 or newer [18, 21]. Additionally, ChaCha is used in several applications, for example, WireGuard (VPN), Keepass (password manager), and Veracrypt (disk encryption). See [12] for a huge list of applications, protocols, and libraries using ChaCha.

Related Work. Since ChaCha is so heavily used, it is very important to understand its security level. Indeed, the cryptanalysis of ChaCha is well understood and several authors studied its security [1, 11, 6, 10, 13, 16, 15, 17, 5, 20, 22, 8, 7, 9] which show weaknesses in the reduced rounds of the cipher.

The cryptanalysis of Salsa20 was introduced by Crowley [6] in 2005. Crowley developed a differential attack against Salsa20/5, namely the 5-round version of Salsa20, and received the $1000 prize offered by Bernstein for the most interesting Salsa20 cryptanalasys in that year. In 2006, Fischer et al [10] improved the attack against Salsa20/5 and presented an attack against Salsa20/6.

Probably the most important cryptanalysis in this regard was proposed by Aumasson et al. at FSE 2008 [1] with the introduction of Probabilistic Neutral Bits (PNBs), showing attacks against Salsa20/7, Salsa20/8, ChaCha20/6 and ChaCha20/7. After that, several authors proposed small enhancements on the attack of Aumasson et al. The work by Shi et al [20] introduced the concept of Column Chaining Distinguisher (CCD) to achieve some incremental advancements over [1] for both Salsa and ChaCha.

Maitra, Paul and Meier [15] studied an interesting observation regarding round reversal of Salsa, but no significant cryptanalytic improvement could be obtained using this method. Maitra [16] used a technique of Chosen IVs to obtain certain improvements over existing results. Dey and Sarkar [8] showed how to chose values for the PNB to further improve the attack.

However, the best results known so far concerning attacks to Salsa and ChaCha were given by Choudhuri and Maitra [5] in FSE 2017. They used the technique of differential-linear cryptanalysis and investigated the mathematical structure of both Salsa and ChaCha in order to find differential characteristics with much higher biases.

Our Contribution. In this work, we provide new linear approximations that can be used to improve attacks against ChaCha. At first sight, this linear approximations seem useless leading to a worst differential-linear bias. However, they have fewer terms which leads to fewer non-linear transitions when extending the attack one round further and to many more neutral bits when applying the techniques of Aumasson [1]. We summarize our findings along with other significant attacks for comparison in Table 1. We should note that it is possible to find attacks with less complexity for related key attacks, but we do not consider them in this work.

Organization of the paper. In Section 2, we provide an overview of previous results, including a description of ChaCha, a summary of differential-linear cryptanalysis and a review of the techniques developed by Choudhuri and Maitra in [5]. In Section 3, we theoretically develop new linear relations between bits of different rounds for ChaCha and then show that these new results leads to a better distinguisher for ChaCha reduced to 6 rounds. Then, in Section

Rounds	Time Complexity	Data Complexity	Reference
4	2^{6}	2^{6}	[5]
4.5	2^{12}	2^{12}	[5]
5	2^{16}	2^{16}	[5]
6	2^{139}	2^{30}	[1]
	2^{136}	2^{28}	[20]
	2^{130}	2^{35}	[5]
	$2^{127.5}$	$2^{37.5}$	[5]
	2^{116}	2^{116}	[5]
	$2^{102.2}$	2^{56}	This work
	2^{75}	2^{75}	This work
7	2^{48}	2^{27}	[1]
	$2^{246.5}$	2^{27}	[20]
	$2^{238.9}$	2^{96}	[16]
	$2^{237.7}$	2^{96}	[5]
	$2^{231.9}$	2^{50}	This work

Table 1: The best attacks against ChaCha with 256-bit key.

4, we show how these new results may be applied to attack ChaCha using the Probabilitic Neutral Bits technique, introduced by Aumasson [1], effectively improving attack against ChaCha reduced to 6 and 7 rounds. Finnaly, Section 5 presents the conclusion.

2 Specifications and Preliminaries

The main notation we will use through out the paper is defined in Table 2. Next we define the algorithm ChaCha.

2.1 ChaCha

The stream cipher Salsa20 was proposed by Bernstein [4] to the *eSTREAM* competition and latter Bernstein proposed ChaCha [3] as an improvement of Salsa20. ChaCha consists of a series of ARX (addition, rotation, and XOR) operations on 32-bit words, being highly efficient in software and hardware. Each round of ChaCha has a total of 16 bitwise XOR, 16 addition modulo 2^{32} and 16 constant-distance rotations.

ChaCha operates on a state of 64 bytes, organized as a 4×4 matrix with 32-bit integers, initialized with a 256-bit key $k_0, k_1, ..., k_7$, a 64-bit nonce v_0, v_1 and a 64-bit counter t_0, t_1 (we may also refer to the nonce and counter words as IV words), and 4 constants $c_0 = $ 0x61707865, $c_1 = $ 0x3320646e, $c_2 = $ 0x79622d32 and $c_3 = $ 0x6b206574. For ChaCha, we have the following initial state matrix:

Notation	Description
X	a 4 × 4 state matrix of ChaCha
$X^{(0)}$	initial state matrix of ChaCha
$X^{(R)}$	state matrix after application of R round functions
Z	output of ChaCha, $Z = X + X^{(R)}$
$x_i^{(R)}$	i^{th} word of the state matrix $X^{(R)}$ (words arranged in row major)
$x_{i,j}^{(R)}$	j^{th} bit of i^{th} word of the state matrix $X^{(R)}$
$x + y$	addition of x and y modulo 2^{32}
$x - y$	subtraction of x and y modulo 2^{32}
$x \oplus y$	bitwise XOR of x and y
$x \lll n$	rotation of x by n bits to the left
$x \ggg n$	rotation of x by n bits to the right
Δx	XOR difference of x and x'. $\Delta x = x \oplus x'$
$\Delta_i^{(R)}$	differential $\Delta_i^{(R)} = x_i^{(R)} \oplus x'^{(R)}_i$
$\Delta_{i,j}^{(R)}$	differential $\Delta_{i,j}^{(R)} = x_{i,j}^{(R)} \oplus x'^{(R)}_{i,j}$
$\Pr(E)$	probability of occurrence of an event E
$\varepsilon_{(x_1 \oplus ... \oplus x_m)}$	bias of event $E = \{\Delta x_1 \oplus ... \oplus \Delta x_m = 0\}$
$\mathcal{I}D$	input differential
$\mathcal{O}D$	output differential

Table 2: Notation

$$X^{(0)} = \begin{pmatrix} x_0^{(0)} & x_1^{(0)} & x_2^{(0)} & x_3^{(0)} \\ x_4^{(0)} & x_5^{(0)} & x_6^{(0)} & x_7^{(0)} \\ x_8^{(0)} & x_9^{(0)} & x_{10}^{(0)} & x_{11}^{(0)} \\ x_{12}^{(0)} & x_{13}^{(0)} & x_{14}^{(0)} & x_{15}^{(0)} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & t_1 & v_0 & v_1 \end{pmatrix}. \tag{1}$$

The state matrix is modified in each round by a *Quarter Round Function* (QRF), named $QR\left(x_a^{(r-1)}, x_b^{(r-1)}, x_c^{(r-1)}, x_d^{(r-1)}\right)$, which receives and updates 4 integers in the following way:

$$\begin{aligned} x_{a\prime}^{(r-1)} &= x_a^{(r-1)} + x_b^{(r-1)} \\ x_{d\prime}^{(r-1)} &= (x_d^{(r-1)} \oplus x_{a\prime}^{(r-1)}) \lll 16 \\ x_{c\prime}^{(r-1)} &= x_c^{(r-1)} + x_{d\prime}^{(r-1)} \\ x_{b\prime}^{(r-1)} &= (x_b^{(r-1)} \oplus x_{c\prime}^{(r-1)}) \lll 12 \\ x_a^{(r)} &= x_{a\prime}^{(r-1)} + x_{b\prime}^{(r-1)} \\ x_d^{(r)} &= (x_{d\prime}^{(r-1)} \oplus x_a^{(r)}) \lll 8 \\ x_c^{(r)} &= x_{c\prime}^{(r-1)} + x_d^{(r)} \\ x_b^{(r)} &= (x_{b\prime}^{(r-1)} \oplus x_c^{(r)}) \lll 7 \end{aligned} \tag{2}$$

One round of ChaCha is defined as 4 applications of QR. There is, however, a difference between odd and even rounds. For odd rounds $r \in \{1, 3, 5, 7, ...\}$,

$X^{(r)}$ is obtained from $X^{(r-1)}$ by applying

$$\begin{aligned}
\left(x_0^{(r)}, x_4^{(r)}, x_8^{(r)}, x_{12}^{(r)}\right) &= QR\left(x_0^{(r-1)}, x_4^{(r-1)}, x_8^{(r-1)}, x_{12}^{(r-1)}\right)\\
\left(x_1^{(r)}, x_5^{(r)}, x_9^{(r)}, x_{13}^{(r)}\right) &= QR\left(x_1^{(r-1)}, x_5^{(r-1)}, x_9^{(r-1)}, x_{13}^{(r-1)}\right)\\
\left(x_2^{(r)}, x_6^{(r)}, x_{10}^{(r)}, x_{14}^{(r)}\right) &= QR\left(x_2^{(r-1)}, x_6^{(r-1)}, x_{10}^{(r-1)}, x_{14}^{(r-1)}\right)\\
\left(x_3^{(r)}, x_7^{(r)}, x_{11}^{(r)}, x_{15}^{(r)}\right) &= QR\left(x_3^{(r-1)}, x_7^{(r-1)}, x_{11}^{(r-1)}, x_{15}^{(r-1)}\right)
\end{aligned}.$$

On the other hand, for even rounds $r \in \{2, 4, 6, 8, , ...\}$ $X^{(r)}$ is calculated from $X^{(r-1)}$ by applying

$$\begin{aligned}
\left(x_0^{(r)}, x_5^{(r)}, x_{10}^{(r)}, x_{15}^{(r)}\right) &= QR\left(x_0^{(r-1)}, x_5^{(r-1)}, x_{10}^{(r-1)}, x_{15}^{(r-1)}\right)\\
\left(x_1^{(r)}, x_6^{(r)}, x_{11}^{(r)}, x_{12}^{(r)}\right) &= QR\left(x_1^{(r-1)}, x_6^{(r-1)}, x_{11}^{(r-1)}, x_{12}^{(r-1)}\right)\\
\left(x_2^{(r)}, x_7^{(r)}, x_8^{(r)}, x_{13}^{(r)}\right) &= QR\left(x_2^{(r-1)}, x_7^{(r-1)}, x_8^{(r-1)}, x_{13}^{(r-1)}\right)\\
\left(x_3^{(r)}, x_4^{(r)}, x_9^{(r)}, x_{14}^{(r)}\right) &= QR\left(x_3^{(r-1)}, x_4^{(r-1)}, x_9^{(r-1)}, x_{14}^{(r-1)}\right)
\end{aligned}.$$

The algorithm ChaCha20/R is then defined as the sum of the initial state with the state after R rounds

$$Z = X + X^{(R)}.$$

One should note that it is possible to parallelize each application of the QRF on each round and also that each round is reversible. Hence, we can compute $X^{(r-1)}$ from $X^{(r)}$. For more information on ChaCha, we refer to [3].

2.2 Differential-Linear Analysis

In this section, we describe the technique of Differential-Linear cryptanalysis as used to attack ChaCha. Let $X^{(r)}$ and $X'^{(r)}$ be two state matrices after r rounds. We denote the differential of the state matrix as $\Delta X^{(r)} = X^{(r)} \oplus X'^{(r)}$ and the differential of individual words as $\Delta x_i^{(r)} = x_i^{(r)} \oplus x_i'^{(r)}$. The cryptanalysis starts by defining a differential for the initial state, called input differential $\mathcal{ID}$. Then, we try to find biases after a certain number of rounds r, denoted as output differential $\mathcal{OD}$. Let $x_{i,j}^{(r)}$ denote the j-th bit of the i-th word of the state matrix after r rounds and let $\mathcal{J}$ be a set of bits. Also, let σ and σ' be linear combinations of bits in the set $\mathcal{J}$

$$\sigma = \left(\bigoplus_{(i,j)\in\mathcal{J}} x_{i,j}^{(r)}\right), \quad \sigma' = \left(\bigoplus_{(i,j)\in\mathcal{J}} x_{i,j}'^{(r)}\right).$$

Then

$$\Delta\sigma = \left(\bigoplus_{(i,j)\in\mathcal{J}} \Delta x_{i,j}^{(r)}\right)$$

is the linear combination of the differentials. We can write

$$\Pr\left[\Delta\sigma = 0 | \Delta X^{(0)}\right] = \frac{1}{2}(1+\varepsilon_d), \tag{3}$$

where ε_d is a differential bias.

Using linear cryptanalysis, it is possible to go further and find new relations between the initial state matrix and the state matrix after $R > r$ rounds. To do so, let $\mathcal{L}$ denote another set of bits and define

$$\rho = \left(\bigoplus_{(i,j)\in\mathcal{L}} x_{i,j}^{(R)}\right), \qquad \rho' = \left(\bigoplus_{(i,j)\in\mathcal{L}} x_{i,j}'^{(R)}\right).$$

Then, as before,

$$\Delta\rho = \left(\bigoplus_{(i,j)\in\mathcal{L}} \Delta x_{i,j}^{(R)}\right).$$

We can define

$$\Pr[\sigma = \rho] = \frac{1}{2}(1+\varepsilon_L),$$

where ε_L is the linear bias. We want to find γ such that

$$\Pr\left[\Delta\rho = 0 | \Delta X^{(0)}\right] = \frac{1}{2}(1+\gamma).$$

To compute γ, we write (to simplify the notation we make the conditional to $\Delta X^{(0)}$ implicit):

$$\begin{aligned}\Pr[\Delta\sigma = \Delta\rho] &= \Pr[\sigma = \rho]\cdot\Pr\left[\sigma' = \rho'\right] + \Pr[\sigma = \bar{\rho}]\cdot\Pr\left[\sigma' = \overline{\rho'}\right] \\ &= \frac{1}{2}\left(1+\varepsilon_L\right)\cdot\frac{1}{2}\left(1+\varepsilon_L\right) + \frac{1}{2}\left(1-\varepsilon_L\right)\cdot\frac{1}{2}\left(1-\varepsilon_L\right) \\ &= \frac{1}{2}\left(1+\varepsilon_L^2\right).\end{aligned}$$

Then,

$$\begin{aligned}\Pr[\Delta\rho = 0] &= \Pr[\Delta\sigma = 0]\cdot\Pr[\Delta\sigma = \Delta\rho] + \Pr[\Delta\sigma = 1]\cdot\Pr[\Delta\sigma = \overline{\Delta\rho}] \\ &= \frac{1}{2}\left(1+\varepsilon_d\right)\cdot\frac{1}{2}\left(1+\varepsilon_L^2\right) + \frac{1}{2}\left(1-\varepsilon_d\right)\cdot\frac{1}{2}\left(1-\varepsilon_L^2\right) \\ &= \frac{1}{2}\left(1+\varepsilon_d\cdot\varepsilon_L^2\right).\end{aligned}$$

Therefore, the differential-linear bias is given by $\gamma = \varepsilon_d\cdot\varepsilon_L^2$, which defines a distinguisher with complexity $\mathcal{O}\left(\frac{1}{\varepsilon_d^2\varepsilon_L^4}\right)$.

2.3 Multi-bit Differential for Reduced Round ChaCha

In this section we review the work presented in [5], which is the basis for our attack. We note that the paper works with Salsa and ChaCha, but here we focus only on ChaCha. In their work, it was developed the theory for selecting specific combination of bits to give high biases for Chacha. To do that the authors analyze the QRF directly, representing each equation in its bits. For example, from the first line of Eq. (2) we can write

$$x_{a\prime,i}^{(r-1)} = x_{a,i}^{(r-1)} \oplus x_{b,i}^{(r-1)} \oplus C_i^1,$$

where C_i^1 denotes the i-th bit of the carry for the first sum, this equation follows from the well known fact that $x + y = x \oplus y \oplus Carry(x, y)$.

After working with all the equations from the QRF, is possible to write bits from round $m-1$ in terms of linear equations from bits from round m and carry bits:

$$x_{b,i}^{(m-1)} = x_{b,i+19}^{(m)} \oplus x_{c,i+12}^{(m)} \oplus x_{d,i}^{(m)} \oplus x_{c,i}^{(m)} \oplus C_i^4 \tag{4}$$

$$x_{a,i}^{(m-1)} = x_{a,i}^{(m)} \oplus x_{b,i+7}^{(m)} \oplus x_{b,i+19}^{(m)} \oplus x_{c,i+12}^{(m)} \oplus x_{d,i}^{(m)} \oplus C_i^4 \oplus C_i^1 \oplus C_i^3 \tag{5}$$

$$x_{c,i}^{(m-1)} = x_{d,i}^{(m)} \oplus x_{c,i}^{(m)} \oplus x_{d,i+8}^{(m)} \oplus x_{a,i}^{(m)} \oplus C_i^2 \oplus C_i^4 \tag{6}$$

$$x_{d,i}^{(m-1)} = x_{d,i+24}^{(m)} \oplus x_{a,i+16}^{(m)} \oplus x_{a,i}^{(m)} \oplus x_{c,i}^{(m)} \oplus x_{b,i+7}^{(m)} \oplus C_i^3 \tag{7}$$

From this equations, we can derive the following Lemma:

Lemma 1. *Let*

$$\begin{aligned}
\Delta A^{(m)} &= \Delta x_{\alpha,0}^{(m)} \oplus \Delta x_{\beta,7}^{(m)} \oplus \Delta x_{\beta,19}^{(m)} \oplus \Delta x_{\gamma,12}^{(m)} \oplus \Delta x_{\delta,0}^{(m)} \\
\Delta B^{(m)} &= \Delta x_{\beta,19}^{(m)} \oplus \Delta x_{\gamma,0}^{(m)} \oplus \Delta x_{\gamma,12}^{(m)} \oplus \Delta x_{\delta,0}^{(m)} \\
\Delta C^{(m)} &= \Delta x_{\delta,0}^{(m)} \oplus \Delta x_{\gamma,0}^{(m)} \oplus \Delta x_{\delta,8}^{(m)} \oplus \Delta x_{\alpha,0}^{(m)} \\
\Delta D^{(m)} &= \Delta x_{\delta,24}^{(m)} \oplus \Delta x_{\alpha,16}^{(m)} \oplus \Delta x_{\alpha,0}^{(m)} \oplus \Delta x_{\gamma,0}^{(m)} \oplus \Delta x_{\beta,7}^{(m)}
\end{aligned}$$

After m rounds of ChaCha, the following holds:

$$\left|\varepsilon_{(A(m))}\right| = \left|\varepsilon_{\left(x_{\alpha,0}^{(m-1)}\right)}\right|, \left|\varepsilon_{\left(B^{(m)}\right)}\right| = \left|\varepsilon_{\left(x_{\beta,0}^{(m-1)}\right)}\right|$$

$$\left|\varepsilon_{\left(C^{(m)}\right)}\right| = \left|\varepsilon_{\left(x_{\gamma,0}^{(m-1)}\right)}\right|, \left|\varepsilon_{\left(D^{(m)}\right)}\right| = \left|\varepsilon_{\left(x_{\delta,0}^{(m-1)}\right)}\right|$$

The tuples $(\alpha, \beta, \gamma, \delta)$ vary depending on whether m is odd or even.

- *Case I. m is odd:*

$$(\alpha, \beta, \gamma, \delta) \in \{(0, 4, 8, 12), (1, 5, 9, 13), (2, 6, 14, 2), (3, 7, 11, 15)\}.$$

- *Case II. m is even:*

$$(\alpha, \beta, \gamma, \delta) \in \{(0,5,10,15),(1,6,11,12),(2,7,8,13),(3,4,9,14)\}.$$

Proof. See [5]. □

Additionally, using the linear approximations for addition proposed by [23], we can use Eqs. (4)-(7) to construct a series of linear approximations for one round of ChaCha:

Lemma 2. *For one active input bit in round $m-1$ and multiple active output bits in round m, the following holds.*

$$\begin{aligned}
x_{b,i}^{(m-1)} &= x_{b,i+19}^{(m)} \oplus x_{c,i+12}^{(m)} \oplus x_{d,i}^{(m)} \oplus x_{c,i}^{(m)} \oplus x_{d,i-1}^{(m)} && w.p. \quad \tfrac{1}{2}\left(1+\tfrac{1}{2}\right) \\
x_{a,i}^{(m-1)} &= x_{a,i}^{(m)} \oplus x_{b,i+7}^{(m)} \oplus x_{b,i+19}^{(m)} \oplus x_{c,i+12}^{(m)} \oplus x_{d,i}^{(m)} \oplus \\
&\quad\; x_{b,i+18}^{(m)} \oplus x_{c,i+11}^{(m)} \oplus x_{d,i-2}^{(m)} \oplus x_{d,i+6}^{(m)} && w.p. \quad \tfrac{1}{2}\left(1+\tfrac{1}{2^4}\right) \\
x_{c,i}^{(m-1)} &= x_{d,i}^{(m)} \oplus x_{c,i}^{(m)} \oplus x_{d,i+8}^{(m)} \oplus x_{a,i}^{(m)} \oplus x_{a,i-1}^{(m)} \oplus \\
&\quad\; x_{d,i+7}^{(m)} \oplus x_{d,i-1}^{(m)} && w.p. \quad \tfrac{1}{2}\left(1+\tfrac{1}{2^2}\right) \\
x_{d,i}^{(m-1)} &= x_{d,i+24}^{(m)} \oplus x_{a,i+16}^{(m)} \oplus x_{a,i}^{(m)} \oplus x_{c,i}^{(m)} \oplus \\
&\quad\; x_{b,i+7}^{(m)} \oplus x_{c,i-1}^{(m)} \oplus x_{b,i+6}^{(m)} && w.p. \quad \tfrac{1}{2}\left(1+\tfrac{1}{2}\right)
\end{aligned}$$

Proof. See [5]. □

Finally, using Lemmas 1 and 2 is possible to find linear approximations for two rounds of ChaCha.

Lemma 3. *Each of the following holds with probability* $\frac{1}{2}\left(1+\frac{1}{2}\right)$

$$\begin{aligned}
x_{8,0}^{(3)} &= x_{13,24}^{(5)} \oplus x_{1,16}^{(5)} \oplus x_{1,0}^{(5)} \oplus x_{9,0}^{(5)} \oplus x_{5,7}^{(5)} \oplus x_{12,0}^{(5)} \oplus x_{8,0}^{(5)} \oplus \\
&\quad\; x_{12,8}^{(5)} \oplus x_{0,0}^{(5)} \oplus x_{2,0}^{(5)} \oplus x_{6,7}^{(5)} \oplus x_{6,19}^{(5)} \oplus x_{10,12}^{(5)} \oplus x_{14,0}^{(5)} \oplus \\
&\quad\; x_{13,0}^{(5)} \oplus x_{1,24}^{(5)} \oplus x_{1,8}^{(5)} \oplus x_{9,8}^{(5)} \oplus x_{5,15}^{(5)} \oplus x_{9,7}^{(5)} \oplus x_{5,14}^{(5)} \\
x_{9,0}^{(3)} &= x_{14,24}^{(5)} \oplus x_{2,16}^{(5)} \oplus x_{2,0}^{(5)} \oplus x_{10,0}^{(5)} \oplus x_{6,7}^{(5)} \oplus x_{13,0}^{(5)} \oplus x_{9,0}^{(5)} \oplus \\
&\quad\; x_{13,8}^{(5)} \oplus x_{1,0}^{(5)} \oplus x_{3,0}^{(5)} \oplus x_{7,7}^{(5)} \oplus x_{7,19}^{(5)} \oplus x_{11,12}^{(5)} \oplus x_{15,0}^{(5)} \oplus \\
&\quad\; x_{14,0}^{(5)} \oplus x_{2,24}^{(5)} \oplus x_{2,8}^{(5)} \oplus x_{10,8}^{(5)} \oplus x_{6,15}^{(5)} \oplus x_{10,7}^{(5)} \oplus x_{6,14}^{(5)} \\
x_{10,0}^{(3)} &= x_{15,24}^{(5)} \oplus x_{3,16}^{(5)} \oplus x_{3,0}^{(5)} \oplus x_{11,0}^{(5)} \oplus x_{7,7}^{(5)} \oplus x_{14,0}^{(5)} \oplus x_{10,0}^{(5)} \oplus \\
&\quad\; x_{14,8}^{(5)} \oplus x_{2,0}^{(5)} \oplus x_{0,0}^{(5)} \oplus x_{4,7}^{(5)} \oplus x_{4,19}^{(5)} \oplus x_{8,12}^{(5)} \oplus x_{12,0}^{(5)} \oplus \\
&\quad\; x_{15,0}^{(5)} \oplus x_{3,24}^{(5)} \oplus x_{3,8}^{(5)} \oplus x_{11,8}^{(5)} \oplus x_{7,15}^{(5)} \oplus x_{11,7}^{(5)} \oplus x_{7,14}^{(5)} \\
x_{11,0}^{(3)} &= x_{12,24}^{(5)} \oplus x_{0,16}^{(5)} \oplus x_{0,0}^{(5)} \oplus x_{8,0}^{(5)} \oplus x_{4,7}^{(5)} \oplus x_{15,0}^{(5)} \oplus x_{11,0}^{(5)} \oplus \\
&\quad\; x_{15,8}^{(5)} \oplus x_{3,0}^{(5)} \oplus x_{1,0}^{(5)} \oplus x_{5,7}^{(5)} \oplus x_{5,19}^{(5)} \oplus x_{9,12}^{(5)} \oplus x_{13,0}^{(5)} \oplus \\
&\quad\; x_{12,0}^{(5)} \oplus x_{0,24}^{(5)} \oplus x_{0,8}^{(5)} \oplus x_{8,8}^{(5)} \oplus x_{4,15}^{(5)} \oplus x_{8,7}^{(5)} \oplus x_{4,14}^{(5)}
\end{aligned}$$

Proof. See [5]. □

With these results, in [5], the authors show that using as $\mathcal{ID}$ at $x_{13,13}^{(0)}$ and $\mathcal{OD}$ at $x_{11,0}^{(3)}$, is possible to obtain $\varepsilon_d = -0.0272 \approx -\frac{1}{2^{5.2}}$, experimentally. And from Lemma 1 is possible to extend to a 4-round differential-linear bias with $\varepsilon_L = 1$ when the $\mathcal{OD}$ is $x_{1,0}^{(4)} \oplus x_{11,0}^{(4)} \oplus x_{12,8}^{(4)} \oplus x_{12,0}^{(4)}$. Further, is possible to extend to a 5-round differential linear bias using the last equation from Lemma 3 with probability $\frac{1}{2}\left(1+\frac{1}{2}\right)$. This gives a total differential-linear 5th round bias of $\varepsilon_d \cdot \varepsilon_L^2 \approx -0.0068 = -\frac{1}{2^{7.2}}$. This leads to a 5 round distinguisher with complexity $\approx 2^{16}$.

Extending the linear approximation for 3 rounds come at a cost. As discussed prior to the above lemma, for ChaCha, setting $i = 0$ in Lemma 1 allows linear approximation of probability 1 for LSB variables. The cost is thus determined by the non LSB variables. A simple count of the non LSB variables in the form (Variable Type, # non LSB occurrence) gives $(x_a, 3), (x_b, 5), (x_c, 3)$, and $(x_d, 2)$. Now, using the probabilities of Lemma 2 and Lemma 3 (to attach the corresponding weight to each variable), the linear bias is

$$\varepsilon_L = \frac{1}{2 \cdot 1 + 3 \cdot 4 + 5 \cdot 1 + 3 \cdot 2 + 2 \cdot 1} = \frac{1}{2^{26}}.$$

This leads to a 6 round bias of $\varepsilon_L^2 \varepsilon_d \approx \frac{1}{2^{57.2}}$. The distinguisher for this bias has a complexity of 2^{116} which was, until now, the currently best known 6 round attack on ChaCha.

3 New Linear Approximations for ChaCha

The attack presented in this section follows the techniques used in [5]. More precisely, we derive a new linear approximation that leads to a better distinguisher for 6 rounds of ChaCha. We start by defining the following lemma:

Lemma 4. *Let*

$$\Delta E^{(m)} = \Delta x_{\alpha,0}^{(m)} \oplus \Delta x_{\beta,7}^{(m)} \oplus \Delta x_{\gamma,0}^{(m)}$$

After m rounds of ChaCha, the following holds:

$$\left|\varepsilon_{(E^{(m)})}\right| = \left|\varepsilon_{(x_{\alpha,0}^{(m-1)} \oplus x_{\beta,0}^{(m-1)})}\right|$$

The tuples (α, β, γ) vary depending on whether m is odd or even.

- *Case I. m odd: $(\alpha, \beta, \gamma) \in \{(0,4,8),(1,5,9),(2,6,10),(3,7,11)\}$*
- *Case II. m even: $(\alpha, \beta, \gamma) \in \{(0,5,10),(1,6,11),(2,7,8),(3,4,9)\}$*

Proof. The proof follows directly from Lemma 1, by doing

$$\Delta E^{(m)} = \Delta A^{(m)} \oplus \Delta B^{(m)}.$$

□

Lemma 5. *When m is even, each of the following holds with probability* $\frac{1}{2}(1+\frac{1}{2})$

$$\begin{aligned} x_{0,0}^{(m-2)} \oplus x_{4,0}^{(m-2)} = {} & x_{0,0}^{(m)} \oplus x_{2,0}^{(m)} \oplus x_{4,26}^{(m)} \oplus x_{5,7}^{(m)} \oplus x_{5,19}^{(m)} \oplus \\ & x_{8,0}^{(m)} \oplus x_{9,7}^{(m)} \oplus x_{9,19}^{(m)} \oplus x_{10,12}^{(m)} \oplus x_{13,0}^{(m)} \oplus \\ & x_{13,8}^{(m)} \oplus x_{14,6}^{(m)} \oplus x_{14,7}^{(m)} \oplus x_{15,0}^{(m)} \oplus \end{aligned}$$

$$\begin{aligned} x_{1,0}^{(m-2)} \oplus x_{5,0}^{(m-2)} = {} & x_{1,0}^{(m)} \oplus x_{3,0}^{(m)} \oplus x_{5,26}^{(m)} \oplus x_{6,7}^{(m)} \oplus x_{6,19}^{(m)} \oplus \\ & x_{9,0}^{(m)} \oplus x_{10,7}^{(m)} \oplus x_{10,19}^{(m)} \oplus x_{11,12}^{(m)} \oplus x_{12,0}^{(m)} \oplus \\ & x_{14,0}^{(m)} \oplus x_{14,8}^{(m)} \oplus x_{15,6}^{(m)} \oplus x_{15,7}^{(m)} \oplus \end{aligned}$$

$$\begin{aligned} x_{2,0}^{(m-2)} \oplus x_{6,0}^{(m-2)} = {} & x_{0,0}^{(m)} \oplus x_{2,0}^{(m)} \oplus x_{6,26}^{(m)} \oplus x_{7,7}^{(m)} \oplus x_{7,19}^{(m)} \oplus \\ & x_{8,12}^{(m)} \oplus x_{10,0}^{(m)} \oplus x_{11,7}^{(m)} \oplus x_{11,19}^{(m)} \oplus x_{12,6}^{(m)} \oplus \\ & x_{12,7}^{(m)} \oplus x_{13,0}^{(m)} \oplus x_{15,0}^{(m)} \oplus x_{15,8}^{(m)} \oplus \end{aligned}$$

$$\begin{aligned} x_{3,0}^{(m-2)} \oplus x_{7,0}^{(m-2)} = {} & x_{1,0}^{(m)} \oplus x_{3,0}^{(m)} \oplus x_{4,7}^{(m)} \oplus x_{4,19}^{(m)} \oplus x_{7,26}^{(m)} \oplus \\ & x_{8,7}^{(m)} \oplus x_{8,19}^{(m)} \oplus x_{9,12}^{(m)} \oplus x_{11,0}^{(m)} \oplus x_{12,0}^{(m)} \oplus \\ & x_{12,8}^{(m)} \oplus x_{13,6}^{(m)} \oplus x_{13,7}^{(m)} \oplus x_{14,0}^{(m)} \oplus \end{aligned}$$

and when m is odd, each of the following also holds with probability $\frac{1}{2}(1+\frac{1}{2})$

$$\begin{aligned} x_{0,0}^{(m-2)} \oplus x_{5,0}^{(m-2)} = {} & x_{0,0}^{(m)} \oplus x_{2,0}^{(m)} \oplus x_{4,7}^{(m)} \oplus x_{4,19}^{(m)} \oplus x_{5,26}^{(m)} \oplus \\ & x_{8,12}^{(m)} \oplus x_{9,7}^{(m)} \oplus x_{9,19}^{(m)} \oplus x_{10,0}^{(m)} \oplus x_{12,0}^{(m)} \oplus \\ & x_{13,6}^{(m)} \oplus x_{13,7}^{(m)} \oplus x_{14,0}^{(m)} \oplus x_{14,8}^{(m)} \oplus \end{aligned}$$

$$\begin{aligned} x_{1,0}^{(m-2)} \oplus x_{6,0}^{(m-2)} = {} & x_{1,0}^{(m)} \oplus x_{3,0}^{(m)} \oplus x_{5,7}^{(m)} \oplus x_{5,19}^{(m)} \oplus x_{6,26}^{(m)} \oplus \\ & x_{9,12}^{(m)} \oplus x_{10,7}^{(m)} \oplus x_{10,19}^{(m)} \oplus x_{11,0}^{(m)} \oplus x_{13,0}^{(m)} \oplus \\ & x_{14,6}^{(m)} \oplus x_{14,7}^{(m)} \oplus x_{15,0}^{(m)} \oplus x_{15,8}^{(m)} \oplus \end{aligned}$$

$$\begin{aligned} x_{2,0}^{(m-2)} \oplus x_{7,0}^{(m-2)} = {} & x_{0,0}^{(m)} \oplus x_{2,0}^{(m)} \oplus x_{6,7}^{(m)} \oplus x_{6,19}^{(m)} \oplus x_{7,26}^{(m)} \oplus \\ & x_{8,0}^{(m)} \oplus x_{10,12}^{(m)} \oplus x_{11,7}^{(m)} \oplus x_{11,19}^{(m)} \oplus x_{12,0}^{(m)} \oplus \\ & x_{12,8}^{(m)} \oplus x_{14,0}^{(m)} \oplus x_{15,6}^{(m)} \oplus x_{15,7}^{(m)} \oplus \end{aligned}$$

$$\begin{aligned} x_{3,0}^{(m-2)} \oplus x_{4,0}^{(m-2)} = {} & x_{1,0}^{(m)} \oplus x_{3,0}^{(m)} \oplus x_{4,26}^{(m)} \oplus x_{7,7}^{(m)} \oplus x_{7,19}^{(m)} \oplus \\ & x_{8,7}^{(m)} \oplus x_{8,19}^{(m)} \oplus x_{9,0}^{(m)} \oplus x_{11,12}^{(m)} \oplus x_{12,6}^{(m)} \oplus \\ & x_{12,7}^{(m)} \oplus x_{13,0}^{(m)} \oplus x_{13,8}^{(m)} \oplus x_{15,0}^{(m)} \oplus \end{aligned}$$

Proof. Let

$$\mathcal{X} = \{(0,4,8,12),(1,5,9,13),(2,6,10,14),(3,7,11,15)\}$$

and

$$\mathcal{Y} = \{(0,5,10,15),(1,6,11,12),(2,7,8,13),(3,4,9,14)\}.$$

Using Eqs. (4) and (5) with $i = 0$ we get that

$$x_{a,0}^{(m-2)} \oplus x_{b,0}^{(m-2)} = x_{a,0}^{(m-1)} \oplus x_{b,7}^{(m-1)} \oplus x_{c,0}^{(m-1)},$$

with $(a,b,c,d) \in \mathcal{X}$ when m is even, and with $(a,b,c,d) \in \mathcal{Y}$ when m is odd. From this point, we can use Eq. (5) and Eq. (6) to replace $x_{a,0}^{(m-1)}$ and $x_{c,0}^{(m-1)}$, respectively, by a linear approximation with probability 1. Also, using Lemma 2, we replace $x_{b,7}^{(m-1)}$ by a linear approximation with probability $\frac{1}{2}(1+\frac{1}{2})$. Therefore, the resultant equation will hold with probability $\frac{1}{2}(1+\frac{1}{2})$. The indexes of the equations are derived by noticing that when making these substitutions we use the opposite set of indexes, since we are now in a different round. For example, when replacing $x_{c,0}^{(m-1)}$, we use Eq. (6), which we can rewrite as

$$x_{c,0}^{(m-1)} = x_{\gamma,0}^{(m)} = x_{\delta,0}^{(m)} \oplus x_{\gamma,0}^{(m)} \oplus x_{\delta,8}^{(m)} \oplus x_{\alpha,0}^{(m)},$$

where $(\alpha,\beta,\gamma,\delta) \in \mathcal{Y}$ such that $\gamma = c$ when m is even, and $(\alpha,\beta,\gamma,\delta) \in \mathcal{X}$ such that $\gamma = c$ when m is odd. The cases for $x_{a,0}^{(m-1)}$ and $x_{b,7}^{(m-1)}$ are analogous. □

3.1 Improved Distinguishers

Using Lemma 5 we can derive a distinguisher for 5 rounds of ChaCha using the attack described in Section 2.2. To do that, we need to find an input differential $\Delta X^{(0)}$ with high probability, i.e.

$$\Pr\left(\Delta x_{a,0}^{(3)} \oplus \Delta x_{b,0}^{(3)} = 0 | \Delta X^{(0)}\right) = \frac{1}{2}(1+\varepsilon_d),$$

where $(a,b) \in \{(0,5),(1,6),(2,7),(3,4)\}$. Like in previous works [1, 5], we found this differential empirically by testing all possible single bit input differentials for each output possibility. The best bias we found was $\varepsilon_d = 0.00048$ for $(a,b) = (3,4)$ when the input differential is given by $\Delta x_{14,6}^{(0)} = 1$, and 0 for all remaining bits. Therefore, from Lemma 5 and Section 2.2 we get that

$$\begin{aligned}\Pr(&\Delta x_{1,0}^{(5)} \oplus \Delta x_{3,0}^{(5)} \oplus \Delta x_{4,26}^{(5)} \oplus \Delta x_{7,7}^{(5)} \oplus \Delta x_{7,19}^{(5)} \oplus \\ &\Delta x_{8,7}^{(5)} \oplus \Delta x_{8,19}^{(5)} \oplus \Delta x_{9,0}^{(5)} \oplus \Delta x_{11,12}^{(5)} \oplus \Delta x_{12,6}^{(5)} \oplus \\ &\Delta x_{12,7}^{(5)} \oplus \Delta x_{13,0}^{(5)} \oplus \Delta x_{13,8}^{(5)} \oplus \Delta x_{15,0}^{(5)} = 0 | \Delta x_{14,6}^{(0)} = 1) = \tfrac{1}{2}(1+\gamma)\end{aligned} \tag{8}$$

where $\gamma = \varepsilon_d \varepsilon_L^2 \approx 0.00012$.

We tested this result empirically by randomly selecting initial values for ChaCha and executing 5 rounds for the initial matrix $X^{(0)}$ and for $X'^{(0)} = X^{(0)} \oplus \Delta X^{(0)}$, and then checking if the equation holds. We executed this procedure for $N = 2^{38}$ iterations and got an estimated value of $\gamma = 0.000117$. This lead to an attack against 5 rounds of ChaCha with complexity 2^{27}.

As the reader may have noticed, this complexity is higher than the attack for 5 rounds proposed in [5]. However, there is an advantage: the equations of Lemma 3 have many more terms than the equations of Lemma 5, this means that when expanding for 6 rounds we may have an advantage using Lemma 5. Indeed, as presented in Section 2.3, we can compute the aggregated bias from the expansion by counting the number of substitutions of each type, in the case of Lemma 3 we have $(x_a, 3)$, $(x_b, 5)$, $(x_c, 3)$ and $(x_d, 2)$. In Section 2.3 we had weights $4, 1, 2, 1$ for x_a, x_b, x_c and x_d, respectively, leading to $\varepsilon_L = \frac{1}{2^{26}}$. On the other hand, from Lemma 5 we have $(x_a, 0)$, $(x_b, 3)$, $(x_c, 3)$ e $(x_d, 3)$, with the same weights. Then, the linear bias is

$$\varepsilon_L = \frac{1}{2^{1+0\cdot4+3\cdot1+3\cdot2+3\cdot1}} = \frac{1}{2^{13}},$$

therefore we have $\varepsilon_d \varepsilon_L^2 \approx 2^{-37.02}$, which leads to an attack against 6 rounds of ChaCha with complexity 2^{75} which is the currently best known 6 round attack on ChaCha.

4 Improved Attacks using Probabilistic Neutral Bits (PNBs)

The only cryptanalytic attack known for reduced round ChaCha is using the proposal of Aumasson [1]. The attack first identify good choices of truncated differentials, then it uses probabilistic backwards computation with the notion of PNBs, finally it estimate the complexity of the attack. This attack is described in several previous works [1, 16, 15], thus in our description we skip several details.

Let $\Delta_i^{(R)}$ be the differential for the i_{th} word of state matrix $X^{(R)}$, thus $\Delta_i^{(R)} = x_i^{(R)} \oplus x'^{(R)}_i$ and let $\Delta_{i,j}^{(R)}$ be the differential for the j_{th} bit of the i_{th} word, thus $\Delta_{i,j}^{(R)} = x_{i,j}^{(R)} \oplus x'^{(R)}_{i,j}$. In [1] the input differential $\mathcal{ID}$ is defined for a single-bit difference $\Delta_{i,j}^{(0)} = 1$ and consider a single-bit output difference $\mathcal{OD}$ after r rounds $\Delta_{p,q}^{(R)}$, such differential is denoted $\left(\Delta_{p,q}^{(R)} | \Delta_{i,j}^{(0)}\right)$. For a fixed key, the bias ε_d of the $\mathcal{OD}$ is defined by

$$\Pr_{v,t}\left(\Delta_{p,q}^{(R)} = 1 | \Delta_{i,j}^{(0)}\right) = \frac{1}{2}(1 + \varepsilon_d), \tag{9}$$

where the probability holds over all nonces v and counters t. Furthermore, considering the key as a random variable, we denote the median value of ε_d by $\varepsilon_d^\star$. Hence, for half of the keys, this differential have a bias of at least $\varepsilon_d^\star$.

Now, assume that the differential $\left(\Delta_{p,q}^{(r)} | \Delta_{i,j}^{(0)}\right)$ of bias ε_d is fixed, and we observe outputs Z and Z' of $R = l + r$ rounds for nonce v, counter t and unknown key k. If we guess the key k we can invert l rounds of the algorithm to get $X^{(r)}$ and $X'^{(r)}$ and compute $\Delta_{p,q}^{(r)}$, let call f the function which executes this procedure. Hence, $f(k, v, t, Z, Z') = \Delta_{p,q}^{(r)}$. From Eq. (9), we expect that

$$\Pr(f(\hat{k}, v, t, Z, Z') = 1) = \begin{cases} \frac{1}{2}(1 + \varepsilon_d), & \text{if } \hat{k} = k \\ 0.5, & \text{if } \hat{k} \neq k \end{cases},$$

thus, if we have several pairs of Z and Z', its possible to test our guesses for k.

Obviously, it is not useful to test all keys since this attack would be slower than exhaustive search, but we can search only over a subkey of $m = 256 - n$

bits, provided we can find a function g that is an approximation of f but only uses m key bits as input. Then, let $\bar{k}$ correspond to the subkey of m bits of key k and let f to be correlated to g with bias ε_a i.e.:

$$\Pr(f(k,v,t,Z,Z') = g(\bar{k},v,t,Z,Z')) = \frac{1}{2}(1+\varepsilon_a). \tag{10}$$

Denote the bias of g by ε, i.e. $\Pr(g(\bar{k},v,t,Z,Z') = 1) = \frac{1}{2}(1+\varepsilon)$, and $\varepsilon^\star$ the median bias of g over all keys, we can approximate ε by $\varepsilon_d\varepsilon_a$.

The problem that remains is how to efficiently find such a function g. In [1], this is done by first identifying key bits that have little influence on the result of $f(k,v,t,Z,Z')$, these are called *probabilistic neutral bits* (PNBs). This is done by defining the *neutrality measure* of a key bit $k_{i,j}$:

Definition 1. *The neutrality measure of the key bit $k_{i,j}$ with respect to the function $f(k,v,t,Z,Z')$ is defined as $\gamma_{i,j}$, where*

$$Pr(f(k,v,t,Z,Z') = f(k^*,v,t,Z,Z')) = \frac{1}{2}(1+\gamma_i)$$

and k^* is computed from key k by inverting the bit $k_{i,j}$, over all possible values of k, v and t.

After computing $\gamma_{i,j}$, for all $i = (0,1,...,7)$ and $j = (0,1,...,31)$, we can define the the set of significant key bits as $\Psi = \{(i,j) : \gamma_{i,j} \leq \gamma\}$ where γ is a threshold value, and then define our approximation g as $g(k_\Psi,v,t,Z,Z') = f(k^*,v,t,Z,Z')$ where k_Ψ is defined as the subkey with key bits in the set Ψ and k^* is computed from k_Ψ by setting $k_{i,j} = 0$ for all $(i,j) \notin \Psi$. Thus, the attack can be evaluated with the following steps:

1. Compute a good differential for r rounds $\left(\Delta^{(r)}_{p,q}|\Delta^{(0)}_{i,j}\right)$ by estimating the bias ε_d for all single-bit $\mathcal{ID}$ with several random combinations of keys, nonces, and counters.

2. Empirically estimate the neutrality measure $\gamma_{r,s}$ for each key bit $k_{r,s}$.

3. Construct the function g by setting all key bit such that $\gamma_{r,s} > \gamma$ to zero and estimate the median bias $\varepsilon^\star$ by empirically measuring bias of g using many randomly chosen keys, nonces, and counters.

4. Estimate the data and time complexity of the attack.

In the following we show that the linear approximations presented in Section 3 actually improves the attacks presented in [5]. This happens, because the fewer terms in the approximations leads to attacks with many more PNBs, hence, improving efficiency.

4.1 ChaCha20/6

For the 6-round cryptanalysis, as in [5], we run each experiment for 2^{34} randomly chosen IV's to get the average and then go for 256 such runs to obtain the median values. We used the linear approximations from the attack presented in Section 3. First, consider

$$\left(\Delta_{3,0}^{(4)} \oplus \Delta_{4,7}^{(4)} \oplus \Delta_{9,0}^{(4)} | \Delta_{14,6}^{(0)}\right).$$

Here go forward 4 rounds and come back 2 rounds. Using $\gamma = 0.4$ we have 210 PNBs, and we obtained $\varepsilon_a = 0.000127$. From that, since we have $\varepsilon_d = 0.00048$ from Section 3, we get an attack with data complexity of 2^{56} and time complexity $2^{102.2}$.

We also computed the complexity for the attack using the same $\mathcal{ID}$ but using the 5 round approximation of Eq. (8). Here go forward 5 rounds and come back 1 round. Using $\gamma = 0.4$ we have 212 PNBs, and we obtained $\varepsilon_a = 0.000107$. From that, since we have $\varepsilon_d = 0.00012$ (here we have to take the bias ε_L into account), we get an attack with data complexity of 2^{61} and time complexity $2^{104.68}$.

4.2 ChaCha20/7

For the 7-round cryptanalysis, we used the same linear approximations. First, consider

$$\left(\Delta_{3,0}^{(4)} \oplus \Delta_{4,7}^{(4)} \oplus \Delta_{9,0}^{(4)} | \Delta_{14,6}^{(0)}\right).$$

Here go forward 4 rounds and come back 3 rounds. Using $\gamma = 0.35$ we have 74 PNBs, and we obtained $\varepsilon_a = 0.000567$. From that, we get an attack with data complexity of 2^{50} and time complexity $2^{231.92}$. We also computed the complexity for the attack using the same $\mathcal{ID}$ but using the 5 round approximation of Eq. (8). Here go forward 5 rounds and come back 2 rounds. Using $\gamma = 0.35$ we have 77 PNBs, and we obtained $\varepsilon_a = 0.000319$. Therefore, get an attack with data complexity of 2^{56} and time complexity $2^{234.45}$.

5 Conclusion

In this paper, we improve the theoretical results of Choudhuri and Maitra by showing new linear approximations and improving the attacks against ChaCha reduced to 6 and 7 rounds. The improvement is counter-intuitive since the linear approximations presented have worst linear-differential bias than in previous works. Because of that, we get an attack against 5 rounds of Chacha with higher complexity. However, the proposed linear equations have fewer terms, which leads to fewer non-linear transitions when extending the attack one round further. Also, having fewer terms, the attack using PNBs are improved, since we are able to find many more neutral key bits.

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 *International Workshop on Fast Software Encryption*, pages 470–488. Springer, 2008.

[2] Daniel J Bernstein. The poly1305-aes message-authentication code. In *International Workshop on Fast Software Encryption*, pages 32–49. Springer, 2005.

[3] Daniel J Bernstein. Chacha, a variant of salsa20. In *Workshop Record of SASC*, volume 8, pages 3–5, 2008.

[4] Daniel J Bernstein. The salsa20 family of stream ciphers. In *New stream cipher designs*, pages 84–97. Springer, 2008.

[5] Arka Rai Choudhuri and Subhamoy Maitra. Significantly improved multi-bit differentials for reduced round salsa and chacha. *IACR Transactions on Symmetric Cryptology*, pages 261–287, 2016.

[6] Paul Crowley. Truncated differential cryptanalysis of five rounds of salsa20. *The State of the Art of Stream Ciphers SASC*, 2006:198–202, 2006.

[7] Sabyasachi Dey, Tapabrata Roy, and Santanu Sarkar. Revisiting design principles of salsa and chacha. *Advances in Mathematics of Communications*, 13(4), 2019.

[8] Sabyasachi Dey and Santanu Sarkar. Improved analysis for reduced round salsa and chacha. *Discrete Applied Mathematics*, 227:58–69, 2017.

[9] Lin Ding. Improved related-cipher attack on salsa20 stream cipher. *IEEE Access*, 7:30197–30202, 2019.

[10] Simon Fischer, Willi Meier, Côme Berbain, Jean-François Biasse, and Matthew JB Robshaw. Non-randomness in estream candidates salsa20 and tsc-4. In *International Conference on Cryptology in India*, pages 2–16. Springer, 2006.

[11] Julio Cesar Hernandez-Castro, Juan ME Tapiador, and Jean-Jacques Quisquater. On the salsa20 core function. In *International Workshop on Fast Software Encryption*, pages 462–469. Springer, 2008.

[12] IANIX. Chacha usage & deployment. `https://ianix.com/pub/chacha-deployment.html`, 2020. Accessed: 2020-01-13.

[13] Tsukasa Ishiguro, Shinsaku Kiyomoto, and Yutaka Miyake. Latin dances revisited: new analytic results of salsa20 and chacha. In *International Conference on Information and Communications Security*, pages 255–266. Springer, 2011.

[14] Adam Langley, W Chang, Nikos Mavrogiannopoulos, Joachim Strombergson, and Simon Josefsson. Chacha20-poly1305 cipher suites for transport layer security (tls). *RFC 7905*, (10), 2016.

[15] S Maitra, G Paul, and W Meier. Salsa20 cryptanalysis: New moves and revisiting old styles. wcc 2015. In *the Ninth International Workshop on Coding and Cryptography*, 2015.

[16] Subhamoy Maitra. Chosen iv cryptanalysis on reduced round chacha and salsa. *Discrete Applied Mathematics*, 208:88–97, 2016.

[17] Nicky Mouha and Bart Preneel. A proof that the arx cipher salsa20 is secure against differential cryptanalysis. *IACR Cryptology ePrint Archive*, 2013:328, 2013.

[18] Stephan Muller. Documentation and analysis of the linux random number generator - federal office for information security (germany's), 2019. `https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/LinuxRNG/LinuxRNG_EN.pdf;jsessionid=6B0F8D7795B80F5EADA3DB3DB3E4043B.1_cid360?__blob=publicationFile&v=19`.

[19] Matthew Robshaw and Olivier Billet. *New stream cipher designs: the eSTREAM finalists*, volume 4986. Springer, 2008.

[20] Zhenqing Shi, Bin Zhang, Dengguo Feng, and Wenling Wu. Improved key recovery attacks on reduced-round salsa20 and chacha. In *International Conference on Information Security and Cryptology*, pages 337–351. Springer, 2012.

[21] Linus Torvalds. Linux kernel source tree, 2016. `https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=818e607b57c94ade9824dad63a96c2ea6b21baf3`.

[22] Yukiyasu Tsunoo, Teruo Saito, Hiroyasu Kubo, Tomoyasu Suzaki, and Hiroki Nakashima. Differential cryptanalysis of salsa20/8. In *Workshop Record of SASC*, volume 28, 2007.

[23] Johan Wallén. Linear approximations of addition modulo 2 n. In *International Workshop on Fast Software Encryption*, pages 261–273. Springer, 2003.