IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2016, No. 2, pp. 261–287. DOI:10.13154/tosc.v2016.i2.261-287

Significantly Improved Multi-bit Differentials for Reduced Round Salsa and ChaCha*

Arka Rai Choudhuri[1] and Subhamoy Maitra[2]

[1] Department of Computer Science, Johns Hopkins University,
3400 N Charles St, Baltimore, MD 21218, USA
achoud@cs.jhu.edu
[2] Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, India
subho@isical.ac.in

Abstract. ChaCha and Salsa are two software oriented stream ciphers that have attracted serious attention in academic as well as commercial domain. The most important cryptanalysis of reduced versions of these ciphers was presented by Aumasson et al. in FSE 2008. One part of their attack was to apply input difference(s) to investigate biases after a few rounds. So far there have been certain kind of limited exhaustive searches to obtain such biases. For the first time, in this paper, we show how to theoretically choose the combinations of the output bits to obtain significantly improved biases. The main idea here is to consider the multi-bit differentials as extension of suitable single-bit differentials with linear approximations, which is essentially a differential-linear attack. As we consider combinations of many output bits (for example 19 for Salsa and 21 for ChaCha), exhaustive search is not possible here. By this method we obtain very high biases for linear combinations of bits in Salsa after 6 rounds and in ChaCha after 5 rounds. These are clearly two rounds of improvement for both the ciphers over the existing works. Using these biases we obtain several significantly improved cryptanalytic results for reduced round Salsa and ChaCha that could not b obtained earlier. In fact, with our results it is now possible to cryptanalyse 6-round Salsa and 5-round ChaCha in practical time.

Keywords: Stream Cipher · ChaCha · Salsa · Non-Randomness · Bias · Probabilistic Neutral Bit (PNB) · ARX Cipher · Differential-Linear Cryptanalysis

1 Introduction

Salsa20 [3] is a stream cipher designed by Bernstein in 2005 as a candidate for the eSTREAM competition [10]. The original proposal was for 20 rounds. The 12-round variant of Salsa20, Salsa20/12 was accepted into the final eSTREAM software portfolio. The ChaCha stream cipher [4], a variant of Salsa, was proposed in early 2008 to conjecturally provide better diffusion and cryptanalytic resistance over Salsa.

While ChaCha was designed some time back, the cipher has received renewed attention recently as the standardization process for inclusion of cipher suites based on ChaCha20-Poly1305 AEAD (ChaCha20 for symmetric encryption, the cipher is subjected to 20 rounds here and Poly1305 for authentication) in TLS1.3 is almost complete [23]. This in turn merits further analysis of both ChaCha and Salsa due to their similar structure.

*A significant portion of this work has been completed during the dissertation work (2015-2016) of the first author under the supervision of second author while the first author was a student of Master's program at the Indian Statistical Institute, Kolkata.

Received: 2016-06-01, Revised: 2016-09-01, Accepted: 2016-11-01, Published: 2017-02-03

Related work. Since their inception, both Salsa and ChaCha have undergone significant cryptanalysis [9, 11, 24, 2, 22, 12, 20, 18, 25, 17, 8] which show weaknesses in the reduced rounds of the ciphers. The attacks in most cases, apply some input differences to the initial state to observe output differences after certain rounds and once one can proceed a few rounds forward as above, it may be possible to invert a few rounds from a final state to obtain further non-randomness. The most important cryptanalysis in this regard was proposed by Aumasson et al. at FSE 2008 [2] with the introduction of Probabilistic Neutral Bits (PNBs). The work by Shi et al [22] introduced the concept of Column Chaining Distinguisher (CCD) to achieve some incremental advancements over [2] for both Salsa and ChaCha. Maitra, Paul and Meier [18] studied an interesting observation regarding round reversal of Salsa, but no significant cryptanalytic improvement could be obtained using this method. An important contribution of the authors in [18] is to correct some parameter values of [2] to obtain better attack complexity. Recently, Maitra [17] used a technique of Chosen IVs to obtain certain improvements over existing results. For this work we use the concept of differential-linear cryptanalysis [13, 5] which follows the recent work by Leurent [16].

Additionally, there have been significant developments in the construction of ARX toolkits [14] with some successful applications [15], but these are yet to exploit Salsa and ChaCha.

Our Contribution. Existing attempts at cryptanalysis of Salsa and ChaCha have largely ignored the structure of the ciphers, instead choosing to treat them as a black box to obtain certain non-randomness (we also call them biases) after a few rounds. In this work, we study the structure of these ciphers to show, for the first time, how to theoretically choose combination of output bits to obtain significantly improved biases thus enabling differential-linear cryptanalysis. With these theoretical results, we use a limited search over the input differences to obtain the best possible biases known so far. To show the significance of our analysis, we present the first known biases for 5/6 rounds of Salsa and 4/4.5/5 rounds of ChaCha. Obtaining such multi-bit biases are not possible by exhaustive search. For example, we consider linear combination of 19-bits for 6 round of Salsa and that would require a $\binom{512}{19}$ search where each case requires significant amount of search effort to experimentally discover the biases. These lead us, for the first time, towards the realm of practical attacks in certain cases for these ciphers (till six rounds for Salsa and five rounds for ChaCha).

Our results explain the dual bit differentials reported by Aumasson et al. in [1] (technical report version of [2]), which we believe were found by exhaustive search. This is suggested by the authors in [2], "Unlike Salsa20, our exhaustive search showed no bias in 4-round ChaCha, be it with one, two, or three target output bits." Using our theoretical results, we indicate why their exhaustive searches for ChaCha did not yield any bias of significance. We substantiate our theoretical findings with experiments.

Finally, we revisit the claim in [2] that "Exploiting multi-bit differentials, does not improve efficiency either". We do so by presenting significantly improved attacks for the reduced round versions of Salsa (till seven rounds) and ChaCha (till six rounds) than existing results in the literature. We have summarized our findings along with the other significant attacks for comparison in Table 1. However, we agree that as the number of rounds increase the number of Probabilistic Neutral Bits (PNBs) fall rapidly in case of multi-bit differentials and thus the significance of our results reduces as the number of rounds increases which is reflected for the attack against 8-round Salsa and 7-round ChaCha, though we could manage slightly better results than the presently known ones [17]. We leave as future work the possibility of combining our results with other existing techniques to determine if further improvement is possible.

Table 1: Complexity of the attacks for different rounds in the reduced-round versions. The last row indicates result related to non-randomness. Salsa20/12 is of 12 rounds and ChaCha20 is of 20 rounds which are proposed by the designer.

Cipher	Round/Key length	Time	Data	Reference
Salsa	5/256	2^{165}	2^{6}	[9]
		2^{167}	2^{7}	[25]
		2^{55}	2^{10}	[22]
		2^{8}	2^{8}	This work
	6/256	2^{177}	2^{15}	[11]
		2^{73}	2^{16}	[22]
		2^{32}	2^{32}	This work
	7/256	2^{151}	2^{26}	[2]
		2^{148}	2^{24}	[22]
		2^{139}	2^{32}	This work
		2^{137}	2^{61}	This work
	8/256	2^{251}	2^{31}	[2]
		2^{250}	2^{27}	[22]
		$2^{245.5}$	2^{96}	[17]
		$2^{244.9}$	2^{96}	This work
ChaCha	4/256	2^{6}	2^{6}	This work
	4.5/256	2^{12}	2^{12}	This work
	5/256	2^{16}	2^{16}	This work
	6/256	2^{139}	2^{30}	[2]
		2^{136}	2^{28}	[22]
		2^{130}	2^{35}	This work
		$2^{127.5}$	$2^{37.5}$	This work
		2^{116}	2^{116}	This work
	7/256	2^{248}	2^{27}	[2]
		$2^{246.5}$	2^{27}	[22]
		$2^{238.9}$	2^{96}	[17]
		$2^{237.7}$	2^{96}	This work
		2^{233}	2^{28}	This work (non-randomness)

Organization of the paper. We give an overview of Salsa and ChaCha in Section 2. In Section 3, theoretical results are presented along with experiments regarding the new biases. We describe the implications of our newly discovered biases on the cryptanalysis of Salsa and ChaCha using Probabilistic Neutral Bits (PNBs) in Section 4. Finally, we conclude this paper in Section 5.

2 Specifications and Preliminaries

The notations to be used in this paper are presented in the Table 2.

Note that throughout this paper we use ε to denote the bias of an event, which is actually $2 \cdot \mathsf{Pr}(\mathsf{event}) - 1$.

Table 2: Notation

Notation	Description
X	the state matrix of the cipher of 16 words
$X^{(0)}$	initial state matrix
$X^{(R)}$	state matrix after application of R round functions
x_i	i^{th} word of the state matrix (words arranged in row major)
$x_i[j]$	j^{th} bit of i^{th} word
$x + y$	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'. $\Delta x = x \oplus x'$
$\varepsilon_{(x_1 \oplus \cdots \oplus x_m)}$	$2 \cdot \mathsf{Pr}[\Delta x_1 \oplus \cdots \oplus \Delta x_m = 0] - 1$

2.1 Salsa

The cipher state of 16 words, where each word is of 32 bits, can be represented as a 4×4 matrix. For Salsa, we have the following state matrix

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

The matrix on the right shows the initial configuration of the state that takes four predefined constants $c_0 =$ `0x61707865`, $c_1 =$ `0x3320646e`, $c_2 =$ `0x79622d32`, $c_3 =$ `0x6b206574` (totaling to 128 bits), 256-bit key $k_0, \ldots, k_7$, 64-bit nonce v_0, v_1 and 64-bit counter t_0, t_1. For the 128-bit version of Salsa, the key words are repeated twice and the constant values differ slightly. In this paper we consider the 256-bit version for all the experiments, similar ideas will work for the 128-bit version though. Further, we will refer to the nonce and counter words together as IV words.

For Salsa the round function consists of 4 simultaneous applications of the quarterround function. The quarterround functions is performed on the vector $\left(x_a^{(r)}, x_b^{(r)}, x_c^{(r)}, x_d^{(r)}\right)$ to update its values as defined below:

$$\left.\begin{array}{rcl} x_b^{(r+1)} & = & x_b^{(r)} \oplus ((x_a^{(r)} + x_d^{(r)}) \lll 7), \\ x_c^{(r+1)} & = & x_c^{(r)} \oplus ((x_b^{(r+1)} + x_a^{(r)}) \lll 9), \\ x_d^{(r+1)} & = & x_d^{(r)} \oplus ((x_c^{(r+1)} + x_b^{(r+1)}) \lll 13), \\ x_a^{(r+1)} & = & x_a^{(r)} \oplus ((x_d^{(r+1)} + x_c^{(r+1)}) \lll 18). \end{array}\right\} \quad (1)$$

In the odd number rounds, called columnrounds, quarterround is applied to the columns (x_0, x_4, x_8, x_{12}), (x_5, x_9, x_{13}, x_1), $(x_{10}, x_{14}, x_2, x_6)$, and $(x_{15}, x_3, x_7, x_{11})$. In the even rounds, called rowrounds, the quarterround is applied to the rows (x_0, x_1, x_2, x_3), (x_5, x_6, x_7, x_4), $(x_{10}, x_{11}, x_8, x_9)$ and $(x_{15}, x_{12}, x_{13}, x_{14})$. Finally, a keystream block of 16-words (or 512 bits) is obtained as $Z = X^{(0)} + X^{(R)}$, where "+" symbolizes wordwise addition modulo 2^{32}, and $X^{(R)} = \mathsf{round}^R(X^{(0)})$. For Salsa20, $R = 20$, but the accepted cipher into eSTREAM [10] software portfolio is Salsa20/12, where $R = 12$.

Each Salsa20 round is reversible as the state-transition operations are reversible, i.e., if $X^{(r+1)} = \mathsf{round}(X^{(r)})$, then $X^{(r)} = \mathsf{round}^{-1}(X^{(r+1)})$, where round^{-1} is the inverse of round.

The inverse of the quarterround function on the vector $\left(x_a^{(r+1)}, x_b^{(r+1)}, x_c^{(r+1)}, x_d^{(r+1)}\right)$ is defined as:

$$\left.\begin{array}{lllll} x_a^{(r)} & = x_a^{(r+1)} \oplus ((x_d^{(r+1)} & + & x_c^{(r+1)}) \lll 18), \\ x_d^{(r)} & = x_d^{(r+1)} \oplus ((x_c^{(r+1)} & + & x_b^{(r+1)}) \lll 13), \\ x_c^{(r)} & = x_c^{(r+1)} \oplus ((x_b^{(r+1)} & + & x_a^{(r)}) \lll 9), \\ x_b^{(r)} & = x_b^{(r+1)} \oplus ((x_a^{(r)} & + & x_d^{(r)}) \lll 7). \end{array}\right\} \tag{2}$$

2.2 ChaCha

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

Similar to Salsa, the rightmost matrix shows the initial state that takes four predefined constants $c_0, \ldots, c_3$ (similar to Salsa), 256-bit key $k_0, \ldots, k_7$, 32-bit block counter t_0 and 96-bit nonce v_0, v_1, v_2. Here the quarterround function consists of four ARX rounds, each of which comprises of an addition (A), a cyclic left rotation (R) and an XOR (X) operation. The quarterround function on the vector $\left(x_a^{(r)}, x_b^{(r)}, x_c^{(r)}, x_d^{(r)}\right)$:

$$\left.\begin{array}{l} x_{a'}^{(r)} = x_a^{(r)} + x_b^{(r)}; \quad x_{d'}^{(r)} = x_d^{(r)} \oplus x_{a'}^{(r)}; \quad x_{d''}^{(r)} = x_{d'}^{(r)} \lll 16; \\ x_{c'}^{(r)} = x_c^{(r)} + x_{d''}^{(r)}; \quad x_{b'}^{(r)} = x_b^{(r)} \oplus x_{c'}^{(r)}; \quad x_{b''}^{(r)} = x_{b'}^{(r)} \lll 12; \\ x_a^{(r+1)} = x_{a'}^{(r)} + x_{b''}^{(r)}; \quad x_{d'''}^{(r)} = x_{d''}^{(r)} \oplus x_a^{(r+1)}; \quad x_d^{(r+1)} = x_{d'''}^{(r)} \lll 8; \\ x_c^{(r+1)} = x_{c'}^{(r)} + x_d^{(r+1)}; \quad x_{b'''}^{(r)} = x_{b''}^{(r)} \oplus x_c^{(r+1)}; \quad x_b^{(r+1)} = x_{b'''}^{(r)} \lll 7; \end{array}\right\} \tag{3}$$

In each of the odd rounds, called columnround, we apply quarterround to the four columns (x_0, x_4, x_8, x_{12}), (x_1, x_5, x_9, x_{13}), $(x_2, x_6, x_{10}, x_{14})$, and $(x_3, x_7, x_{11}, x_{15})$. In each of the even rounds, called diagonalround, we apply quarterround to the diagonals $(x_0, x_5, x_{10}, x_{15})$, $(x_1, x_6, x_{11}, x_{12})$, (x_2, x_7, x_8, x_{13}), and (x_3, x_4, x_9, x_{14}). As before, we define $X^{(R)} = \mathsf{round}^R(X^{(0)})$, and the keystream block $Z = X^{(0)} + X^{(R)}$. For ChaCha20, $R = 20$. As with Salsa, each round of ChaCha is reversible.

2.3 Differential

Given two states $X^{(r)}, X'^{(r)}$, we denote the differential of individual words by $\Delta x_i^{(r)} = x_i^{(r)} \oplus x_i'^{(r)}$. For example, '$\Delta x_{13}^{(0)} = 2^5$' means that we have two initial states $X^{(0)}, X'^{(0)}$ that differ at the 5^{th} bit of the 13^{th} word.

From the perspective of cryptanalysis, we are interested in introducing a difference at the initial state (call it Input Differential or $\mathcal{ID}$) and then attempt to obtain certain biases corresponding to combinations of some output bits (call it Output Differential or $\mathcal{OD}$). In this direction, one can compute $\mathsf{Pr}(\Delta x_p^{(r)}[q] = 0 | \Delta x_i^{(0)} = 2^j) = \frac{1}{2}(1 + \varepsilon_d)$, where the probability is estimated for a fixed key and all possible choices of nonces and counter words, other than the constraints imposed due to the input differences. Here, the bias is denoted by ε_d. In fact, one can consider a more general scenario as

$$\mathsf{Pr}\left[\left(\bigoplus_u \Delta x_{p_u}^{(r)}[q_u]\right) = 0 | \Delta x_{i_0}^{(0)} = 2^{j_{00}} + 2^{j_{01}} + \ldots, \Delta x_{i_1}^{(0)} = 2^{j_{10}} + 2^{j_{11}} + \ldots, \cdots\right] = \frac{1}{2}(1 + \varepsilon_d),$$

where one may observe the biases of certain linear combination of output differences given the input differences at one or more positions.

2.4 Differential-linear analysis

Differential-linear cryptanalysis was discovered by [13, 5] and has since been used for the cryptanalysis of various ciphers. We follow the recent work by Leurent [16] (without the partitioning of data). To explain its working we follow the heuristic analysis of [5], and for simplicity we denote

$$\Delta\sigma = \sigma \oplus \sigma' = \left(\bigoplus_u \Delta x^{(r)}_{p_u}[q_u]\right),\ \rho = \left(\bigoplus_v x^{(R)}_{p_v}[q_v]\right),\ \rho' = \left(\bigoplus_v x'^{(R)}_{p_v}[q_v]\right),$$

where $R > r$. Since the input difference is implicit, in this section, we don't specify it separately. We rewrite the differential bias and the linear approximation as follows, $\mathsf{Pr}\left[\Delta\sigma = 0\right] = \mathsf{Pr}\left[\sigma \oplus \sigma' = 0\right] = \frac{1}{2}(1+\varepsilon_d)$, $\mathsf{Pr}\left[\sigma = \rho\right] = \frac{1}{2}(1+\varepsilon_L)$. Given these, we want to find the bias γ such that

$$\mathsf{Pr}\left[\Delta\rho = 0\right] = \mathsf{Pr}\left[\rho \oplus \rho' = 0\right] = \frac{1}{2}(1+\gamma).$$

Now

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

and,

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

Hence the differential-linear bias is $\varepsilon_d \cdot \varepsilon_L^2$. This leads to a distinguisher of complexity $\mathcal{O}\left(\frac{1}{\varepsilon_d^2\cdot\varepsilon_L^4}\right)$.

Generally we require $\mathcal{O}(\frac{1}{pq^2})$ samples when we like to distinguish between two events, one with probability p and the other with probability $p(1+q)$, where q is small. For a detailed understanding on such complexity estimates, one may refer to [6, 7, 21]. In case of differential-linear cryptanalysis, for each sample, we require two executions of the cipher, both with same key, but different IVs. Thus, the actual complexity should be multiplied by two. However, at the time of comparison, we do not multiply the complexity estimates by two in our results, as it is not done in existing works too.

3 Differential-linear Biases

In this section we develop the theory for selecting specific combination of bits to give high biases and experimentally substantiate our findings. The main contributions of this section can be summarized as:

- Develop theory for selecting combination of output bits to obtain significant biases in Salsa and ChaCha.
- Improve on the best known biases for 4 rounds of Salsa and 3 rounds of ChaCha.
- First reported biases for 5/6 rounds of Salsa and 4/4.5/5 rounds of ChaCha.

In Appendix A, we demonstrate experimentally that these theoretical results extend to second-order (and higher) differentials.

Due to the slight differences, specifically in the number of updates in a round of Salsa and ChaCha, we deal with the ciphers differently. But importantly, the underlying idea is the same in both the cases.

3.1 Linear approximations with $\varepsilon_L = 1$

In this subsection we shall deal with the cases where the linear approximations hold with probability 1. Hence, in this case, $\mathsf{Pr}[\Delta\rho = 0] = \frac{1}{2}(1 + \varepsilon_d)$.

3.1.1 Salsa

We start with Salsa as the update functions are easier to handle than those of ChaCha.

Triple bit

Once m rounds of Salsa are run with an input difference, we have the output differences of the state $X^{(m)}$. The idea is to look for linear operations from the updates that primarily involve differentials from the m^{th} round. The structure of Salsa allow for us to get these combinations directly from the quarterround updates. We detail this procedure using the following lemma and proof.

Lemma 1. *Let us define* $\Delta Y^{(m)} = \Delta x_\alpha^{(m)}[13] \oplus \Delta x_\beta^{(m)}[0] \oplus \Delta x_\gamma^{(m)}[0]$ *and* $\Delta Y'^{(m)} = \Delta x_{\alpha'}^{(m)}[18] \oplus \Delta x_{\beta'}^{(m)}[0] \oplus \Delta x_{\gamma'}^{(m)}[0]$. *Then, after m rounds of Salsa, the following holds :*

$$\left|\varepsilon_{(Y^{(m)})}\right| = \left|\varepsilon_{(x_\alpha^{(m-1)}[13])}\right| \text{ and } \left|\varepsilon_{(Y'^{(m)})}\right| = \left|\varepsilon_{(x_{\alpha'}^{(m-1)}[18])}\right|.$$

The tuples (α, β, γ) *and* $(\alpha', \beta', \gamma')$ *vary depending on whether m is odd or even.*

- **Case I.** *m odd:* $(\alpha, \beta, \gamma) \in \{ (12, 4, 8), (1, 9, 13), (6, 14, 2), (11, 3, 7) \}$, $(\alpha', \beta', \gamma') \in \{ (0, 8, 12), (5, 13, 1), (10, 2, 6), (15, 7, 11) \}$
- **Case II.** *m even:* $(\alpha, \beta, \gamma) \in \{ (3, 1, 2), (4, 6, 7), (9, 11, 8), (14, 12, 13) \}$, $(\alpha', \beta', \gamma') \in \{ (0, 2, 3), (5, 7, 4), (10, 8, 9), (15, 13, 14) \}$

Proof. We will focus on two updates in the quarterround of the m^{th} round. Namely, $x_d^{(m)} = x_d^{(m-1)} \oplus ((x_c^{(m)} + x_b^{(m)}) \lll 13)$ and $x_a^{(m)} = x_a^{(m-1)} \oplus ((x_d^{(m)} + x_c^{(m)}) \lll 18)$ where a, b, c and d take values according to the described specifications for Salsa. Converting them to bit equations, $x_d^{(m)}[i+13] = x_d^{(m-1)}[i+13] \oplus x_c^{(m)}[i] \oplus x_b^{(m)}[i] \oplus C_{\mathsf{carry}}[i]$, $x_a^{(m)}[i+18] = x_a^{(m-1)}[i+18] \oplus x_d^{(m)}[i] \oplus x_c^{(m)}[i] \oplus C'_{\mathsf{carry}}[i]$. When $i = 0$, $C'_{\mathsf{carry}}[i] = C_{\mathsf{carry}}[i] = 0$ and the carry variables in this case will henceforth be ignored. Due to the linearity of the operations, the differential equations follow directly.

$$\Delta x_d^{(m)}[13] \oplus \Delta x_c^{(m)}[0] \oplus \Delta x_b^{(m)}[0] = \Delta x_d^{(m-1)}[13],$$
$$\Delta x_a^{(m)}[18] \oplus \Delta x_d^{(m)}[0] \oplus \Delta x_c^{(m)}[0] = \Delta x_a^{(m-1)}[18].$$

Since we are interested solely with the bias, the corresponding bias equations are, $\varepsilon_{(x_d^{(m)}[13] \oplus x_c^{(m)}[0] \oplus x_b^{(m)}[0])} = \varepsilon_{(x_d^{(m-1)}[13])}$, $\varepsilon_{(x_a^{(m)}[18] \oplus x_d^{(m)}[0] \oplus x_c^{(m)}[0])} = \varepsilon_{(x_a^{(m-1)}[18])}$. Further, taking the absolute value of the bias,

$$\left|\varepsilon_{(x_d^{(m)}[13] \oplus x_c^{(m)}[0] \oplus x_b^{(m)}[0])}\right| = \left|\varepsilon_{(x_d^{(m-1)}[13])}\right|,$$
$$\left|\varepsilon_{(x_a^{(m)}[18] \oplus x_d^{(m)}[0] \oplus x_c^{(m)}[0])}\right| = \left|\varepsilon_{(x_a^{(m-1)}[18])}\right|.$$

To obtain the tuples (α, β, γ) and $(\alpha', \beta', \gamma')$, we replace a, b, c and d by their relevant values. □

In essence, the lemma states that we are able to find a linear approximation of an active bit (in certain positions) from round m to round $m+1$ with probability 1. This gives rise to previously unknown biases for 5 rounds of Salsa, and substantially higher biases for the 4 rounds. There have been previously reported biases for 5 rounds of Salsa in [11, 18]. However, those biases require differences in the key-bits in addition to the IV bits, and hence cannot be considered for cryptanalytic attacks. This is because the attack model only accepts differences in IV's, given that there will be no control over the secret key bits.

We have presented some of the 4 round biases in Table 8 (in Appendix B) and 5 round biases in Table 3. These are experimental results and we emphasize that, in this section, we generally study the biases which are at least 0.01. Thus, it is enough for us to estimate the biases as the average over 2^{20} randomly chosen keys and IV's during experimentation. However, when the biases become low for higher rounds, then we increase the experimental runs and report them explicitly. For our limited search, we search over all 128 single bit input differences in the IV to obtain significant biases in the specific multi-bit combination. For a specific single input difference, we look at only 8 possible (see Lemma 1) 3 bit combinations in the output state to see which of them give the highest bias (8 instead of $\binom{512}{3}$).

Table 3: Best triple bit differentials for 5 rounds of Salsa

$\mathcal{ID}$	$\mathcal{OD}$	Bias
$\Delta x_7^{(0)} = 2^0$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	-0.1142
$\Delta x_8^{(0)} = 2^0$	$\Delta x_2^{(5)}[0] \oplus \Delta x_{14}^{(5)}[0] \oplus \Delta x_6^{(5)}[13]$	-0.0982
$\Delta x_8^{(0)} = 2^7$	$\Delta x_2^{(5)}[0] \oplus \Delta x_{14}^{(5)}[0] \oplus \Delta x_6^{(5)}[13]$	-0.0758
$\Delta x_8^{(0)} = 2^{27}$	$\Delta x_3^{(5)}[0] \oplus \Delta x_7^{(5)}[0] \oplus \Delta x_{11}^{(5)}[13]$	-0.0613
$\Delta x_8^{(0)} = 2^{30}$	$\Delta x_2^{(5)}[0] \oplus \Delta x_{14}^{(5)}[0] \oplus \Delta x_6^{(5)}[13]$	0.0583
$\Delta x_7^{(0)} = 2^{30}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.0559
$\Delta x_7^{(0)} = 2^{18}$	$\Delta x_2^{(5)}[0] \oplus \Delta x_{14}^{(5)}[0] \oplus \Delta x_6^{(5)}[13]$	0.0512
$\Delta x_7^{(0)} = 2^{27}$	$\Delta x_2^{(5)}[0] \oplus \Delta x_{14}^{(5)}[0] \oplus \Delta x_6^{(5)}[13]$	-0.0507

Now consider the input and output difference as explained in the first row of Table 3. The input difference is $\Delta x_7^{(0)}[0]$, the output difference is $\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$ and the bias ϵ_d has been observed as -0.1142. This bias after 5 rounds immediately provides a distinguisher with time and data complexity $\frac{1}{\frac{1}{2} \cdot \epsilon_d^2} < 154$. That is with 2^8 samples, it is enough to distinguish 5-round Salsa from a uniform random source.

These sets of 3 multi-bits can be further combined, but care should be taken with regards to the independence assumptions. In our case, we use independence assumptions in a very limited setting.

Dual bit

We have seen how we can express a single active bit from a round of Salsa as a linear combination of 3 bits from the next round. One naturally asks if it would work for combinations of two output bits. The answer for the dual bits is less favourable. We shall see that we can combine certain combinations of two output bits to get some reduction in bias from previous round, but in practice these do not yield useful results beyond round 4.

As before, we have the following lemma.

Lemma 2. *Let* $\Delta Y^{(m)} = \Delta x_\alpha^{(m)}[9] \oplus \Delta x_\beta^{(m)}[0]$ *and* $\Delta Y'^{(m-1)} = \Delta x_\alpha^{(m-1)}[9] \oplus \Delta x_\gamma^{(m-1)}[0]$.

After m rounds of Salsa, the following is true

$$\left|\varepsilon_{(Y^{(m)})}\right| = \left|\varepsilon_{(Y'^{(m-1)})}\right|. \tag{4}$$

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

- **Case I.** *m odd:* $(\alpha, \beta, \gamma) \in \{\ (8,4,0),\ (13,9,5),\ (2,14,10),\ (7,3,15)\ \}$
- **Case II.** *m even:* $(\alpha, \beta, \gamma) \in \{\ (2,1,0),\ (7,6,5),\ (8,11,10),\ (13,12,15)\ \}$

Proof. We use the following quarterround update: $x_c^{(m)} = x_c^{(m-1)} \oplus ((x_b^{(m)} + x_a^{(m)}) \lll 9)$. The remaining proof follows similar to the previous lemma. □

In itself, the lemma might not seem useful. If we need to apply the Piling-Up Lemma [19] to $\Delta Y'^{(m-1)}$, we require $\Delta x_\alpha^{(m-1)}[9]$ and $\Delta x_\gamma^{(m-1)}[0]$ to be independent. While this does not hold theoretically, for practical purposes it seems to hold good. Hence, we can rewrite (4) as

$$\left|\varepsilon_{(Y^{(m)})}\right| \approx \left|\varepsilon_{(x_\alpha^{(m-1)}[9])}\right| \cdot \left|\varepsilon_{(x_\gamma^{(m-1)}[0])}\right| \tag{5}$$

The fact that the assumption holds good for practical purposes is illustrated in Table 4, where, for notational convenience we denote $\Delta A = \Delta x_\alpha^{(m-1)}[9]$ and $\Delta B = \Delta x_\gamma^{(m-1)}[0]$.

Table 4: Best dual bit differentials for $m = 4$, characterized by α, β and i.

$\mathcal{ID}$	(α, β, γ)	$\varepsilon_{(Y^{(m)})}$	$\varepsilon_{(A)} \cdot \varepsilon_{(B)}$
$\Delta x_8^{(0)} = 2^{26}$	(7,6,5)	−0.6143	−0.6107
$\Delta x_7^{(0)} = 2^{26}$	(2,1,0)	−0.5708	−0.5684
$\Delta x_8^{(0)} = 2^{27}$	(7,6,5)	0.4677	0.4642
$\Delta x_7^{(0)} = 2^{27}$	(2,1,0)	0.4616	0.4584
$\Delta x_7^{(0)} = 2^{28}$	(2,1,0)	0.3201	0.3153
$\Delta x_8^{(0)} = 2^{28}$	(7,6,5)	0.3193	0.3146

The above observations explain the dual multi-bit differentials observed in [2], and also provide similar previously unobserved biases. To observe these dual bit differentials, we require two highly biased single bit differentials from the previous round. Due to the lack of such high single output bit differential biases starting with the 4^{th} round, we do not observe these dual bit biases starting from the 5^{th} round.

3.1.2 ChaCha

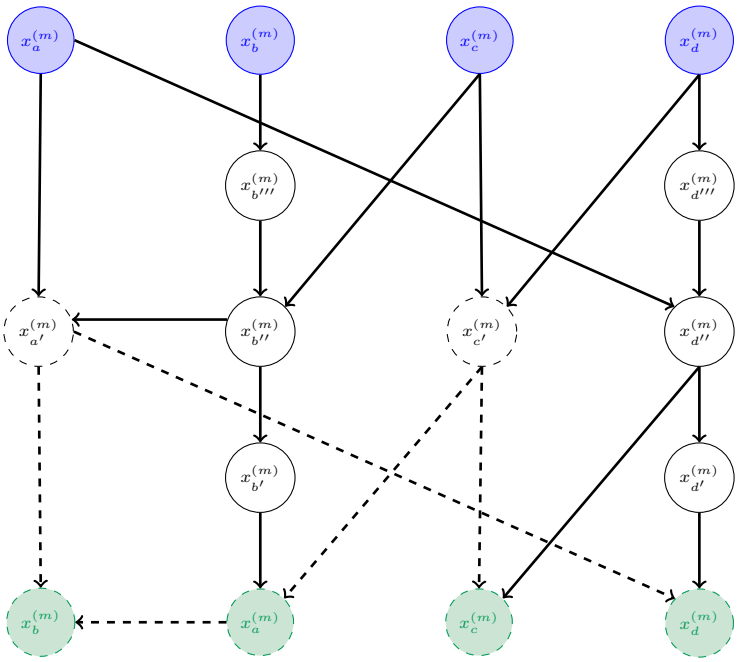
While the results pertaining ChaCha are in similar vein to those of Salsa, we follow a different path to obtain them. This is because of the increased number of operations in an update in ChaCha. First, we split each update of the ChaCha quarterround and write them

as bit equations only using the XOR operation,

$$\begin{aligned}
x_{a'}^{(m-1)}[i] &= x_a^{(m-1)}[i] \oplus x_b^{(m-1)}[i] \oplus C^1_{\mathsf{carry}}[i], \\
x_{d'}^{(m-1)}[i] &= x_{a'}^{(m-1)}[i] \oplus x_d^{(m-1)}[i],\ x_{d''}^{(m-1)}[i+16] = x_{d'}^{(m-1)}[i] \qquad (6)\\
x_{c'}^{(m-1)}[i] &= x_c^{(m-1)}[i] \oplus x_{d''}^{(m-1)}[i] \oplus C^2_{\mathsf{carry}}[i], \\
x_{b'}^{(m-1)}[i] &= x_{c'}^{(m-1)}[i] \oplus x_b^{(m-1)}[i],\ x_{b''}^{(m-1)}[i+12] = x_{b'}^{(m-1)}[i] \qquad (7)\\
x_a^{(m)}[i] &= x_{a'}^{(m-1)}[i] \oplus x_{b''}^{(m-1)}[i] \oplus C^3_{\mathsf{carry}}[i], \\
x_{d'''}^{(m-1)}[i] &= x_a^{(m)}[i] \oplus x_{d''}^{(m-1)}[i],\ x_d^{(m)}[i+8] = x_{d'''}^{(m-1)}[i] \qquad (8)\\
x_c^{(m)}[i] &= x_{c'}^{(m-1)}[i] \oplus x_d^{(m)}[i] \oplus C^4_{\mathsf{carry}}[i] \\
x_{b'''}^{(m-1)}[i] &= x_c^{(m)}[i] \oplus x_{b''}^{(m-1)}[i]\ x_b^{(m)}[i+7] = x_{b'''}^{(m-1)}[i] \qquad (9)
\end{aligned}$$

At round m we have access to the differentials of the variables of the form $x^{(m)}$. From them, as in Salsa, we would like to obtain partial or complete information of the differentials of the variables from round $m-1$.

We draw a derivation graph to indicate this. The nodes that contain variables which can be completely determined are indicated by a full circle, otherwise they are indicated by a dotted circle. Arrows leaving a dotted circle are also dotted to indicate they do not carry complete information of the variable. We trivially observe that none of the variables from round $m-1$ can be determined completely. For notational convenience we put only the variable into each node of the graph, but it is implicit that we are considering the differentials of these variables. Using the derivation graph and equations (6) to (9), we



express differentials of variables in round $m-1$ by differentials of variables in round m.

$$\begin{aligned}
\Delta x_b^{(m-1)}[i] &= \Delta x_{b'}^{(m-1)}[i] \oplus \Delta x_{c'}^{(m-1)}[i] \\
&= \Delta x_b^{(m)}[i+19] \oplus \Delta x_c^{(m)}[i+12] \oplus \Delta x_d^{(m)}[i] \oplus \Delta x_c^{(m)}[i] \oplus \Delta C^4_{\mathsf{carry}}[i] \qquad (10)
\end{aligned}$$

$$\begin{aligned}\Delta x_a^{(m-1)}[i] &= \Delta x_{a'}^{(m-1)}[i] \oplus \Delta x_b^{(m-1)}[i] \oplus \Delta C^1_{\mathsf{carry}}[i] \\ &= \Delta x_a^{(m)}[i] \oplus \Delta x_b^{(m)}[i+7] \oplus \Delta x_b^{(m)}[i+19] \oplus \Delta x_c^{(m)}[i+12] \\ &\oplus \Delta x_d^{(m)}[i] \oplus \Delta C^4_{\mathsf{carry}}[i] \oplus \Delta C^1_{\mathsf{carry}}[i] \oplus \Delta C^3_{\mathsf{carry}}[i] \end{aligned} \tag{11}$$

$$\begin{aligned}\Delta x_c^{(m-1)}[i] &= \Delta x_{c'}^{(m-1)}[i] \oplus \Delta x_{d''}^{(m-1)}[i] \oplus \Delta C^2_{\mathsf{carry}}[i] \\ &= \Delta x_d^{(m)}[i] \oplus \Delta x_c^{(m)}[i] \oplus \Delta x_d^{(m)}[i+8] \oplus \Delta x_a^{(m)}[i] \oplus \Delta C^2_{\mathsf{carry}}[i] \oplus \Delta C^4_{\mathsf{carry}}[i] \end{aligned} \tag{12}$$

$$\begin{aligned}\Delta x_d^{(m-1)}[i] &= \Delta x_{d'}^{(m-1)}[i] \oplus \Delta x_{a'}^{(m-1)}[i] \\ &= \Delta x_d^{(m)}[i+24] \oplus \Delta x_a^{(m)}[i+16] \oplus \Delta x_a^{(m)}[i] \oplus \Delta x_c^{(m)}[i] \oplus \Delta x_b^{(m)}[i+7] \oplus \Delta C^3_{\mathsf{carry}}[i] \end{aligned} \tag{13}$$

For $\varepsilon_L = 1$, the structure of ChaCha allows us to set $i = 0$ in each of the above equations for the carry to vanish completely. These observations lead to the following lemma, and the proof follows as before.

Lemma 3. *Let*

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

After m rounds of ChaCha*, the following holds:*

$$\left|\varepsilon_{(A^{(m)})}\right| = \left|\varepsilon_{(x_\alpha^{(m-1)}[0])}\right|, \quad \left|\varepsilon_{(B^{(m)})}\right| = \left|\varepsilon_{(x_\beta^{(m-1)}[0])}\right|,$$
$$\left|\varepsilon_{(C^{(m)})}\right| = \left|\varepsilon_{(x_\gamma^{(m-1)}[0])}\right|, \quad \left|\varepsilon_{(D^{(m)})}\right| = \left|\varepsilon_{(x_\delta^{(m-1)}[0])}\right|.$$

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

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

We demonstrate the experimental support of this lemma in Table 9 (in Appendix B) and table 5. For ChaCha, the lemma requires a combination of either 4 or 5 output bits for biases to propagate one round. This clarifies why the authors of [2] could not find significant biases for all combinations of two and three bits. A brute force search for all possible 4 or 5 bit combinations would be infeasible given the current computing resources. We use a limited search similar to the one described for Salsa.

Considering the first row of Table 5, we note $\epsilon_d = 0.1984$ and thus $\frac{1}{\frac{1}{2}\cdot\epsilon_d^2} < 51$. That is with 2^6 samples, it is enough to distinguish 4-round ChaCha from a uniform random source.

3.1.3 ChaCha Half Round

In this section, for the first time to our knowledge, we discuss the biases of a ChaCha half round. From (3) and equations (6) to (9), an important observation is that the equations (6)-(7) are almost the same as those of (8)-(9) with differences only in the

Table 5: Best multi-bit differentials for 4 rounds of ChaCha

$\mathcal{ID}$	$\mathcal{OD}$	Bias
$\Delta x_{12}^{(0)} = 2^{21}$	$\Delta x_2^{(4)}[0] \oplus \Delta x_7^{(4)}[7] \oplus \Delta x_7^{(4)}[19] \oplus \Delta x_8^{(4)}[12] \oplus \Delta x_{13}^{(4)}[0]$	0.1984
$\Delta x_{14}^{(0)} = 2^{21}$	$\Delta x_0^{(4)}[0] \oplus \Delta x_5^{(4)}[7] \oplus \Delta x_5^{(4)}[19] \oplus \Delta x_{10}^{(4)}[12] \oplus \Delta x_{15}^{(4)}[0]$	0.1979
$\Delta x_{15}^{(0)} = 2^{21}$	$\Delta x_1^{(4)}[0] \oplus \Delta x_6^{(4)}[7] \oplus \Delta x_6^{(4)}[19] \oplus \Delta x_{11}^{(4)}[12] \oplus \Delta x_{12}^{(4)}[0]$	0.1973
$\Delta x_{13}^{(0)} = 2^{21}$	$\Delta x_3^{(4)}[0] \oplus \Delta x_4^{(4)}[7] \oplus \Delta x_4^{(4)}[19] \oplus \Delta x_9^{(4)}[12] \oplus \Delta x_{14}^{(4)}[0]$	0.1972
$\Delta x_{13}^{(0)} = 2^{9}$	$\Delta x_3^{(4)}[0] \oplus \Delta x_4^{(4)}[7] \oplus \Delta x_4^{(4)}[19] \oplus \Delta x_9^{(4)}[12] \oplus \Delta x_{14}^{(4)}[0]$	−0.1427
$\Delta x_{14}^{(0)} = 2^{9}$	$\Delta x_0^{(4)}[0] \oplus \Delta x_5^{(4)}[7] \oplus \Delta x_5^{(4)}[19] \oplus \Delta x_{10}^{(4)}[12] \oplus \Delta x_{15}^{(4)}[0]$	−0.1424
$\Delta x_{12}^{(0)} = 2^{9}$	$\Delta x_2^{(4)}[0] \oplus \Delta x_7^{(4)}[7] \oplus \Delta x_7^{(4)}[19] \oplus \Delta x_8^{(4)}[12] \oplus \Delta x_{13}^{(4)}[0]$	−0.1419
$\Delta x_{15}^{(0)} = 2^{9}$	$\Delta x_1^{(4)}[0] \oplus \Delta x_6^{(4)}[7] \oplus \Delta x_6^{(4)}[19] \oplus \Delta x_{11}^{(4)}[12] \oplus \Delta x_{12}^{(4)}[0]$	−0.1417

numeric argument to the cyclic rotations. This allows us to split the quarterround into two (unequal) "halves".

We rewrite equations (6)-(7) as the first "half" (m to $m+0.5$) of a round of ChaCha as:

$$\begin{aligned} x_a^{(m+0.5)}[i] &= x_a^{(m)}[i] \oplus x_b^{(m)}[i] \oplus C^1_{\mathsf{carry}}[i] \\ x_{d'}^{(m)}[i] &= x_a^{(m+0.5)}[i] \oplus x_d^{(m)}[i] \\ x_d^{(m+0.5)}[i+16] &= x_{d'}^{(m)}[i] \end{aligned}$$

and

$$\begin{aligned} x_c^{(m+0.5)}[i] &= x_c^{(m)}[i] \oplus x_d^{(m+0.5)}[i] \oplus C^2_{\mathsf{carry}}[i] \\ x_{b'}^{(m)}[i] &= x_c^{(m+0.5)}[i] \oplus x_b^{(m)}[i] \\ x_b^{(m+0.5)}[i+12] &= x_{b'}^{(m)}[i] \end{aligned}$$

Using similar techniques to those discussed in the previous section, we represent bits of a variable of round m using bits from variables of round $m+0.5$ using the equations above.

$$x_a^{(m)}[i] = x_a^{(m+0.5)}[i] \oplus x_b^{(m+0.5)}[i+12] \oplus x_c^{(m+0.5)}[i] \oplus C^1_{\mathsf{carry}}[i] \tag{14}$$

$$x_b^{(m)}[i] = x_b^{(m+0.5)}[i+12] \oplus x_c^{(m+0.5)}[i] \tag{15}$$

$$x_c^{(m)}[i] = x_c^{(m+0.5)}[i] \oplus x_d^{(m+0.5)}[i] \oplus C^2_{\mathsf{carry}}[i] \tag{16}$$

$$x_d^{(m)}[i] = x_d^{(m+0.5)}[i+16] \oplus x_a^{(m+0.5)}[i] \tag{17}$$

Interestingly, we see that bias of variables $x_b^{(m)}[i]$ and $x_d^{(m)}[i]$ can be derived from round $m+0.5$ without any reduction in value for all i. This follows from the fact that there are no carry bits in equations (15) and (17). We shall exploit this property of the half round, along with the results discussed in the previous section to push biases over 1.5 rounds.

Observe equation (12). When $i = 0$,

$$\Delta x_c^{(m-1)}[0] = \Delta x_d^{(m)}[0] \oplus \Delta x_c^{(m)}[0] \oplus \Delta x_d^{(m)}[8] \oplus \Delta x_a^{(m)}[0]$$

only one term on the right side of the equation has a non-zero bit position. Similarly, if we consider the other equations (11), (10) and (13) with $i = 0$, each of them have more than one term with non-zero bit positions[1]. But of the 4 equations, only (12) has non-zero bit positions limited to the b and d variables. This allows us to bypass the need to deal with carry differentials for the "half" round.

[1]When $i > 0$, each of the mentioned equations already have a carry bit involved, and hence not of interest to us.

As consecutive rounds are different, the roles of variables a, b, c and d change over these two rounds. This leads to complications, but for simplicity, we illustrate our method using an example.

Consider $\Delta x_8^{(3)}[0]$. Since x_8 always takes the value of variable c in the update operations, we shall illustrate how we move this bias without diminishing its value over 1.5 rounds using equations (12), (14), (16) and (17). Care must be taken to consider updates from the even round (3 to 4) and half an odd round (4 to 4.5).

$$\begin{aligned}
\Delta x_8^{(3)}[0] &= \Delta x_2^{(4)}[0] \oplus \Delta x_8^{(4)}[0] \oplus \Delta x_{13}^{(4)}[8] \oplus \Delta x_{13}^{(4)}[0] \\
&= \left(\Delta x_2^{(4.5)}[0] \oplus \Delta x_6^{(4.5)}[12] \oplus \Delta x_{10}^{(4.5)}[0]\right) \oplus \left(\Delta x_8^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[0]\right) \\
&\oplus \left(\Delta x_1^{(4.5)}[0] \oplus \Delta x_{13}^{(4.5)}[16]\right) \oplus \left(\Delta x_1^{(4.5)}[8] \oplus \Delta x_{13}^{(4.5)}[24]\right) \\
&= \Delta x_1^{(4.5)}[0] \oplus \Delta x_1^{(4.5)}[8] \oplus \Delta x_2^{(4.5)}[0] \oplus \Delta x_6^{(4.5)}[12] \oplus \Delta x_8^{(4.5)}[0] \\
&\oplus \Delta x_{10}^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[0] \oplus \Delta x_{13}^{(4.5)}[16] \oplus \Delta x_{13}^{(4.5)}[24]
\end{aligned}$$

Similar equations can be defined for x_9, x_{10} and x_{11} (all variable c in the update operations). Further, using a limited search over the input differences, the best obtained 4.5-round multi bit differentials are presented in Table 10 in Appendix B. Essentially, we have now been able to move certain linear approximations across 1.5 rounds without any reduction in bias.

Following Table 10, we consider the input difference $\Delta x_{13}^{(0)}[13]$ and the output difference $\Delta x_0^{(4.5)}[0] \oplus \Delta x_0^{(4.5)}[8] \oplus \Delta x_1^{(4.5)}[0] \oplus \Delta x_5^{(4.5)}[12] \oplus \Delta x_{11}^{(4.5)}[0] \oplus \Delta x_9^{(4.5)}[0] \oplus \Delta x_{15}^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[16] \oplus \Delta x_{12}^{(4.5)}[24]$. The value of ϵ_d is 0.0282. This bias after 4.5 rounds provides a distinguisher with time and data complexity $\frac{1}{\frac{1}{2}\cdot\epsilon_d^2} < 2515$. That is with 2^{12} samples, it is enough to distinguish 4.5-round ChaCha from a uniform random source.

3.2 Linear Approximation with $\varepsilon_L < 1$

In this section we study the linear approximations which hold with probability < 1. The non linearity in Salsa and ChaCha arise solely from modular addition. We use some classical results of the linear approximation of addition (one may refer to [26] for more detailed analysis).

Lemma 4. *For the modular addition operation $s = a + b$, the approximation $s[i] = a[i] \oplus b[i] \oplus a[i-1]$ holds with probability $\frac{3}{4}$.*

Next we define a linear mask that will prove useful in the subsequent results.

Definition 1. Γ_i denotes a linear masking vector over $GF(2)$ which has 1 only in positions of i and $i+1$. Then, given 32-bit x, $x \cdot \Gamma_i = x[i+1] \oplus x[i]$ where $\cdot$ denotes the standard inner product.

Using this linear mask, the following result can be obtained.

Lemma 5. *Given $x, y \in \{0,1\}^{32}$, the following holds for $0 \leq i \leq 30$:*
$Pr[\Gamma_i \cdot (x+y) = x[i+1] \oplus y[i+1]] = \frac{1}{2}\left(1 - \frac{1}{2}\right)$.

The sketch of the proof can be found in Appendix C.1. For subtraction, a similar result is obtained and due to its similarity with the previous result, we skip the proof.

Lemma 6. *Given $x, y \in \{0,1\}^{32}$, the following holds for $0 \leq i \leq 30$:*
$Pr[\Gamma_i \cdot (x-y) = x[i+1] \oplus y[i+1]] = \frac{1}{2}\left(1 + \frac{1}{2}\right)$.

In the case of both Salsa and ChaCha, we consider linear approximations with one active input bit and multiple active output bits. The approximations follow from the structure of the ciphers and the above discussed results.

3.2.1 Salsa

We start with linear approximations over one round that hold with probability less than 1. This follows from the repeated application of the previous lemmas.

Lemma 7. *For one active input bit in round $m-1$ and multiple active output bits in round m, the following holds:*

$$\begin{aligned}
x_b^{(m-1)}[i+7] &= x_b^{(m)}[i+7] \oplus x_a^{(m)}[i] \oplus x_a^{(m)}[i-1] \oplus x_d^{(m)}[i-18] \oplus x_c^{(m)}[i-18] \oplus x_d^{(m)}[i] \\
&\quad \oplus x_c^{(m)}[i-13] \oplus x_b^{(m)}[i-13] \oplus x_b^{(m)}[i-14], \quad w.p.\ \frac{1}{2}\left(1-\frac{1}{2^3}\right) \\
x_c^{(m-1)}[i+9] &= x_c^{(m)}[i+9] \oplus x_b^{(m)}[i] \oplus x_a^{(m)}[i] \\
&\quad \oplus x_a^{(m)}[i-1] \oplus x_d^{(m)}[i-18] \oplus x_c^{(m)}[i-18], \quad w.p.\ \frac{1}{2}\left(1-\frac{1}{2^2}\right) \\
x_d^{(m-1)}[i+13] &= x_d^{(m)}[i+13] \oplus x_c^{(m)}[i] \oplus x_b^{(m)}[i] \oplus x_b^{(m)}[i-1], \quad w.p.\ \frac{1}{2}\left(1+\frac{1}{2}\right) \\
x_a^{(m-1)}[i+18] &= x_a^{(m)}[i+18] \oplus x_d^{(m)}[i] \oplus x_c^{(m)}[i] \oplus x_c^{(m)}[i-1], \quad w.p.\ \frac{1}{2}\left(1+\frac{1}{2}\right)
\end{aligned}$$

However, since this reduces the differential bias considerably in each case, the result by itself is not useful.

Experimentally, the most significant biases for 4 rounds of Salsa are in x_d, hence we limit our focus for linear approximations over 2 rounds to an input active bit in x_d. Using Lemma 1, we first apply a linear approximation to x_d to get an approximation with 3 bits that occur with probability 1. To each of the three bits we apply the above lemma to get an approximation over 2 rounds, m to $m+2$. Here we have considered the case for m even, while the case for m odd is symmetric in nature.

Lemma 8. *When m is even, each of the following holds with probability $\frac{1}{2}\left(1+\frac{1}{2^6}\right)$.*

$$\begin{aligned}
x_1^{(m)}[13] &= x_9^{(m+2)}[0] \oplus x_8^{(m+2)}[19] \oplus x_{11}^{(m+2)}[19] \oplus x_{11}^{(m+2)}[18] \oplus x_{13}^{(m+2)}[0] \oplus x_{12}^{(m+2)}[23] \\
&\quad \oplus x_{15}^{(m+2)}[23] \oplus x_{15}^{(m+2)}[22] \oplus x_{14}^{(m+2)}[5] \oplus x_{13}^{(m+2)}[5] \oplus x_1^{(m+2)}[13] \oplus x_0^{(m+2)}[6] \oplus x_0^{(m+2)}[5] \\
&\quad \oplus x_3^{(m+2)}[20] \oplus x_2^{(m+2)}[20] \oplus x_3^{(m+2)}[6] \oplus x_2^{(m+2)}[25] \oplus x_1^{(m+2)}[25] \oplus x_1^{(m+2)}[24],
\end{aligned}$$

$$\begin{aligned}
x_6^{(m)}[13] &= x_{14}^{(m+2)}[0] \oplus x_{13}^{(m+2)}[19] \oplus x_{12}^{(m+2)}[19] \oplus x_{12}^{(m+2)}[18] \oplus x_2^{(m+2)}[0] \oplus x_1^{(m+2)}[23] \\
&\quad \oplus x_0^{(m+2)}[23] \oplus x_0^{(m+2)}[22] \oplus x_3^{(m+2)}[5] \oplus x_2^{(m+2)}[5] \oplus x_6^{(m+2)}[13] \oplus x_5^{(m+2)}[6] \oplus x_5^{(m+2)}[5] \\
&\quad \oplus x_4^{(m+2)}[20] \oplus x_7^{(m+2)}[20] \oplus x_4^{(m+2)}[6] \oplus x_7^{(m+2)}[25] \oplus x_6^{(m+2)}[25] \oplus x_6^{(m+2)}[24],
\end{aligned}$$

$$\begin{aligned}
x_{11}^{(m)}[13] &= x_3^{(m+2)}[0] \oplus x_2^{(m+2)}[19] \oplus x_1^{(m+2)}[19] \oplus x_1^{(m+2)}[18] \oplus x_7^{(m+2)}[0] \oplus x_6^{(m+2)}[23] \\
&\quad \oplus x_5^{(m+2)}[23] \oplus x_5^{(m+2)}[22] \oplus x_4^{(m+2)}[5] \oplus x_7^{(m+2)}[5] \oplus x_{11}^{(m+2)}[13] \oplus x_{10}^{(m+2)}[6] \oplus x_{10}^{(m+2)}[5] \\
&\quad \oplus x_9^{(m+2)}[20] \oplus x_8^{(m+2)}[20] \oplus x_9^{(m+2)}[6] \oplus x_8^{(m+2)}[25] \oplus x_{11}^{(m+2)}[25] \oplus x_{11}^{(m+2)}[24],
\end{aligned}$$

$$\begin{aligned}
x_{12}^{(m)}[13] &= x_4^{(m+2)}[0] \oplus x_7^{(m+2)}[19] \oplus x_6^{(m+2)}[19] \oplus x_6^{(m+2)}[18] \oplus x_8^{(m+2)}[0] \oplus x_{11}^{(m+2)}[23] \\
&\quad \oplus x_{10}^{(m+2)}[23] \oplus x_{10}^{(m+2)}[22] \oplus x_9^{(m+2)}[5] \oplus x_8^{(m+2)}[5] \oplus x_{12}^{(m+2)}[13] \oplus x_{15}^{(m+2)}[6] \oplus x_{15}^{(m+2)}[5] \\
&\quad \oplus x_{14}^{(m+2)}[20] \oplus x_{13}^{(m+2)}[20] \oplus x_{14}^{(m+2)}[6] \oplus x_{13}^{(m+2)}[25] \oplus x_{12}^{(m+2)}[25] \oplus x_{12}^{(m+2)}[24].
\end{aligned}$$

Remark 1. We are interested in the case when $m=4$, and the above linear approximation has 1 active bit in round 4 and 19 active bits in round 6. The ε_d useful in this scenario is with $\mathcal{ID}$ at $x_7^{(0)}[0]$, $\mathcal{OD}$ at $x_1^{(4)}[13]$ and value $\varepsilon_d = -0.1142 \approx -\frac{1}{2^{3.13}}$. As we described in Section 3.1.1, this extends to 5-round differential-linear bias with $\varepsilon_L = 1$, which is available in the first row of Table 3. Now, for the 6-th round, we have $\varepsilon_L = \frac{1}{2^6}$. This gives a total

differential-linear 6^{th} round bias of $\varepsilon_d \cdot \varepsilon_L^2 \approx -\frac{1}{2^{15.13}}$. This leads to a six round distinguisher with complexity $\approx 2^{32}$. We ran experiments with 2^{42} randomly chosen key and IVs for the output difference with 19 bits, i.e., $\Delta x_0^{(6)}[6] \oplus \Delta x_0^{(6)}[5] \oplus \Delta x_1^{(6)}[13] \oplus \Delta x_1^{(6)}[25] \oplus \Delta x_1^{(6)}[24] \oplus \Delta x_2^{(6)}[25] \oplus \Delta x_2^{(6)}[20] \oplus \Delta x_3^{(6)}[6] \oplus \Delta x_3^{(6)}[20] \oplus \Delta x_{11}^{(6)}[19] \oplus \Delta x_{11}^{(6)}[18] \oplus \Delta x_8^{(6)}[19] \oplus \Delta x_9^{(6)}[0] \oplus \Delta x_{15}^{(6)}[23] \oplus \Delta x_{15}^{(6)}[22] \oplus \Delta x_{12}^{(6)}[23] \oplus \Delta x_{13}^{(6)}[0] \oplus \Delta x_{13}^{(6)}[5] \oplus \Delta x_{14}^{(6)}[5]$. The experimental bias -0.000028 exactly supports the probability as we described above.

We have briefly sketched the proof for one case each of Lemma 7 and Lemma 8 in Appendix C.2 and C.3.

This method becomes hard to handle for linear approximations with 3 rounds, but we can still obtain the value of the linear bias over 3 rounds. This is done by counting the number of variables of type x_a, x_b, x_c and x_d in an equation from Lemma 8. The counts in the form (Variable Type, # occurrences) are (x_a, 4), (x_b, 4), (x_c, 5), and (x_d, 6).

One should note that one of the x_d variables allow for linear approximation 1, and is hence discounted in further calculations. Following from Lemmas 7 and 8, we can calculate the probability of the bias for three rounds. The calculation leads to a linear bias over 3 rounds of value $\varepsilon_L = \frac{1}{2^{6+4\cdot1+4\cdot3+5\cdot2+5\cdot1}} = \frac{1}{2^{37}}$. For the ε_d previously considered, this leads to a 7 round bias of $\varepsilon_d \cdot \varepsilon_L^2 \approx \frac{1}{2^{95.13}}$. However, the distinguisher for this bias has a complexity of 2^{191} which is worse than the best known 7 round attack and further, for a similar reason this method is unlikely to work for 8 rounds of Salsa.

3.2.2 ChaCha

The method for ChaCha follows from the ideas used in Salsa.

Lemma 9. *For one active input bit in round $m-1$ and multiple active output bits in round m, the following holds.*

$x_b^{(m-1)}[i] = x_b^{(m)}[i+19] \oplus x_c^{(m)}[i+12] \oplus x_d^{(m)}[i] \oplus x_c^{(m)}[i] \oplus x_d^{(m)}[i-1]$ *w.p.* $\frac{1}{2}\left(1+\frac{1}{2}\right)$,

$x_a^{(m-1)}[i] = x_a^{(m)}[i] \oplus x_b^{(m)}[i+7] \oplus x_b^{(m)}[i+19] \oplus x_c^{(m)}[i+12] \oplus x_d^{(m)}[i] \oplus x_b^{(m)}[i+18] \oplus x_c^{(m)}[i+11] \oplus x_d^{(m)}[i-2] \oplus x_d^{(m)}[i+6]$ *w.p.* $\frac{1}{2}\left(1+\frac{1}{2^4}\right)$,

$x_c^{(m-1)}[i] = x_d^{(m)}[i] \oplus x_c^{(m)}[i] \oplus x_d^{(m)}[i+8] \oplus x_a^{(m)}[i] \oplus x_a^{(m)}[i-1] \oplus x_d^{(m)}[i+7] \oplus x_d^{(m)}[i-1]$ *w.p.* $\frac{1}{2}\left(1+\frac{1}{2^2}\right)$,

$x_d^{(m-1)}[i] = x_d^{(m)}[i+24] \oplus x_a^{(m)}[i+16] \oplus x_a^{(m)}[i] \oplus x_c^{(m)}[i] \oplus x_b^{(m)}[i+7] \oplus x_c^{(m)}[i-1] \oplus x_b^{(m)}[i+6]$ *w.p.* $\frac{1}{2}\left(1+\frac{1}{2}\right)$.

We follow the method used for finding biases for 4.5 rounds of ChaCha. But at this point we are unable to proceed further due to the restriction that linear approximations must hold with probability 1. We pick x_c as it has the only one non LSB term in Lemma 3. The LSB terms are approximated one round further using Lemma 3, but the other term uses the above lemma to derive the following result.

Lemma 10. *Each of the following holds with probability* $\frac{1}{2}\left(1+\frac{1}{2}\right)$.

$$\begin{aligned}
x_8^{(3)}[0] &= x_{13}^{(5)}[24] \oplus x_1^{(5)}[16] \oplus x_1^{(5)}[0] \oplus x_9^{(5)}[0] \oplus x_5^{(5)}[7] \\
&\oplus x_{12}^{(5)}[0] \oplus x_8^{(5)}[0] \oplus x_{12}^{(5)}[8] \oplus x_0^{(5)}[0] \oplus x_2^{(5)}[0] \oplus x_6^{(5)}[7] \\
&\oplus x_6^{(5)}[19] \oplus x_{10}^{(5)}[12] \oplus x_{14}^{(5)}[0] \oplus x_{13}^{(5)}[0] \oplus x_1^{(5)}[24] \oplus x_1^{(5)}[8] \\
&\oplus x_9^{(5)}[8] \oplus x_5^{(5)}[15] \oplus x_9^{(5)}[7] \oplus x_5^{(5)}[14] \\
x_9^{(3)}[0] &= x_{14}^{(5)}[24] \oplus x_2^{(5)}[16] \oplus x_2^{(5)}[0] \oplus x_{10}^{(5)}[0] \oplus x_6^{(5)}[7] \\
&\oplus x_{13}^{(5)}[0] \oplus x_9^{(5)}[0] \oplus x_{13}^{(5)}[8] \oplus x_1^{(5)}[0] \oplus x_3^{(5)}[0] \oplus x_7^{(5)}[7] \\
&\oplus x_7^{(5)}[19] \oplus x_{11}^{(5)}[12] \oplus x_{15}^{(5)}[0] \oplus x_{14}^{(5)}[0] \oplus x_2^{(5)}[24] \oplus x_2^{(5)}[8] \\
&\oplus x_{10}^{(5)}[8] \oplus x_6^{(5)}[15] \oplus x_{10}^{(5)}[7] \oplus x_6^{(5)}[14]
\end{aligned}$$

$$\begin{aligned}
x_{10}^{(3)}[0] &= x_{15}^{(5)}[24] \oplus x_3^{(5)}[16] \oplus x_3^{(5)}[0] \oplus x_{11}^{(5)}[0] \oplus x_7^{(5)}[7] \\
&\oplus x_{14}^{(5)}[0] \oplus x_{10}^{(5)}[0] \oplus x_{14}^{(5)}[8] \oplus x_2^{(5)}[0] \oplus x_0^{(5)}[0] \oplus x_4^{(5)}[7] \\
&\oplus x_4^{(5)}[19] \oplus x_8^{(5)}[12] \oplus x_{12}^{(5)}[0] \oplus x_{15}^{(5)}[0] \oplus x_3^{(5)}[24] \oplus x_3^{(5)}[8] \\
&\oplus x_{11}^{(5)}[8] \oplus x_7^{(5)}[15] \oplus x_{11}^{(5)}[7] \oplus x_7^{(5)}[14] \\
x_{11}^{(3)}[0] &= x_{12}^{(5)}[24] \oplus x_0^{(5)}[16] \oplus x_0^{(5)}[0] \oplus x_8^{(5)}[0] \oplus x_4^{(5)}[7] \\
&\oplus x_{15}^{(5)}[0] \oplus x_{11}^{(5)}[0] \oplus x_{15}^{(5)}[8] \oplus x_3^{(5)}[0] \oplus x_1^{(5)}[0] \oplus x_5^{(5)}[7] \\
&\oplus x_5^{(5)}[19] \oplus x_9^{(5)}[12] \oplus x_{13}^{(5)}[0] \oplus x_{12}^{(5)}[0] \oplus x_0^{(5)}[24] \oplus x_0^{(5)}[8] \\
&\oplus x_8^{(5)}[8] \oplus x_4^{(5)}[15] \oplus x_8^{(5)}[7] \oplus x_4^{(5)}[14]
\end{aligned}$$

Remark 2. With $\mathcal{ID}$ at $x_{13}^{(0)}[13]$, $\mathcal{OD}$ at $x_{11}^{(3)}[0]$, we obtain $\varepsilon_d = -0.0272 \approx -\frac{1}{2^{5.2}}$. As we described in Section 3.1.2, this extends to 4-round differential-linear bias with $\varepsilon_L = 1$, when the $\mathcal{OD}$ is $x_1^{(4)}[0] \oplus x_{11}^{(4)}[0] \oplus x_{12}^{(4)}[8] \oplus x_{12}^{(4)}[0]$. Further, in this section it is shown that $x_{11}^{(3)}[0] = x_{12}^{(5)}[24] \oplus x_0^{(5)}[16] \oplus x_0^{(5)}[0] \oplus x_8^{(5)}[0] \oplus x_4^{(5)}[7] \oplus x_{15}^{(5)}[0] \oplus x_{11}^{(5)}[0] \oplus x_{15}^{(5)}[8] \oplus x_3^{(5)}[0] \oplus x_1^{(5)}[0] \oplus x_5^{(5)}[7] \oplus x_5^{(5)}[19] \oplus x_9^{(5)}[12] \oplus x_{13}^{(5)}[0] \oplus x_{12}^{(5)}[0] \oplus x_0^{(5)}[24] \oplus x_0^{(5)}[8] \oplus x_8^{(5)}[8] \oplus x_4^{(5)}[15] \oplus x_8^{(5)}[7] \oplus x_4^{(5)}[14]$ with probability $\frac{1}{2}\left(1+\frac{1}{2}\right)$. The gives a total differential-linear 5^{th} 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 $\approx 2^{16}$.

Remark 3. As with Salsa, extending 3 rounds come at a cost. As discussed prior to the above lemma, for ChaCha, setting $i = 0$ in Lemma 3 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 9 and Lemma 10 (to attach the corresponding weight to each variable), the linear bias is $\varepsilon_L = \frac{1}{2^{1+3\cdot4+5\cdot1+3\cdot2+2\cdot1}} = \frac{1}{2^{26}}$. 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 is the currently best known 6 round attack on ChaCha.

To obtain the exact active output bits, we need to start expanding each term on the right side of the equations in Lemma 10. Depending on whether the term is LSB or not we would apply either Lemma 3 or Lemma 9 to do so. The exact number of bits is not easy to estimate without explicitly writing down the terms as cancellations may occur. Due to the high complexity of the the distinguisher, it is impossible to verify this bias experimentally. This method is unlikely to be useful for 7 or more rounds of ChaCha.

4 Implication of the new biases towards the cryptanalysis using Probabilistic Neutral Bits (PNBs)

The only cryptanalytic attack known for reduced round Salsa and ChaCha are using the decade old proposal of [1, 2]. The basic idea was to move forward a few rounds identifying a bias corresponding to one or more input bit differences and at the same time coming back through a few rounds considering that a few key bits (PNBs) will not affect the reversal much and thus one does not require to search those bits initially. Thus, only searching a subset of all the key bits, one may deduce whether the could identify those bits successfully by the help of a distinguisher. In case that is possible, we have the complexity of exhaustive search in that subset multiplied by the complexity of checking the distinguisher plus the complexity of the exhaustive search for the PNBs. Since this idea has been discussed in quite a few papers in great details [1, 2, 18, 17], we refer to these papers and skip the technical details. The main terms involved in assessing the complexity of this attack, as referred in the above-mentioned papers, are as follows:

- Bias in the forward direction after r rounds: ϵ_d.

- The n number of PNBs given a bias γ that relates to the threshold probability $\frac{1}{2}(1+\gamma)$ to choose the PNBs. The number of non-PNB key bits are thus $m = 256 - n$.

- The bias ϵ_a in the reverse direction considering all the PNBs for $R - r$ rounds.

- The bias ϵ which is generally approximated as $\epsilon_a \cdot \epsilon_d$ and considered for calculating the overall complexity of the attack on R rounds of the cipher.

- Following [1, 2], given the number of samples N and the probability of false alarm is $P_{fa} = 2^{-\alpha}$, the complexity of the attack is given by

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

for probability of non-detection $P_{nd} = 1.3 \times 10^{-3}$.

The main advantage we obtain here is we have significant biases for 5 and 6 rounds of Salsa as well as 4, 4.5 and 5 rounds of ChaCha. Thus, while comparing with the existing attack complexities, we need to come back less number of rounds while considering the PNBs. As correctly envisaged in earlier works [1, 2], while we consider the number of PNBs for multi-bit output differences, it reduces drastically in comparison to single-bit. That is why we could not increase the attack for more rounds than the existing works. However, for smaller number of reverse rounds, the number of PNBs are quite significant and that helps us to improve the existing complexities by a huge margin in certain cases as explained in Table 1. This works significantly better than existing results for 7 rounds of Salsa and 6 rounds of ChaCha. However, for 8 rounds of Salsa and 7 rounds of ChaCha we could only obtain slight improvements.

4.1 Salsa

For Salsa we have already described that we get the best known biases for 4, 5 and 6 rounds and that provides real time cryptanalysis of Salsa till 6 rounds that could not be achieved earlier. Now we consider the cases for 7 and 8 rounds. Note that, the biases have been calculated so far with a set of random keys and IV's as described in the earlier section. In this section, we run the experiments afresh as we need to run the experiments with the same key and different IV's in each run. We go for a number of such runs and then take the median value as considered in [1, 2] so that it is expected that the results will work for at least half of the randomly chosen keys. Here we run each experiment for 2^{30} randomly chosen IV's to get the average and then go for 256 such runs to obtain the median values.

4.1.1 7-round

Consider the input difference $\Delta x_7^{(0)}[0]$, which we will use for all the PNB based analysis here. First we take the output difference $\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$ as in the first row of Table 3. Here we go forward 5 rounds and come back 2 rounds. We have 149 PNBs here when we consider $\gamma = 0.27$. We obtained $\epsilon_a = 0.002510, \epsilon_d = -0.116754, \epsilon = -0.000294$. This gives us the data complexity of $2^{31.5}$ and time complexity of $2^{138.5}$ for $\alpha = 124$.

We can still make it better by going forward 6 rounds and coming back 1 round only with the same input difference. However, for the output difference will now consider 19 bits, i.e., we will take $\Delta x_0^{(6)}[6] \oplus \Delta x_0^{(6)}[5] \oplus \Delta x_1^{(6)}[13] \oplus \Delta x_1^{(6)}[25] \oplus \Delta x_1^{(6)}[24] \oplus \Delta x_2^{(6)}[25] \oplus \Delta x_2^{(6)}[20] \oplus \Delta x_3^{(6)}[6] \oplus \Delta x_3^{(6)}[20] \oplus \Delta x_{11}^{(6)}[19] \oplus \Delta x_{11}^{(6)}[18] \oplus \Delta x_8^{(6)}[19] \oplus \Delta x_9^{(6)}[0] \oplus \Delta x_{15}^{(6)}[23] \oplus \Delta x_{15}^{(6)}[22] \oplus \Delta x_{12}^{(6)}[23] \oplus \Delta x_{13}^{(6)}[0] \oplus \Delta x_{13}^{(6)}[5] \oplus \Delta x_{14}^{(6)}[5]$. We have 180 PNBs here when we consider $\gamma = 0.3$. The median value of ϵ_a is checked here through experiments which is 0.000386. We consider the estimated value of $\epsilon_d = -0.000028$ as in Section 3.2.1. Then we estimate $\epsilon = \epsilon_d \cdot \epsilon_a$. This gives us the data complexity of $2^{60.95}$ and time complexity of $2^{136.98}$ for $\alpha = 125$.

4.1.2 8-round

Here we need to follow the idea of chosen IV cryptanalysis as explained in [17]. Corresponding to the two key words in a column and given the constant word, we choose only those IV words such that the number of differences after the quarterround is the same as in the case when the modulo addition + (nonlinear) is replaced by the linear operation $\oplus$. The differential thus passes with probability one. This happens for proper choices of IV words given the key words. Naturally, one may require an additional data complexity of $2^{32 \cdot 3} = 2^{96}$ at maximum for maintaining the information regarding the proper choice of IV's.

In this case we go forward 5 rounds and come back 3 rounds (going forward 6 rounds and coming back 2 rounds, we do not get better result here). We consider the input and output difference as explained in the first row of Table 3. The input difference is $\Delta x_7^{(0)}[0]$, the output difference is $\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$. We have 40 PNBs here when we consider $\gamma = 0.1$ and further we add two more PNBs by trial and error. The PNBs can be represented as 0x7e000000, 0x00000000, 0x000001c0, 0x07800800 (0x03800800), 0x00080000, 0x0003fff0 (0x0001fff0), 0x001e0000, 0x807e0003. The 32-bit binary patterns correspond to the PNBs when we have a 1. For two words, we have a corresponding pair in parenthesis, showing the initial PNBs that we obtained for $\gamma = 0.1$ and then we have added one more key-bit in each case after several experiments. That is, we consider 42 PNBs in total. We run each experiment for 2^{34} randomly chosen IV's for each randomly selected key and go for 2048 such runs to obtain the median values. Note that out of these runs, we could obtain 1003 runs (around half) where we have proper choices of IV's in approximately 2^{32} cases. The rest of the runs are for such randomly chosen keys that there is no IV so that the difference(s) pass with probability one. Based on these 1003 data, we obtained $\epsilon_a = 0.000752, \epsilon_d = -0.233198, \epsilon = -0.000178$. Note that the ϵ_d bias gets doubled when we consider the chosen IVs and this helps in obtaining better complexity as discovered in [17]. This gives us $N = 2^{30.78}$ and time complexity of $2^{244.85}$ for $\alpha = 15.5$. The data complexity in this case is 2^{96} in worst case similar to that of [17] as discussed above. Thus, this provides comparable result as could be obtained in [17].

4.2 ChaCha

We have already described that it is possible to obtain distinguishers for 4, 4.5 and 5 rounds of ChaCha in real time. Now let us consider the PNB based attacks.

4.2.1 6-round

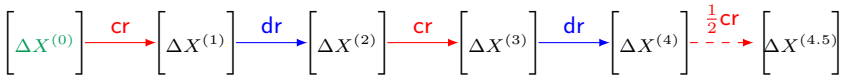
For the 6-round cryptanalysis, we run each experiment for 2^{34} randomly chosen IV's to get the average and then go for 256 such runs to obtain the median values. First consider the $\mathcal{ID}$ and $\mathcal{OD}$ as in the first row of Table 5. We go forward 4 rounds and come back 2 rounds. We have 159 PNBs here when we consider $\gamma = 0.4$. We obtained $\epsilon_a = 0.000534, \epsilon_d = 0.212786, \epsilon = 0.000110$. This gives us the data complexity of $2^{34.39}$ and time complexity of $2^{131.40}$ for $\alpha = 131$.

This may also be studied as 4.5 rounds forward and 1.5 rounds backward with the $\mathcal{ID}$ (same as above) and $\mathcal{OD}$ as in the first row of Table 10. With $\gamma = 0.5$, we have 161 PNBs in this case. With similar experiments, we obtain $\epsilon_a = 0.003958, \epsilon_d = 0.026652, \epsilon = 0.000106$. This provides the data complexity of $2^{34.51}$ and time complexity of $2^{129.53}$ for $\alpha = 133$.

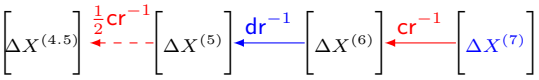
However, the best result can be obtained when we go forward 5 rounds and come back 1 round. We use the $\mathcal{ID}$, $\mathcal{OD}$ as given in Section 3.2.2 considering the linear combination of 21 bits at the output. With $\gamma = 0.5$, we have 166 PNBs in this case. Then experimentally, we obtain $\epsilon_a = 0.0028, \epsilon_d = 0.0068$, and estimate $\epsilon = \epsilon_a \cdot \epsilon_d = 0.000019$. This provides the data complexity of $2^{37.5}$ and time complexity of $2^{127.5}$ for $\alpha = 135$.

4.2.2 7-round

Here we need to go for chosen-IV cryptanalysis as in [17]. With 4-round forward and 3-round backward or 5-round forward and 2 round backward, we could not achieve competitive complexity as attained in [17]. However, with 4.5-round forward and 2.5-round backward (see Figure 1), we could manage comparable complexity as in [17].



(a) Forward direction for 4.5 rounds



(b) Reverse direction for 2.5 rounds

Figure 1: The two steps procedure for attacking 7 rounds of ChaCha using a 4.5 round bias. The columnround (shortened to cr) and diagonalround (shortened to dr) are represented by red and blue arrows respectively.The half round of ChaCha discussed earlier is represented by a dashed arrow. We follow the same notations for the reverse rounds.

We have 50 PNBs here when we consider $\gamma = 0.27$. We run each experiment for 2^{34} randomly chosen IV's and go for 2048 such runs. Out of these runs, we get 1572 many cases where the differences pass with probability one in the first round. Based on these 1572 many data, we obtained $\epsilon_a = 0.001162, \epsilon_d = 0.136828, \epsilon = 0.000152$. This provides $N = 2^{31.6}$ and time complexity $2^{237.65}$ for $\alpha = 23$. The data complexity here is similar to [17], which is 2^{96} in worst case.

4.2.3 7-round: Non-randomness

In [18], it has been discussed that choosing one or more differences in proper locations one may reverse one round of Salsa such that no difference is introduced in the constant locations. Thus, we may have differences in key or IV bits and we may consider that as the starting point. In this manner we may have the biases visible with the same value for one more round. Similar technique works for ChaCha also, and we show that it is possible to

come back half of the initial column round. Let us describe this with an example. Consider that we have a difference $\Delta x_{13}^{(0.5)}[13]$. If one comes back half round then it is possible to obtain valid initial states with differences in the keyword k_5 (i.e., x_9) and IV v_0 (i.e., x_{13}). This provide the $\mathcal{ID}$s. With such input key and IV differences, we obtain output difference $\Delta x_3^{(4.5)}[0] \oplus \Delta x_3^{(4.5)}[8] \oplus \Delta x_0^{(4.5)}[0] \oplus \Delta x_4^{(4.5)}[12] \oplus \Delta x_{10}^{(4.5)}[0] \oplus \Delta x_8^{(4.5)}[0] \oplus \Delta x_{14}^{(4.5)}[0] \oplus \Delta x_{15}^{(4.5)}[16] \oplus \Delta x_{15}^{(4.5)}[24]$ with a bias $\epsilon_d = 0.774648$.

We have 51 PNBs here when we consider $\gamma = 0.27$. We run each experiment for 2^{30} randomly chosen IV's and go for 256 such runs. We obtained $\epsilon_a = 0.000788, \epsilon = 0.000614$. This provides the data complexity $2^{27.8}$ and time complexity $2^{232.8}$ for $\alpha = 28$. We reemphasize that this improved result is to demonstrate non-randomness only and not cryptanalysis of the cipher as we have to accept differences in the key-bits too in addition to IV bit differences.

5 Conclusion

In this paper, we develop the first known theoretical results with respect to choosing a combination of output bits to obtain significantly improved biases. Building on this theory, and a limited search, we obtain the best known biases for 4 rounds of Salsa and 3 rounds of ChaCha. Using our strategy, after almost a decade, we report first known biases for 5/6 rounds of Salsa and 4/4.5/5 rounds of ChaCha. We also demonstrate that our theory extends to higher order differentials. Such multi-bit biases cannot be identified by exhaustive search methods due to the huge complexity involved. That is, the main contribution of this work is to identify how to obtain biased multi-bit differentials of Salsa and ChaCha by theoretical analysis. Surprisingly, such efforts have never been reported in literature and the ciphers have been studied primarily as black boxes while exploring the differentials. Automated search techniques did not provide much significant results either. This is the first time we looked into the structure of the ciphers in great details to obtain such highly biased multi-bit differentials.

Despite prior claims that multi-bit differentials cannot be effectively used in an attack, we obtain best known results in both Salsa and ChaCha using multi-bit output differentials. Significantly, for the lower rounds of Salsa (5 and 6) and ChaCha (4, 4.5 and 5), we move the attacks into practical realms for the first time. One may have a look at Table 1 in the introduction to see how it took quite a few years to achieve such low complexities for practical attacks. Further we could significantly improve the time complexities for 7-round Salsa and 6-round ChaCha. Due to the significant reduction of PNBs for more reverse rounds, we could only manage to slightly improve the existing time complexities for 8-round Salsa and 7-round ChaCha. We believe that our techniques might be advanced further to obtain better cryptanalysis for these ciphers.

Acknowledgments: The authors are grateful to the anonymous reviewers for their detailed technical and editorial comments. Based on these comments, the presentation and the results improved significantly compared to the initial submission. In fact, the six round bias of Salsa, explained in Remark 1, was first pointed out by an anonymous reviewer during the first round of review. The second author likes to acknowledge the Centre of Excellence in Cryptology, Indian Statistical Institute for supporting this work.

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. *IACR Cryptology ePrint Archive*, 2007:472, 2007.

[2] Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New features of Latin dances: analysis of Salsa, ChaCha, and Rumba. In *Fast Software Encryption*, pages 470–488. Springer, 2008.

[3] Daniel J Bernstein. Salsa20 specification. *eSTREAM Project algorithm description, http://www.ecrypt.eu.org/stream/salsa20pf.html*, 2005.

[4] Daniel J Bernstein. ChaCha, a variant of Salsa20. In *Workshop Record of SASC*, volume 8, 2008.

[5] Eli Biham, Orr Dunkelman, and Nathan Keller. Enhancing differential-linear cryptanalysis. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, pages 254–266, 2002.

[6] Céline Blondeau and Kaisa Nyberg. Joint Data and Key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and Its Impact to Data Complexity. http://eprint.iacr.org/2015/935, 2015

[7] Andrey Bogdanov and Elmar Tischhauser. On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2. In *Fast Software Encryption*, pages 19–38. Springer, 2013.

[8] Julio César Hernández Castro, Juan M. Estévez-Tapiador, and Jean-Jacques Quisquater. On the Salsa20 core function. In *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, pages 462–469, 2008.

[9] Paul Crowley. Truncated differential cryptanalysis of five rounds of Salsa20. *IACR Cryptology ePrint Archive*, 2005:375, 2005.

[10] The ECRYPT stream cipher project. eSTREAM portfolio of stream ciphers. http://www.ecrypt.eu.org/stream/.

[11] Simon Fischer, Willi Meier, Côme Berbain, Jean-François Biasse, and Matthew J. B. Robshaw. Non-randomness in eSTREAM Candidates Salsa20 and TSC-4. In *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, pages 2–16, 2006.

[12] Tsukasa Ishiguro, Shinsaku Kiyomoto, and Yutaka Miyake. Latin Dances Revisited: New Analytic Results of Salsa20 and ChaCha. In *Information and Communications Security - 13th International Conference, ICICS 2011, Beijing, China, November 23-26, 2011. Proceedings*, pages 255–266, 2011.

[13] Susan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, pages 17–25, 1994.

[14] Gaëtan Leurent. Analysis of Differential Attacks in ARX Constructions. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 226–243, 2012.

[15] Gaëtan Leurent. Construction of Differential Characteristics in ARX Designs Application to Skein. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 241–258, 2013.

[16] Gaëtan Leurent. Improved differential-linear cryptanalysis of 7-round chaskey with partitioning. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 344–371, 2016.

[17] Subhamoy Maitra. Chosen IV cryptanalysis on reduced round ChaCha and Salsa. *Discrete Applied Mathematics*, 208:88 – 97, 2016.

[18] Subhamoy Maitra, Goutam Paul, and Willi Meier. Salsa20 cryptanalysis: New moves and revisiting old styles. In *WCC 2015, the Ninth International Workshop on Coding and Cryptography, April 13-17, 2015, Paris, France.*, 2015. See also http://eprint.iacr.org/2015/217.

[19] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 386–397, 1993.

[20] Nicky Mouha and Bart Preneel. A Proof that the ARX Cipher Salsa20 is Secure against Differential Cryptanalysis. *IACR Cryptology ePrint Archive*, 2013:328, 2013.

[21] Ali Aydin Selcuk. On Probability of Success in Linear and Differential Cryptanalysis. Journal of Cryptology 21(1): 131-147 (2008)

[22] Zhenqing Shi, Bin Zhang, Dengguo Feng, and Wenling Wu. Improved Key Recovery Attacks on Reduced-Round Salsa20 and ChaCha. In *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*, pages 337–351, 2012.

[23] The Transport Layer Security (TLS) Protocol Version 1.3 draft-ietf-tls-tls13-12. https://tools.ietf.org/html/draft-ietf-tls-tls13-12.

[24] Yukiyasu Tsunoo, Teruo Saito, Hiroyasu Kubo, Tomoyasu Suzaki, and Hiroki Nakashima. Differential Cryptanalysis of Salsa20/8, 2007.

[25] Vesselin Velichkov, Nicky Mouha, Christophe De Cannière, and Bart Preneel. UNAF: A special set of additive differences with application to the differential analysis of ARX. In *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, pages 287–305, 2012.

[26] Johan Wallén. Linear approximations of addition modulo 2^n. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, pages 261–273, 2003.

A Higher Order Differentials

In this section we refer to some $\mathcal{ID}/\mathcal{OD}$ combinations in Section 3.1 for studying the higher order differentials. The notation follows from the work in [22] and is a slight departure from the notations being used thus far. The reader should note that the notations used in this section are limited only to this section. For instance, we use matrices of the form X_i for $i > 0$ where we define $\Delta X_i = X_0 \oplus X_i$. We refer to a word u of ΔX_i by $\Delta x_{i,u}$.

A.1 Second-Order Differential

Let $X_0^{(0)}$ be the initial state matrix, $X_1^{(0)}$, $X_2^{(0)}$ and $X_3^{(0)}$ be the associated initial matrices with a single-bit input difference $\Delta x_{1,i_1}^{(0)} = 2^{j_1}$, a single-bit input difference $\Delta x_{2,i_2}^{(0)} = 2^{j_2}$ and the double-bit differences $\Delta x_{3,i_1}^{(0)} = 2^{j_1}$ and $\Delta x_{3,i_2}^{(0)} = 2^{j_2}$ respectively. The second order single-bit output difference after r rounds is defined as

$$\Delta x_p^{(r)}[q] = \Delta x_{0,p}^{(r)}[q] \oplus x_{1,p}^{(r)}[q]\Delta x_{2,p}^{(r)}[q] \oplus x_{3,p}^{(r)}[q].$$

For input X, this is denoted by $(\Delta x_p^{(r)}[q] \mid \Delta x_{i_1}^{(0)}[j_1], \Delta x_{i_2}^{(0)}[j_2])$. Similar to the bias defined earlier, the bias of the second order output difference is denoted by

$$\mathsf{Pr}[\Delta x_p^{(r)}[q] \mid \Delta x_{i_1}^{(0)}[j_1] = 1, \Delta x_{i_1}^{(0)}[j_1] = 1] = \frac{1}{2}(1 + \varepsilon_d),$$

where the probability holds over all keys, nonces and counters.

A.2 Second Order Multi-bit Differential

Shi et al. [22] reported high single bit second order differentials for 4 rounds of Salsa and 3 rounds of ChaCha. Our theoretical results for multi-bits discussed earlier extend directly to the second order differentials due to the linearity of operations. This is illustrated for Salsa in Table 6 (better biases compared to first order differential as in Table 3) and for ChaCha in Table 7 (better biases compared to first order differential as in Table 5).

Table 6: Best second order triple bit differentials for 5 rounds of Salsa

$\mathcal{ID} : \Delta x_{i_1}^{(0)}[j_1], \Delta x_{i_2}^{(0)}[j_2]$	$\mathcal{OD}$	Bias
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{23}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.3310
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{24}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.3232
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{25}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.3212
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{26}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.3096
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{27}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.3010
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{28}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.2914
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{10}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.2808
$\Delta x_7^{(0)} = 2^{17}, \Delta x_8^{(0)} = 2^{29}$	$\Delta x_9^{(5)}[0] \oplus \Delta x_{13}^{(5)}[0] \oplus \Delta x_1^{(5)}[13]$	0.2800

But unfortunately, as remarked by Shi et al. in [22], we do not know how to use these high biases to good effect when considering the reverse rounds. We refer the reader to [22] for details regarding construction of Row Chaining Distinguishers (RCD) for second order differentials.

We ran extensive tests to observe third order differentials for Salsa and ChaCha, but were unable to obtain significantly better biases than those of second order.

The limited search differs slightly from the case of the first order differential. Here, we search over all possible pairs and triplets of input differences in the IV for second and third

Table 7: Best second order 5-bit differentials for 4 rounds of ChaCha

$\mathcal{ID} : \Delta x_{i_1}^{(0)}[j_1], \Delta x_{i_2}^{(0)}[j_2]$	$\mathcal{OD}$	Bias
$\Delta x_{14}^{(0)} = 2^{15}, \Delta x_{15}^{(0)} = 2^{15}$	$\Delta x_1^{(4)}[0] \oplus \Delta x_6^{(4)}[7] \oplus \Delta x_6^{(4)}[19] \oplus \Delta x_{11}^{(4)}[12] \oplus \Delta x_{12}^{(4)}[0]$	−0.4314
$\Delta x_{12}^{(0)} = 2^{15}, \Delta x_{13}^{(0)} = 2^{20}$	$\Delta x_3^{(4)}[0] \oplus \Delta x_4^{(4)}[7] \oplus \Delta x_4^{(4)}[19] \oplus \Delta x_9^{(4)}[12] \oplus \Delta x_{14}^{(4)}[0]$	−0.4313
$\Delta x_{13}^{(0)} = 2^{15}, \Delta x_{14}^{(0)} = 2^{20}$	$\Delta x_0^{(4)}[0] \oplus \Delta x_5^{(4)}[7] \oplus \Delta x_5^{(4)}[19] \oplus \Delta x_{10}^{(4)}[12] \oplus \Delta x_{15}^{(4)}[0]$	−0.4311
$\Delta x_{12}^{(0)} = 2^{20}, \Delta x_{15}^{(0)} = 2^{15}$	$\Delta x_2^{(4)}[0] \oplus \Delta x_7^{(4)}[7] \oplus \Delta x_7^{(4)}[19] \oplus \Delta x_8^{(4)}[12] \oplus \Delta x_{13}^{(4)}[0]$	−0.4308
$\Delta x_{14}^{(0)} = 2^{16}, \Delta x_{15}^{(0)} = 2^{20}$	$\Delta x_1^{(4)}[0] \oplus \Delta x_6^{(4)}[7] \oplus \Delta x_6^{(4)}[19] \oplus \Delta x_{11}^{(4)}[12] \oplus \Delta x_{12}^{(4)}[0]$	0.4011
$\Delta x_{13}^{(0)} = 2^{16}, \Delta x_{14}^{(0)} = 2^{20}$	$\Delta x_0^{(4)}[0] \oplus \Delta x_5^{(4)}[7] \oplus \Delta x_5^{(4)}[19] \oplus \Delta x_{10}^{(4)}[12] \oplus \Delta x_{15}^{(4)}[0]$	0.4004
$\Delta x_{13}^{(0)} = 2^{23}, \Delta x_{14}^{(0)} = 2^{20}$	$\Delta x_0^{(4)}[0] \oplus \Delta x_5^{(4)}[7] \oplus \Delta x_5^{(4)}[19] \oplus \Delta x_{10}^{(4)}[12] \oplus \Delta x_{15}^{(4)}[0]$	0.3988
$\Delta x_{14}^{(0)} = 2^{23}, \Delta x_{15}^{(0)} = 2^{20}$	$\Delta x_1^{(4)}[0] \oplus \Delta x_6^{(4)}[7] \oplus \Delta x_6^{(4)}[19] \oplus \Delta x_{11}^{(4)}[12] \oplus \Delta x_{12}^{(4)}[0]$	0.3988

order respectively. This gives $\binom{128}{2}$ and $\binom{128}{3}$ differences to search over, but we search over only 8 possible output bit combinations as in the case of the first order differential.

B A few tables

Table 8: Best triple bit differentials for 4 rounds of Salsa

$\mathcal{ID}$	$\mathcal{OD}$	Bias
$\Delta x_8^{(0)} = 2^{20}$	$\Delta x_{12}^{(4)}[0] \oplus \Delta x_{13}^{(4)}[0] \oplus \Delta x_{14}^{(4)}[13]$	−0.9999
$\Delta x_7^{(0)} = 2^{20}$	$\Delta x_{11}^{(4)}[0] \oplus \Delta x_8^{(4)}[0] \oplus \Delta x_9^{(4)}[13]$	−0.9999
$\Delta x_7^{(0)} = 2^{21}$	$\Delta x_{11}^{(4)}[0] \oplus \Delta x_8^{(4)}[0] \oplus \Delta x_9^{(4)}[13]$	0.9998
$\Delta x_8^{(0)} = 2^{21}$	$\Delta x_{12}^{(4)}[0] \oplus \Delta x_{13}^{(4)}[0] \oplus \Delta x_{14}^{(4)}[13]$	0.9998
$\Delta x_8^{(0)} = 2^{22}$	$\Delta x_{12}^{(4)}[0] \oplus \Delta x_{13}^{(4)}[0] \oplus \Delta x_{14}^{(4)}[13]$	0.9997
$\Delta x_7^{(0)} = 2^{22}$	$\Delta x_{11}^{(4)}[0] \oplus \Delta x_8^{(4)}[0] \oplus \Delta x_9^{(4)}[13]$	0.9997
$\Delta x_8^{(0)} = 2^{13}$	$\Delta x_{12}^{(4)}[0] \oplus \Delta x_{13}^{(4)}[0] \oplus \Delta x_{14}^{(4)}[13]$	−0.9996
$\Delta x_7^{(0)} = 2^{13}$	$\Delta x_{11}^{(4)}[0] \oplus \Delta x_8^{(4)}[0] \oplus \Delta x_9^{(4)}[13]$	−0.9996

Table 9: Best multi-bit differentials for 3 rounds of ChaCha

$\mathcal{ID}$	$\mathcal{OD}$	Bias
$\Delta x_{12}^{(0)} = 2^{10}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000
$\Delta x_{12}^{(0)} = 2^{11}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000
$\Delta x_{12}^{(0)} = 2^{12}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000
$\Delta x_{12}^{(0)} = 2^{13}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000
$\Delta x_{12}^{(0)} = 2^{14}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000
$\Delta x_{12}^{(0)} = 2^{15}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000
$\Delta x_{12}^{(0)} = 2^{8}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	−1.0000
$\Delta x_{12}^{(0)} = 2^{9}$	$\Delta x_0^{(3)}[0] \oplus \Delta x_4^{(3)}[7] \oplus \Delta x_4^{(3)}[19] \oplus \Delta x_8^{(3)}[12] \oplus \Delta x_{12}^{(4)}[0]$	1.0000

C Proof of some results

C.1 Proof of Lemma 5

We revisit the definition of the carry bit $C[i]$ of the modular addition $s = x + y$.

Table 10: Best multi-bit differentials for 4.5 rounds of ChaCha

$\mathcal{ID}$	$\mathcal{OD}$	Bias
$\Delta x_{13}^{(0)} = 2^{13}$	$\Delta x_0^{(4.5)}[0] \oplus \Delta x_0^{(4.5)}[8] \oplus \Delta x_1^{(4.5)}[0] \oplus \Delta x_5^{(4.5)}[12] \oplus \Delta x_{11}^{(4.5)}[0] \oplus \Delta x_9^{(4.5)}[0] \oplus \Delta x_{15}^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[16] \oplus \Delta x_{12}^{(4.5)}[24]$	0.0282
$\Delta x_{14}^{(0)} = 2^{13}$	$\Delta x_1^{(4.5)}[0] \oplus \Delta x_1^{(4.5)}[8] \oplus \Delta x_2^{(4.5)}[0] \oplus \Delta x_6^{(4.5)}[12] \oplus \Delta x_8^{(4.5)}[0] \oplus \Delta x_{10}^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[0] \oplus \Delta x_{13}^{(4.5)}[16] \oplus \Delta x_{13}^{(4.5)}[24]$	0.0278
$\Delta x_{15}^{(0)} = 2^{13}$	$\Delta x_2^{(4.5)}[0] \oplus \Delta x_2^{(4.5)}[8] \oplus \Delta x_3^{(4.5)}[0] \oplus \Delta x_7^{(4.5)}[12] \oplus \Delta x_9^{(4.5)}[0] \oplus \Delta x_{11}^{(4.5)}[0] \oplus \Delta x_{13}^{(4.5)}[0] \oplus \Delta x_{14}^{(4.5)}[16] \oplus \Delta x_{14}^{(4.5)}[24]$	0.0277
$\Delta x_{12}^{(0)} = 2^{13}$	$\Delta x_3^{(4.5)}[0] \oplus \Delta x_3^{(4.5)}[8] \oplus \Delta x_0^{(4.5)}[0] \oplus \Delta x_4^{(4.5)}[12] \oplus \Delta x_{10}^{(4.5)}[0] \oplus \Delta x_8^{(4.5)}[0] \oplus \Delta x_{14}^{(4.5)}[0] \oplus \Delta x_{15}^{(4.5)}[16] \oplus \Delta x_{15}^{(4.5)}[24]$	0.0266
$\Delta x_{12}^{(0)} = 2^{25}$	$\Delta x_3^{(4.5)}[0] \oplus \Delta x_3^{(4.5)}[8] \oplus \Delta x_0^{(4.5)}[0] \oplus \Delta x_4^{(4.5)}[12] \oplus \Delta x_{10}^{(4.5)}[0] \oplus \Delta x_8^{(4.5)}[0] \oplus \Delta x_{14}^{(4.5)}[0] \oplus \Delta x_{15}^{(4.5)}[16] \oplus \Delta x_{15}^{(4.5)}[24]$	−0.0116
$\Delta x_{13}^{(0)} = 2^{25}$	$\Delta x_0^{(4.5)}[0] \oplus \Delta x_0^{(4.5)}[8] \oplus \Delta x_1^{(4.5)}[0] \oplus \Delta x_5^{(4.5)}[12] \oplus \Delta x_{11}^{(4.5)}[0] \oplus \Delta x_9^{(4.5)}[0] \oplus \Delta x_{15}^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[16] \oplus \Delta x_{12}^{(4.5)}[24]$	−0.0106
$\Delta x_{15}^{(0)} = 2^{25}$	$\Delta x_2^{(4.5)}[0] \oplus \Delta x_2^{(4.5)}[8] \oplus \Delta x_3^{(4.5)}[0] \oplus \Delta x_7^{(4.5)}[12] \oplus \Delta x_9^{(4.5)}[0] \oplus \Delta x_{11}^{(4.5)}[0] \oplus \Delta x_{13}^{(4.5)}[0] \oplus \Delta x_{14}^{(4.5)}[16] \oplus \Delta x_{14}^{(4.5)}[24]$	−0.0105
$\Delta x_{15}^{(0)} = 2^{29}$	$\Delta x_1^{(4.5)}[0] \oplus \Delta x_1^{(4.5)}[8] \oplus \Delta x_2^{(4.5)}[0] \oplus \Delta x_6^{(4.5)}[12] \oplus \Delta x_8^{(4.5)}[0] \oplus \Delta x_{10}^{(4.5)}[0] \oplus \Delta x_{12}^{(4.5)}[0] \oplus \Delta x_{13}^{(4.5)}[16] \oplus \Delta x_{13}^{(4.5)}[24]$	−0.0104

$$C[i+1] = x[i] \cdot y[i] \oplus (x[i] \oplus y[i]) \cdot C[i],$$

where $C[0] = 0$. $s[i]$ is now defined as

$$s[i] = x[i] \oplus y[i] \oplus C[i].$$

Let $V = x[i] \cdot y[i] \oplus x[i] \oplus y[i] \oplus (x[i] \oplus y[i] \oplus 1) \cdot C[i]$. Hence, from the definition of the linear masking vector,

$$\Gamma_i \cdot (x + y) = V \oplus x[i+1] \oplus y[i+1].$$

To prove the lemma, we need to determine the probability that V is 0.

$$\begin{aligned}
\mathsf{Pr}[V = 0] &= \mathsf{Pr}[V = 0 \mid C[i] = 0] \cdot \mathsf{Pr}[C[i] = 0] + \mathsf{Pr}[V = 0 \mid C[i] = 1] \cdot \mathsf{Pr}[C[i] = 1] \\
&= \mathsf{Pr}[x[i] \cdot y[i] \oplus x[i] \oplus y[i] = 0] \cdot \mathsf{Pr}[C[i] = 0] + \mathsf{Pr}[x[i] \cdot y[i] \oplus 1 = 0] \cdot \mathsf{Pr}[C[i] = 1] \\
&= \frac{1}{4} \cdot \mathsf{Pr}[C[i] = 0] + \frac{1}{4} \cdot \mathsf{Pr}[C[i] = 1] \\
&= \frac{1}{4} = \frac{1}{2}\left(1 - \frac{1}{2}\right)
\end{aligned}$$

Hence,

$$\Gamma_i \cdot (x + y) = x[i+1] \oplus y[i+1]$$

with probability $\frac{1}{2}\left(1 - \frac{1}{2}\right)$.

C.2 Proof of Lemma 7

We give a brief proof of one of the cases in Lemma 7. The goal is to represent a single input active bit from round $m-1$ using bits of round m where $\varepsilon_L < 1$. We start again with the quarterround updates for Salsa.

$$
\begin{aligned}
x_b^{(m)}[i+7] &= x_b^{(m-1)}[i+7] \oplus (x_a^{(m-1)}[i] \oplus x_d^{(m-1)}[i] \oplus C_1[i]) \\
x_c^{(m)}[i+9] &= x_c^{(m-1)}[i+9] \oplus (x_b^{(m-1)}[i] \oplus x_a^{(m-1)}[i] \oplus C_2[i]) \\
x_d^{(m)}[i+13] &= x_d^{(m-1)}[i+13] \oplus (x_c^{(m)}[i] \oplus x_b^{(m)}[i] \oplus C_3[i]) \\
x_a^{(m)}[i+18] &= x_a^{(m-1)}[i+18] \oplus (x_d^{(m)}[i] \oplus x_c^{(m)}[i] \oplus C_4[i])
\end{aligned}
$$

Here we demonstrate by the linear approximation of x_b which is the most complicated of the four cases. The other cases follow similarly. At each step we use one of the linear approximations mentioned in Section 3.2.

$$
\begin{aligned}
x_b^{(m-1)}[i+7] &= x_b^{(m)}[i+7] \oplus (x_a^{(m-1)}[i] \oplus x_d^{(m-1)}[i] \oplus C_1[i]) \\
&= x_b^{(m)}[i+7] \oplus (x_a^{(m-1)}[i] \oplus x_d^{(m-1)}[i] \oplus x_a^{(m-1)}[i-1]) \text{ w.p. } \frac{1}{2}\left(1+\frac{1}{2}\right) \\
&= x_b^{(m)}[i+7] \oplus (x_a^{(m-1)}[i] \oplus x_a^{(m-1)}[i-1] \oplus (x_d^{(m)}[i] \\
&\quad \oplus (x_c^{(m)}[i-13] \oplus x_b^{(m)}[i-13] \oplus C_3[i-13]))) \text{ w.p. } \frac{1}{2}\left(1+\frac{1}{2}\right) \\
&= x_b^{(m)}[i+7] \oplus (x_a^{(m-1)}[i] \oplus x_a^{(m-1)}[i-1] \oplus (x_d^{(m)}[i] \\
&\quad \oplus x_c^{(m)}[i-13] \oplus x_b^{(m)}[i-13] \oplus x_b^{(m)}[i-13])) \text{ w.p. } \frac{1}{2}\left(1+\frac{1}{2^2}\right) \\
&= x_b^{(m)}[i+7] \oplus (x_a^{(m)}[i] \oplus x_a^{(m)}[i-1] \oplus \Gamma_{i-19} \cdot (x_d^m + x_c^m) \\
&\quad \oplus (x_d^{(m)}[i] \oplus x_c^{(m)}[i-13] \oplus x_b^{(m)}[i-13] \oplus x_b^{(m)}[i-13])) \text{ w.p. } \frac{1}{2}\left(1+\frac{1}{2^2}\right) \\
&= x_b^{(m)}[i+7] \oplus (x_a^{(m)}[i] \oplus x_a^{(m)}[i-1] \oplus x_d^m[i-18] \oplus x_c^m[i-18] \\
&\quad \oplus (x_d^{(m)}[i] \oplus x_c^{(m)}[i-13] \oplus x_b^{(m)}[i-13] \oplus x_b^{(m)}[i-13])) \text{ w.p. } \frac{1}{2}\left(1-\frac{1}{2^3}\right)
\end{aligned}
$$

C.3 Proof of Lemma 8

Given Lemma 7 and Lemma 1, we sketch the proof for a case in Lemma 8. In the first step we use the result of Lemma 1 to express $x_1^{(4)}[13]$ in terms of bits from round 5. Then using the results in Lemma 7, we expand each term in round 5 to get an expression in round 6 with the corresponding probability.

$$
\begin{aligned}
x_1^{(4)}[13] &= x_1^{(5)}[13] \oplus x_9^{(5)}[0] \oplus x_{13}^{(5)}[0] \\
&= x_1^{(6)}[13] \oplus x_0^{(6)}[6] \oplus x_0^{(6)}[5] \oplus x_3^{(6)}[20] \oplus x_2^{(6)}[20] \\
&\quad \oplus x_3^{(6)}[6] \oplus x_2^{(6)}[25] \oplus x_1^{(6)}[25] \oplus x_1^{(6)}[24] \\
&\quad \oplus x_9^{(5)}[0] \oplus x_{13}^{(5)}[0] && \text{w.p. } \frac{1}{2}\left(1 - \frac{1}{2^3}\right) \\
&= x_1^{(6)}[13] \oplus x_0^{(6)}[6] \oplus x_0^{(6)}[5] \oplus x_3^{(6)}[20] \oplus x_2^{(6)}[20] \\
&\quad \oplus x_3^{(6)}[6] \oplus x_2^{(6)}[25] \oplus x_1^{(6)}[25] \oplus x_1^{(6)}[24] \\
&\quad \oplus x_9^{(6)}[0] \oplus x_8^{(6)}[19] \oplus x_{11}^{(6)}[19] \oplus \oplus x_{11}^{(6)}[18] \oplus x_{13}^{(5)}[0] && \text{w.p. } \frac{1}{2}\left(1 - \frac{1}{2^4}\right) \\
&= x_1^{(6)}[13] \oplus x_0^{(6)}[6] \oplus x_0^{(6)}[5] \oplus x_3^{(6)}[20] \oplus x_2^{(6)}[20] \\
&\quad \oplus x_3^{(6)}[6] \oplus x_2^{(6)}[25] \oplus x_1^{(6)}[25] \oplus x_1^{(6)}[24] \\
&\quad \oplus x_9^{(6)}[0] \oplus x_8^{(6)}[19] \oplus x_{11}^{(6)}[19] \oplus \oplus x_{11}^{(6)}[18] \\
&\quad \oplus x_{13}^{(6)}[0] \oplus x_{12}^{(6)}[23] \oplus x_{15}^{(6)}[23] \oplus x_{15}^{(6)}[22] \oplus x_{14}^{(6)}[5] \oplus x_{13}^{(6)}[5] && \text{w.p. } \frac{1}{2}\left(1 + \frac{1}{2^6}\right)
\end{aligned}
$$