Designs, Codes and Cryptography
https://doi.org/10.1007/s10623-020-00736-9

Proving the biases of Salsa and ChaCha in differential attack

Sabyasachi Dey[1] · **Santanu Sarkar**[2]

Received: 19 June 2019 / Revised: 21 November 2019 / Accepted: 5 February 2020

Abstract
Salsa and ChaCha are two of the most famous stream ciphers in recent times. Most of the attacks available so far against these two ciphers are differential attacks, where a difference is given as an input in the initial state of the cipher and in the output some correlation is investigated. This correlation works as a distinguisher. All the key recovery attacks against these ciphers are based on these observed distinguishers. However, the distinguisher in the differential attack was purely an experimental observation, and the reason for this bias was unknown so far. In this paper, we provide a full theoretical proof of both the observed distinguishers for Salsa and ChaCha. In the key recovery attack, the idea of probabilistically neutral bit also plays a vital role. Here, we also theoretically explain the reason of a particular key bit of Salsa to be probabilistically neutral. This is the first attempt to provide a theoretical justification of the idea of differential key recovery attack against these two ciphers.

Keywords Salsa · ChaCha · Probabilistic neutral bits · Bias · Theoretical justification

Mathematics Subject Classification 94A60

1 Introduction

Salsa20 was designed by D. Bernstein in the year 2005 as a candidate for the eStream [2] project organised in Europe by EU ECRYPT. It was one of the finalists in this competition.

The first version of this work [30] was presented in the "Eleventh International Workshop on Coding and Cryptography (WCC 2019)".

This is one of several papers published in *Designs, Codes and Cryptography* comprising the "Special Issue on Coding and Cryptography 2019".

✉ Santanu Sarkar
sarkar.santanu.bir@gmail.com

Sabyasachi Dey
sabya@hyderabad.bits-pilani.ac.in

[1] Department of Mathematics, Birla Institute of Technology and Science Pilani, Hyderabad, Jawahar Nagar, Hyderabad 500078, India

[2] Department of Mathematics, Indian Institute of Technology Madras, Sardar Patel Road, Chennai 600036, India

The original version of Salsa has 20 rounds. However, the designer submitted the 12 rounds version in eStream. The first cryptanalysis against Salsa was presented by Crowley [5], who attacked it up to five rounds. Later, six rounds and seven rounds Salsa were attacked respectively by Fischer et al. [12] and Tsunoo et al. [20].

ChaCha is a variant of Salsa, designed in 2008 by Bernstein. ChaCha has a similar structure as Salsa, but uses a more complicated round function. ChaCha was begun to be adopted in Chrome in 2013 [3]. It officially became an IETF RFC for use in TLS, adopted by Google and many others, in 2016.

So far, Salsa and ChaCha have been attacked only up to eight and seven rounds, respectively. The central ideas of most of the attacks are based on differential distinguisher. In this attack procedure, a difference is given as an input in the initial state. After running the cipher by a few rounds, if some bias is observed at the output, the cipher is said to be distinguishable up to that round. In both these ciphers, it has been experimentally observed that on applying a difference at a single bit, some bias can be observed at a bit of the output up to fourth round. This bias is called forward bias. This bias has been exploited to produce key recovery attacks against the ciphers. Fischer et al. [12] used it in a key recovery attack up to sixth round in Salsa. In 2008, Aumasson et al. [1] produced a significant improvement in this attack by introducing the idea of probabilistically neutral bits. By using the differential in the 4th round, they attacked Salsa up to 8th round. Similarly, using a differential in the 3rd round, ChaCha has been attacked up to 7th round. The time complexities of these attacks were 2^{251} for Salsa and 2^{248} for ChaCha. Afterwards, there have been further improvements in the attack complexities in the works of [14,15,19]. Maitra [14] improved the observed bias for Salsa and ChaCha by properly choosing IVs in the key recovery attack. In 2012, Shi et al. [19] introduced a multi-step method of recovering the key. In FSE 2016, Choudhuri and Maitra [4] found a distinguisher in the 5th round of these ciphers by a linear combination of multiple bits of the output. In 2017, Dey and Sarkar [7] provided an algorithm to get a better set of probabilistic neutral bits. Several recent works in this area can be found in [6,9–11,17]. But still, none of the works has been able to extend the key recovery attack up to the next round for any of the ciphers in the last decade.

For a major improvement in this attack method, we need a detailed analysis of the whole attack procedure. The whole attack idea is so far mostly based only on experimental observations. Starting from the distinguisher, the idea of probabilistic neutral bits, meet-in-the middle attack, etc. mostly rely on experimental observations. A detailed investigation on the internal mechanism of the attacks is required. This may help to find some logical way to achieve some new trick that can provide a significant improvement. In the past, there have been several works in the field of stream ciphers which focus on investigating some experimental results and provide explanation of those results theoretically. For example, if we look at the research works on the stream cipher RC4, we can see various works on the theoretical justifications of several experimental results. In FSE 2001, Mantin and Shamir [16] proved a significant bias in the second output byte. In 2013 FSE, Isobe et al. [13] observed another bias in RC4 and provided a theoretical justification for this bias. In Journal of Cryptology 2014, Sengupta et al. [18] gave the accurate theoretical proof of biases of output bytes in RC4 towards zero. In Cryptography and Communication 2019, Dey and Sarkar [8] provided theoretical justification of the rth round output Z_r of RC4 towards r.

However, in the differential attack approach on Salsa and ChaCha, there have not been many theoretical works which investigate the reason of the experimental results. In this work, we go through the internal mechanism of how the differential distinguisher is created. We

try to explain the forward bias theoretically for both Salsa and ChaCha. Then, we give a brief insight into the idea of a probabilistically neutral bit, which plays a vital role in the key recovery attack.

Paper organisation

- In Sect. 2, we provide the structures of the two ciphers and discuss the idea of distinguisher in the differential attack.
- Section 3 provides the proofs of some theoretical results on probability which we use later in proving the forward bias for the two ciphers.
- Section 4, we provide a few points about our process and terminologies of the proof.
- In Sect. 5 we prove the forward bias observed in Salsa.
- Section 6 proves the forward bias for ChaCha.
- In Sect. 7 we give a theoretical justification for a probabilistic neutral bit for Salsa.
- Section 8 concludes the paper.

2 Structure

Salsa Salsa uses a 256 bit key in its algorithm. Another 128 bit key version is also there, which replicates its 128 bit key to another copy and produces total 256 keybits. The internal state of Salsa is basically a matrix of size 4×4. Each of the 16 cells contains a 32-bit binary number, which is called a 'word'. These 16 words or cells can be divided into three categories:

(1) *Constant cells* these cells lie in the diagonal of the matrix. These contain some predefined constant 32-bit numbers and are denoted by c_i's. The values of these cells are given below in hexadecimal form:

$$c_0 = 0x61707865, \quad c_1 = 0x3320646e, \quad c_2 = 0x79622d32, \quad c_3 = 0x6b206574.$$

(2) *Key cells* there are eight cells in the matrix which contain the keybits. These are denoted by $k_0, k_1, \ldots, k_7$.

(3) *Counters and nonces* there are four cells t_0, t_1, v_0 and v_1 which are counters and nonces. These are also called IVs.

$$X = \begin{pmatrix} X_0 & X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} \\ X_{12} & X_{13} & X_{14} & X_{15} \end{pmatrix} = \begin{pmatrix} c_0 & k_0 & k_1 & k_2 \\ k_3 & c_1 & v_0 & v_1 \\ t_0 & t_1 & c_2 & k_4 \\ k_5 & k_6 & k_7 & c_3 \end{pmatrix}.$$

The initial matrix is denoted by X or $X^{(0)}$. This matrix is updated by a function called quarterround. Each of the updates is called a round. After completion of r rounds, we denote the updated matrix as $X^{(r)}$. The original version of Salsa has 20 rounds. The quarterround function works on a 4-tuple (a, b, c, d). It uses three operations: addition modulo 2^{32} $(+)$, left rotation $(\lll)$ and XOR $(\oplus)$. The function is as follows:

$$\begin{aligned} b &= b \oplus ((a + d) \lll 7), \\ c &= c \oplus ((b + a) \lll 9), \\ d &= d \oplus ((c + b) \lll 13), \\ a &= a \oplus ((d + c) \lll 18). \end{aligned}$$

This function works on the columns and rows of the matrix in alternative round. The four entries of the columns (similarly rows) are taken to be the inputs (a, b, c, d) of the quarterround function. After application of the function, the updated (a, b, c, d) replaces the previous four entries of the columns or rows. However, the order of choosing the (a, b, c, d) for the four columns are (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 $(X_{15}, X_3, X_7, X_{11})$. For the rows the order is: (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 $(X_{15}, X_{12}, X_{13}, X_{14})$. In the odd rounds it works on the columns and this is called columnround. Application on rows are performed in the even rounds and called rowround.

The final round is denoted by R and the corresponding matrix by $X^{(R)}$. This $X^{(R)}$ is added with the initial matrix $X^{(0)}$ by usual matrix addition modulo 2^{32}. The sum is given as the output:

$$Z = X^{(0)} + X^{(R)}.$$

ChaCha ChaCha has the same key size and basic design as Salsa. Similar to Salsa, it also uses a 128 bit IV. However, the positions of the constant cells, IV cells and key cells are different in ChaCha from Salsa. The first row contains the constant cells, second and third row contain the key cells and fourth row contains the IV cells. The values of the constant cells are also the same as that of Salsa.

$$X = \begin{pmatrix} X_0 & X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} \\ X_{12} & X_{13} & X_{14} & X_{15} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & t_1 & v_0 & v_1 \end{pmatrix}.$$

The quarterround function used in ChaCha is different from Salsa. It is given as follows:

$$\begin{aligned} a &= a + b, \quad d = ((d \oplus a) \lll 16), \\ c &= c + d, \quad b = ((b \oplus c) \lll 12), \\ a &= a + b, \quad d = ((d \oplus a) \lll 8), \\ c &= c + d, \quad b = ((b \oplus c) \lll 7). \end{aligned}$$

In ChaCha, unlike Salsa, the quarterround is applied on the columns and diagonals. The odd rounds are columnrounds and the even rounds are diagonalrounds. The order of choosing (a, b, c, d) for columns is (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 $(X_3, X_7, X_{11}, X_{15})$. For diagonals it is as: $(X_0, X_5, X_{10}, X_{15})$, $(X_1, X_6, X_{11}, X_{12})$, (X_2, X_7, X_8, X_{13}) and (X_3, X_4, X_9, X_{14}).

The output generation function is also same as Salsa:

$$Z = X^{(0)} + X^{(R)}.$$

Notations we give a list of the notations used in this paper.

- X_i denotes the ith cell of the matrix X.
- $X_i[j]$ denotes the jth bit of the cell X_i.
- $X^{(r)}$ denotes the matrix after r rounds.
- $X_i^{(r)}[j]$ denotes the jth bit of the ith cell of the matrix after r rounds.
- location (p, q) denotes the qth bit of the pth cell.
- If we add two numbers a and b, by $(a+b)[j]$ we denote the jth bit of the sum $a+b$. For example, the jth bit of the sum of two cells X_{i_1} and X_{i_2} is denoted by $(X_{i_1} + X_{i_2})[j]$.
- The difference at the position (i, j) of X and X' is denoted by $\Delta_i[j]$, i.e., $\Delta_i[j] = X_i[j] \oplus X'_i[j]$.

- $\Pr(A)$ denotes the probability of the event A.
- For any event A, by A^c we denote the complement event of A, i.e., the event that A does not occur. Therefore, $\Pr(A^c) = 1 - \Pr(A)$.
- $\mathcal{P}_{(i,j)}$ denotes the probability of the event that the matrices X and X' have the same value at position (i, j), i.e., $\Delta_i[j] = 0$. So, $\mathcal{P}_{(i,j)} = \Pr(X_i[j] = X'_i[j]) = \Pr(\Delta_i[j] = 0)$. In general, when we compare two numbers a and a', $\mathcal{P}_{(a,j)}$ denotes the probability that the jth bit of a and a' are equal. In the figures, we have denoted the probability by $\mathcal{P}$.
- PNB denotes probabilistic neutral bits.

Idea of distinguisher in the differential attack here we discuss in brief the idea of a distinguisher in the differential attack on Salsa and ChaCha. The initial matrix X is constructed by choosing the IV randomly. A single bit (i, j) of a IV is changed in the initial matrix X producing a new matrix X'. As in the work of Aumasson et al. [1], the difference is given at position $(7, 31)$ in Salsa, and at position $(13, 13)$ in ChaCha. Then, the algorithm is applied for r rounds on both X and X'. After r rounds, a correlation is observed at some bit position (p, q) of X and X', i.e., the bit (p, q) of X and X' (after r rounds) are equal with a probability other than $\frac{1}{2}$. So, according to the notation mentioned before, the probability $\mathcal{P}_{(p,q)} = \frac{1}{2}(1 + \epsilon_d)$, where $\epsilon_d \neq 0$ is the bias. This bias is used to distinguish the cipher output from a random sequence. This is called the forward bias. It is an experimental observation and no theoretical approach has been made to justify the proper reason for this bias. In this paper, we attempt to prove this bias. For Salsa, the position (p, q) is $(1, 14)$ after 4 rounds and for ChaCha it is $(11, 0)$ after 3 rounds. We track the propagation of the input difference after each round and reach the rth round to find the bias in the position (p, q). In this context, we clearly mention here that throughout the paper whenever we say that a cell or bit is "not affected" or "not influenced" by the difference, we mean that up to that round, the same cell or bit position has the same value for X and X'.

In the next section, we provide some theoretical results based on probability. After that, in Sects. 5 and 6 we use these results to prove the biases observed in the fourth round of Salsa and third round of ChaCha.

3 Some mathematical results

Here, at first we state without proof a lemma from [7]. For the proof, one can check [7].

Lemma 1 *$a = a_{31}a_{30}a_{29}\cdots a_0$ and $b = b_{31}b_{30}b_{29}\cdots b_0$ be two independent uniformly randomly chosen numbers of 32 bits. Let $b' = b'_{31}b'_{30}b'_{29}\cdots b'_0$ be a number which differs at exactly one bit (say nth, $n \leq 31$) from b. Consider $S = a + b \bmod 2^{32}$ and $S' = a + b' \bmod 2^{32}$. Then for any $k \geq 0$ such that $n + k \leq 31$, the probability that S and S' will differ at $(n + k)$-th bit is $\frac{1}{2^k}$.*

One important fact that we need to clarify here is that, when the $(n + k)$th bit of S and S' differs, it implies that all the bits from nth to $(n + k - 1)$th differs. it is not possible that for some $i < k$, the $(n + i)$th bit would be equal even if the $(n + k)$th bit has difference. So, in other words, we can state the lemma in a better way that that: The difference would propagate to $(n + k)$th bit with probability $\frac{1}{2^k}$.

We will use this result repeatedly in our proofs in this paper.

Next, we deal with a case which is slightly more complicated and generalised. Suppose we add two different pairs of numbers a with b and a' with b', where the bits a_i's of a and a'_i's of a' match with some given probabilities p_i's. Similarly, bits of b and b' match with

probabilities q_i's. In such case, can we find the probabilities for the bits of $a + b$ to be equal to the corresponding bits of $a' + b'$? Let us start with the simple case when both a and b are single bit numbers. For any number X, by X_i we denote the ith bit of X, from the right side (the least significant bit is the 0th bit).

Theorem 1 *Let a, b be two independent uniformly randomly chosen single bit numbers. Let a', b' be single bit numbers chosen from a joint distribution such that $\Pr(a = a') = p$, $\Pr(b = b') = q$ and $\Pr(b = 0) = \Pr(a = 0) = \frac{1}{2}$. Let $s = a + b$ and $s' = a' + b'$. Then the probability $\Pr(s_i = s'_i)$ can be given by*:

(1) $pq + (1 - p)(1 - q)$ *for* $i = 0$.
(2) $\frac{1+pq}{2}$ *for* $i = 1$.

Proof (1) *For* $i = 0$ in this case, $s_0 = a \oplus b$ and $s'_0 = a' \oplus b'$. So, $s_0 = s'_0$ implies $s_0 \oplus s'_0 = 0$. Thus $(a \oplus a') \oplus (b \oplus b') = 0$. So, either $(a \oplus a') = (b \oplus b') = 0$ or $(a \oplus a') = (b \oplus b') = 1$. In the first case, the probability is:

$$\begin{aligned}\Pr((a = a') \cap (b = b')) &= \Pr(a = a') \cdot \Pr(b = b') \text{ (independence)} \\ &= pq.\end{aligned}$$

In the second case, the probability is

$$\Pr((a \neq a') \cap (b \neq b')) = \Pr(a \neq a') \cdot \Pr(b \neq b') = (1 - p)(1 - q).$$

(2) *For* $i = 1$ since a and b are single bits, so s_1 and s'_1 are basically the carries generated in the previous sums. So, we find out the possible ways of generation of a different carry in $a + b$ and $a' + b'$.

(a) *Event 1* one of (a, b) and (a', b') is $(0, 0)$ and the other one is $(1, 1)$. In this case, the tuple $(1, 1)$ generates a carry 1, but $(0, 0)$ does not. So, $s_i \neq s'_i$. Now, the probability of this event we calculate in the following way.
We first focus on the tuple (a, b). The probability that it takes the value either $(0, 0)$ or $(1, 1)$ is $\frac{1}{2}$, since there are total four possible options. Once this value is assigned, (a', b') must choose the other possible value. This means, a' must not be equal to a and b' must not be equal to b. This probability should be $(1 - p)(1 - q)$. Therefore, the probability of this whole event is $\frac{1}{2}(1 - p)(1 - q)$.

(b) *Event 2* $a = a' = 1$ and $b \neq b'$. In this case, one of the tuples (a, b) and (a', b') is $(1, 1)$, which generates carry 1, and the other one is $(1, 0)$, which generates carry 0. Now, $\Pr(a = a' = 1) = \Pr(a = a') \cdot \Pr(a = 1) = p \cdot \frac{1}{2} = \frac{p}{2}$. Then, $\Pr(b \neq b') = (1 - q)$. Therefore, the probability of this event is $\frac{p}{2}(1 - q)$.

(c) *Event 3* $b = b' = 1$ and $a \neq a'$. This event is similar to the previous one. By similar arguments it can be shown that the probability is $\frac{q}{2}(1 - p)$.
These three events are mutually disjoint and other than these three events there is no way of producing $s_i \neq s'_i$. Therefore the probability of $s_i \neq s'_i$ is the sum of these three probabilities, which is

$$\frac{1}{2}(1 - p)(1 - q) + \frac{p}{2}(1 - q) + \frac{q}{2}(1 - p) = \frac{1 - pq}{2}.$$

Therefore, the probability of equality can be given by $1 - \left(\frac{1-pq}{2}\right) = \frac{1+pq}{2}$.

□

Now, let us generalise this result little bit further. Suppose a, b, a', b' are not single bit numbers. We focus on some ith bit of them. Suppose we know the correlation between the corresponding bits of a and a' (and similarly for b and b'). The question is whether we can find a probabilistic relation between the ith bit of s and s'. One important point that we have to keep in mind is that the ith bits of s depends not only on a_i and b_i, but also on the carry produced from the previous bits. Suppose, by c_i and c'_i we denote the carries generated at $(i-1)$th bits of s and s' respectively, which are to be added in the ith bit. So, $s_i = a_i \oplus b_i \oplus c_i$ (same for s'). So, the probability of the equality of the carries c_i and c'_i plays a vital role in this case. In this context, we find the relation between $\Pr(c_{i+1} \neq c'_{i+1})$ and $\Pr(c_i \neq c'_i)$.

Theorem 2 *Suppose a, b be two independent and randomly chosen n-bit numbers. Let a', b' be n-bit numbers such that for all $i = \{0, 1, \ldots, (n-1)\}$, $\Pr(a_i = a'_i)$ is given by p_i and $\Pr(b_i = b'_i)$ is given by q_i. Suppose $s = a + b$ and $s' = a' + b'$. Then,*

$$\Pr(c_{i+1} \neq c'_{i+1}) = \Pr(c_i \neq c'_i) \cdot \left(1 - \frac{p_i + q_i - p_i q_i}{2}\right) + \Pr(c_i = c'_i)\frac{(1 - p_i q_i)}{2}.$$

Proof Suppose $c_i \neq c'_i$. Without loss of generality, let us assume that $c_i = 1$ and $c'_i = 0$. Let us find the possible cases where c_{i+1} and c'_{i+1} are different.
Event 1 one of (a_i, b_i) and (a'_i, b'_i) is $(0, 0)$ and the other one is $(1, 1)$. If $(a_i, b_i) = (1, 1)$ and $(a'_i, b'_i) = (0, 0)$, then $a_i + b_i + c_i = 3$, which generates a carry 1. On the other side, $a'_i + b'_i + c'_i = 0$, which generates carry 0. Therefore $c_{i+1} \neq c'_{i+1}$.
If $(a'_i, b'_i) = (1, 1)$ and $(a_i, b_i) = (0, 0)$, then $a_i + b_i + c_i = 1$, which generates carry 0 and $a'_i + b'_i + c'_i = 2$, which generates carry 1. Therefore, again $c_{i+1} \neq c'_{i+1}$.
The probability of this event is $\frac{(1-p_i)(1-q_i)}{2}$, as computed in the previous theorem.
Event 2 $(a_i, b_i), (a'_i, b'_i) \in \{(0, 1), (1, 0)\}$ and $(a_i, b_i) = (a'_i, b'_i)$. In this case, $a_i + b_i + c_i = 2$, which generates carry 1 and $a'_i + b'_i + c'_i = 1$, which generates carry 0. So, $c_{i+1} \neq c'_{i+1}$. The probability of this event can be calculated as $\Pr(a_i = a'_i) \cdot \Pr(b_i = b'_i) \cdot \Pr(a_i \neq b_i) = \frac{p_i q_i}{2}$.
Event 3 $(a_i, b_i), (a'_i, b'_i) \in \{(0, 1), (1, 0)\}$ and $(a_i, b_i) \neq (a'_i, b'_i)$. Similar to the previous event, one can easily verify that in this event $c_{i+1} \neq c'_{i+1}$. Also, by similar arguments as the previous event, the probability can be calculated as $\frac{(1-p_i)(1-q_i)}{2}$.
Event 4 $a_i = a'_i, b_i = 1, b'_i = 0$. In this case, $c_i + a_i + b_i \geq 2$, which gives a carry 1. On the other hand, $c'_i + a'_i + b'_i$ can be at most 1, which gives a carry 0. The probability of this event is $p_i \frac{1-q_i}{2}$.
Event 5 $b_i = b'_i, a_i = 1, a'_i = 0$. This can also be computed similarly as event 4. Probability of this event is $q_i \frac{1-p_i}{2}$.
These five events are mutually disjoint. So, given $c_i \neq c'_i$, the total probability of the event $(c_{i+1} = c'_{i+1})$ can be computed by adding them. This sum is $1 - \frac{p_i + q_i - p_i q_i}{2}$.
On the other hand, if $(c_i = c'_i)$, then both of them can be either 0 or 1. If 0, then we can treat a_i, b_i, a'_i, b'_i like the least significant bits as in Theorem 1. So, the probability is $\frac{1-p_i q_i}{2}$.
Similarly, if both the carries are 1, it can be shown similarly that the probability is $\frac{(1-p_i q_i)}{2}$.
So, the total probability of $\Pr(c_{i+1} \neq c'_{i+1})$ can be calculated as

$$\Pr(c_i \neq c'_i) \cdot \left(1 - \frac{p_i + q_i - p_i q_i}{2}\right) + \Pr(c_i = c'_i)\frac{(1 - p_i q_i)}{2}.$$

□

Theorem 3 *Consider a, b, a', b' as the n-bit numbers similar to Theorem 2. Also, s, s', c_j and c'_j are as mentioned. Now, suppose at the ith bit, $c_i \neq c'_i$. The probabilities $\Pr(a_i = a'_i)$*

and $\Pr(b_i = b'_i)$ *are given by* p *and* q, *respectively. Then, the probability that the* i*th bits of the sums* s *and* s' *match is* $p(1-q)+q(1-p)$. *On the other hand, if* $c_i = c'_i$, *this probability is* $pq + (1-p)(1-q)$.

Proof Without loss of generality, let us assume that $c_i = 1$ and $c'_i = 0$. Now, $s_i = a_i \oplus b_i \oplus c_i = a_i \oplus b_i \oplus 1$ and $s'_i = a'_i \oplus b'_i \oplus c'_i = a'_i \oplus b'_i$. Therefore $s_i = s'_i$ implies $a_i \oplus b_i \oplus a'_i \oplus b'_i = 1$. Therefore,

(1) either $a_i \oplus a'_i = 0$ and $b_i \oplus b'_i = 1$,
(2) or $a_i \oplus a'_i = 1$ and $b_i \oplus b'_i = 0$.

Probability of the first event is $\Pr(a_i = a'_i) \cdot \Pr(b_i \neq b'_i) = p(1-q)$. Similarly, probability of the second event is $q(1-p)$.
Adding these two probabilities, we get the desired result. If $c_i = c'_i$, the result can be proved in a similar manner. □

Lemma 2 $a = a_{31}a_{30}a_{29}\cdots a_0$ *and* $b = b_{31}b_{30}b_{29}\cdots b_0$ *be two independent arbitrarily chosen numbers of 32 bits. Let* $b' = b'_{31}b'_{30}b'_{29}\cdots b'_0$ *be a number which differs at exactly* m *consecutive bits (say at the bit position* $n, n+1, \ldots n+m-1$, $n+m-1 \leq 31$*) from* b. *Consider* $S = a + b \mod 2^{32}$ *and* $S' = a' + b' \mod 2^{32}$. c_i *and* c'_i *are the carries generated at the* $(i-1)$*th bit in* S *and* S', *respectively. Then for any* $k > 0$ *such that* $(n+m-1)+k \leq 31$, *the probability that* $c_{n+m-1+k}$ *and* $c'_{n+m-i+k}$ *will differ is* $\frac{1}{2^k}$.

Proof We prove it by induction on m. For $m = 1$ it is true from Lemma 1, because the $n+k$th bit of S and S' can differ only if c_{n+k} and c'_{n+k} differs, since $b_{n+k} = b'_{n+k}$.

Let us assume that it is true for some $m = r$. We prove it for $m = r+1$. So, $b_i \neq b'_i$ for $i \in n, n+1, \ldots, n+r$. Since the statement is true for $m = r$, for the block $b_{n+r-1}b_{n+r-2}, \ldots, n$, we can say that the carry c_{n+r} and c'_{n+r} differs with probability $\frac{1}{2}$ (taking $k = 1$). Now, we apply Theorem 2, where $i = n+r$, $p_i = \Pr(a_i = a'_i) = 1$, $q_i = \Pr(b_i = b'_i) = 0$ and get $\Pr(c_{n+r+1} = c'_{n+r+1}) = \frac{1}{2}$. So, for $k = 1$ the result is proved. For the next bits, since $b_i = b'_i$, we can apply the technique of Lemma 1 and get the result. □

4 Few points about the proof

In the next three sections, we provide theoretical explanation of respectively the bias for Salsa, bias for ChaCha and a probabilistically neutral bit. In this procedure we track the propagation of the difference at each round along with the corresponding probability. Before proceeding to the proof, in this section we declare several facts regarding our methods of the proofs.

(1) The events $(X_i[j] = X'_i[j])$ and $(\Delta_i[j] = 0)$ are basically the same. Though it is a very clear observation, we mention it for the convenience of the reader since both the notations have been used in our proofs. Such events appear repeatedly in our proofs. For convenience we denote the probability of such event by $\mathcal{P}_{(i,j)}$. In general, for any two 32-bit numbers a and a', by $\mathcal{P}_{(a,j)}$ we denote the probability $\Pr(a_j = a'_j)$, i.e., the probability that jth bit of a and a' bit are equal.
(2) Initially, $\mathcal{P}_{(7,31)} = 0$ and for any other bit (i, j), $\mathcal{P}_{(i,j)} = 1$. Now, it is expected that as the algorithm progresses, the correlation between X and X' decreases, i.e., the probability $\mathcal{P}_{(i,j)}$ comes closer and closer to 0.5 and eventually becomes 0.5. So, one important point is that our aim is to provide theoretical justification only wherever we observe

some bias. This means, in the tracking of the difference propagation, if for any bit we observe that the unbiasedness is already attained, obviously we do not attempt to justify this probability mathematically. For example, suppose we experimentally observe that after a few rounds of Salsa or ChaCha, for some bit (i, j), $\mathcal{P}_{(i,j)} = \frac{1}{2}$, i.e., there is no bias. Therefore we do not attempt to prove this probability theoretically.

(3) While computing the probability of an event, unless the probability is a rational number, we write it up to two decimal places or at most three decimal places (in few cases). Only in some cases where the probability is a rational number we go beyond three decimal places to write the exact value.

(4) We prove our results based on the mathematical results proved in Sect. 3, which are based on the assumption of independence and randomness of the numbers. In reality, for both Salsa and ChaCha, though in the initial matrix the independence and randomness is true, as the algorithm progresses, we can't claim the independence to hold in every step. However, in our proofs wherever we use these results, we assume this independence. In most of the cases the theoretical results match very well with the experimental observations. In a few instances where this assumption of independence did not give very accurate results, we prove the results in a different way (for example, Lemma 5, ChaCha).

(5) There are several cases where multiple results have exactly similar proofs. In such cases we avoid repeating the detailed proof and just refer to the theorem with the similar proof.

(6) We explain here the meaning of a few terms or phrases which we use in our proofs.

- "Containing a difference" when we say that the cell/word X_i contains a difference at position j, we mean that at that stage $X_i[j] \neq X'_i[j]$, or $X_i[j] \oplus X'_i[j] = 1$.
- "Transmission/transmit" when a cell/word is updated, if any of the cells involved in the procedure contains a difference, a difference can generate in the updated cell also. We call this the "transmission" of difference. For example: suppose X_a is updated as: $X_a = X_a \oplus X_b$. Now suppose, before the update, $X_a = X'_a$, but the pth bit of X_b is different from and X'_b. Then after the operation, the updated X_a and X'_a will have a difference at the pth bit. In such case we say that the difference is transmitted from X_b to X_a.
- "Propagation of difference" suppose we add two cells/words, say X_a and X_b. Now, $X_a = X'_a$. Between X_b and X'_b, all the bits except the pth bit are equal, but the pth bit is different. In the sum $X_a + X_b$ and $X'_a + X'_b$, the pth bit would surely be different. But due to the carry generated in the sum, the $(p + 1)$th can also be different. We call this "propagation of difference".

5 Proof of the forward bias of Salsa

In this section we prove the forward bias for Salsa, i.e., the probability that with the input difference at position $(7, 31)$, after four rounds the bit $(1, 14)$ of X and X' matches, which is experimentally observed to be approximately 0.56.

5.1 First round

In the first round, at first we calculate $\mathcal{P}_{(15,0)} = \Pr(X_{15}[0] = X'_{15}[0])$. In that context, we have the following result.

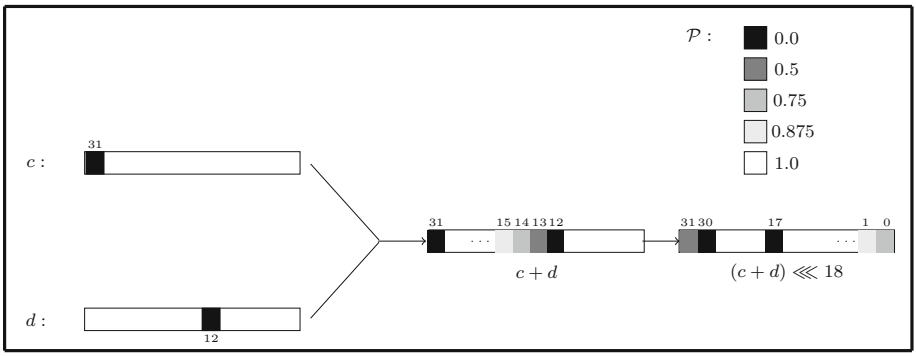

Fig. 1 Differential propagation during the operation $(c + d) \lll 18$

Lemma 3 *After the completion of the first round, the probability that the 0th bit of* X_{15} *and* X'_{15} *matches is* 0.75.

Proof In the columnround operation on the last row of X, we have:

$$(a, b, c, d) = (X_{15}, X_3, X_7, X_{11}).$$

The input difference is given in the 31st bit of X_7. Since in the quarterround function b (here X_3 and X'_3) is updated first, and c (here it is X_7) is not involved in the update function of b, there is no difference between X_3 and X'_3 after the update, i.e., $X_3 = X'_3$.
In the next step, c (X_7 and X'_7) is updated. It involves the variables a and b, which are X_{15} and X_3, respectively. Both of them are the same for X and X' so far. Therefore, these variables do not bring any difference between X and X'. The only difference is caused by the XOR of c, which is X_7. So, after the update X_7 and X'_7 have only one difference, which is at the 31st bit.
$d = d \oplus ((c + b) \lll 13)$ in the update of d, the involved variables are d, c, and b. Now, $d = X_{11}$ and $b = X_3$ do not have any difference between X and X'. c has difference at the 31st bit. So, in the addition $c + b$, exactly one difference is there between X and X', which is at the 31st bit. After the rotation by 13 bits, that difference comes at the 12th bit. So, X_{11} and X'_{11} has only one bit difference, which is at the 12th bit.
$a = a \oplus ((d + c) \lll 18)$ in the final update, a has no difference between X and X'. d has a difference at the 12th position, and c has difference at 31st. In the sum $c + d$, difference will be at the 12th and 31st bit. But, since this is an addition operation, the difference at the 12th bit may propagate to the next bit. The probability of this propagation we calculate using Lemma 1. Here, $n = 12, k = 2, n + k = 14$. So, the probability of the propagation is $\frac{1}{2^2} = \frac{1}{4}$. After left rotation by 18 bits, this difference comes to the 0th bit. So,

$$\mathcal{P}_{(15,0)} = \Pr(X_{15}[0] = X'_{15}[0]) = 1 - \frac{1}{4} = 0.75.$$

□

In the following Fig. 1, we present how the difference propagates during the operation $(c + d) \lll 18$. Let us denote the probability $\Pr(c_j = c'_j)$ by $\mathcal{P}_{c,j}$, $\Pr(d_j = d'_j)$ by $\mathcal{P}_{d,j}$ and $\Pr((c + d)_j = (c' + d')_j)$ by $\mathcal{P}_{c+d,j}$. The two blocks on the left represents the 32-bit numbers c and d. Each box represents a bit position. The probability of the equality of a bit is represented by the color of its corresponding box. The bit positions where there is no difference, i.e., the probability of equality ($\mathcal{P}$) is 1, are colored white. The bits in which there

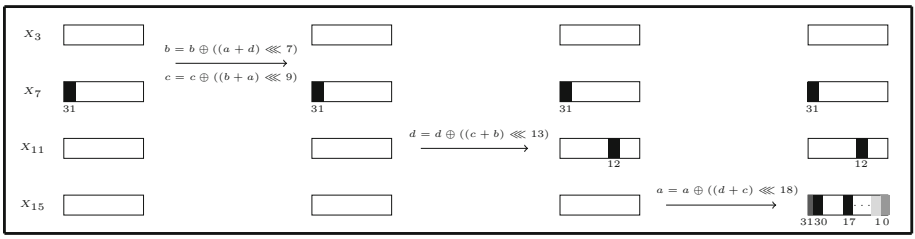

Fig. 2 Differential propagation in the first round on the last column of Salsa

is a difference, i.e, $\mathcal{P} = 0$, are colored dark black. Bits where the probability is between 0 and 1, are represented by grey of different darkness. As the value increases, the darkness of the grey gradually decreases, turning towards white. The value of $\mathcal{P}$ for each color is given in the figure.

Initially, only the 31st bit of c and 12th bit of d contains the difference. So, those positions are colored black. All the other bits are white. After the addition, the difference is there in the 31st and 12th bit of $(c + d)$, i.e., $\mathcal{P}_{c+d,j} = 0$ for $j = 12, 31$. Moreover, the difference at the 12th bit propagates to the next bits with probability $\frac{1}{2^k}$. So, $\mathcal{P}_{c+d,12+k} = 1 - \frac{1}{2^k}$, which has been represented up to 3 bits in the picture by the grey colors of different darknesses. In the next step the rotation takes place and each of the bits rotated by 18 bits to the right.

Figure 2 represents the propagation of the difference during the first round of Salsa. The four blocks in each column represents $(X_3, X_7, X_{11}, X_{15})$. The initial difference is at $(7, 31)$ which is colored black. The first column in the figure represents the initial situation. The second column represents the situation after first two steps of quarterround. The third and the last column represents the situation after third and fourth step respectively. In the last block of the last column, the dots represent the fact that the propagation has not stopped at the 1st bit, rather continues to the next bits. The colors of the boxes represent the value of the probability $\mathcal{P}$ as already given in the Fig. 1.

5.2 Second round

In the second round, we prove a few small but important results which we will use later.

Lemma 4 *After the completion of the second round, we have the following results*:

(1) $\mathcal{P}_{(11,0)} = 1$.
(2) $\mathcal{P}_{(4,i)} = 1$, *for all* $i < 12$.
(3) $\mathcal{P}_{(12,24+i)} = 1 - \frac{1}{2^i}$, *for* $0 \leq i \leq 7$. *Also*, $\mathcal{P}_{(12,i)} = 1 - \frac{1}{2^{8+i}}$, *for* $i = 0, 1$.

Proof (1) In the third row, we have $(a, b, c, d) = (X_{10}, X_{11}, X_{12}, X_{13})$. In the first step, we have:

$$b = b \oplus ((a + d) \lll 7).$$

Here, a and d have no difference in X and X', because after the first round, the propagation of the difference is within the fourth column only. So, the only difference can be produced by the XOR of b, which has only one difference at the 12th bit. Therefore, the 0th bit is not affected by this operation.

(2) In the second row, we have $(a, b, c, d) = (X_5, X_6, X_7, X_4)$. In the first round, the cell X_4 is not affected by the propagation of the input difference, since the propagation stays

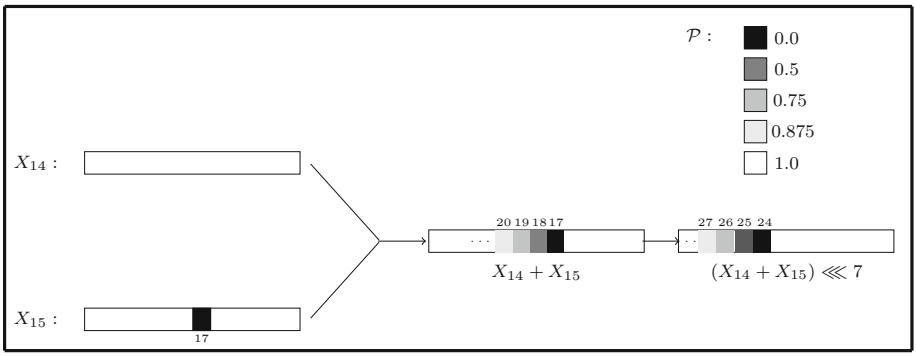

Fig. 3 Differential propagation during the operation $(X_{14} + X_{15}) \lll 7$ according to Lemma 4

within the fourth column only. In the second round, the first step is $X_6 = X_6 \oplus ((X_5 + X_4) \lll 7)$. This does not involve X_7, so there is no propagation of bias. Now

- $X_7 = X_7 \oplus ((X_6 + X_5) \lll 9)$: only X_7 contains a difference at 31st bit. So, after the update, the difference would be there at the 31st bit only.
- $X_4 = X_4 \oplus ((X_6 + X_7) \lll 13)$: in this operation, since X_7 has a difference at the 31st bit, it brings difference in the 12th bit of X_4 (because of the left rotation by 13 bits). The bits on the left side of the 12th bit are also affected probabilistically because of the carry propagation. But, the bits which are on the right side of 12th bit are not affected by the difference by this operation. Therefore, even after completion of the second round, we have $X_4[i] = X'_4[i]$, for all $i < 12$.

(3) In the first round, $X_{12} = X'_{12}$. In the second round, it is updated by the function:

$$b = b \oplus ((a + d) \lll 7),$$

where a, d are respectively X_{15} and X_{14} from the 1st round. The 17th bit of X_{15} contains a difference. On the addition $X_{15} + X_{14}$, this difference come to the 17th bit of the sum and propagates to the next bits with the probability $\frac{1}{2^i}$. So, using Lemma 1, for the probability that the $(17 + i)$th bit of the sum $X_{15} + X_{14}$ would be equal to that of $X'_{15} + X'_{14}$ is $1 - \frac{1}{2^i}$. These bits gets rotated by 7 bits and XORed with X_{12}. So, the 17th bit shifts to the 24th bit. The 25th bit shifts to the 0th bit. Therefore the result follows. A pictorial representation of the sum and rotation of $(X_{14} + X_{15} \lll 7)$ is given in Fig. 3. □

5.3 Third round

Now we provide the proofs of some biases that we observe in the third round. In these proofs, we are going to use the results obtained in the first and second round. The third round is a columnround. We prove a few results for the first, second and fourth column.

5.3.1 First column

In this column, the tuple (a, b, c, d) is (X_0, X_4, X_8, X_{12}). We have the following results.

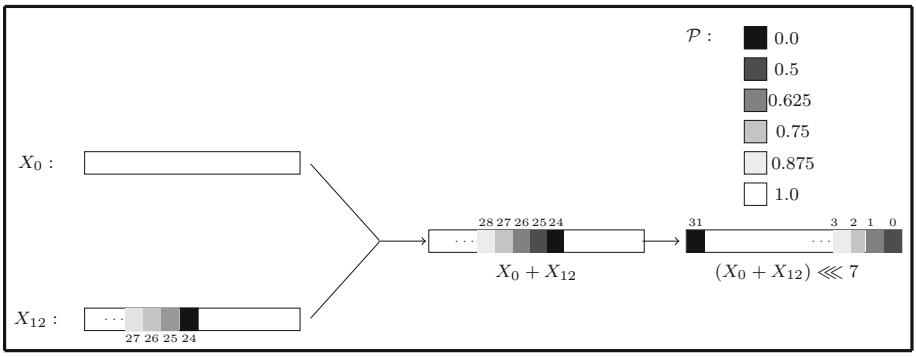

Fig. 4 Differential propagation during the operation $(X_0 + X_{12}) \lll 7$ according to Theorem 4

Theorem 4 *After the third round,*

$$\mathcal{P}_{(4,i)} = \begin{cases} \frac{1}{2} & for\ i = 0, \\ \frac{5}{8} & for\ i = 1, \\ 1 - \frac{1}{2^i} & for\ 2 \le i \le 8. \end{cases}$$

Proof In the third round X_4 is updated by the function

$$X_4 = X_4 \oplus ((X_0 + X_{12}) \lll 7),$$

where X_4, X_0 and X_{12} are from the second round. We focus on the sum $(X_0 + X_{12})$.

Now, since X_0 is on the first row, it is not affected by the propagation up to the 2nd round because after the first round no cell of the first row is affected by the difference. Therefore we only focus on X_{12}. From Lemma 4 in the second round, we know, $\mathcal{P}_{(12,24+i)} = 1 - \frac{1}{2^i}$. In the sum $X_0 + X_{12}$, for some $(24+i)$th bit $(i \ge 0)$ we try to find $\Pr((X_0 + X_{12})[24+i] \ne (X'_0 + X'_{12})[24+i])$. Let us call this event A_i. For X_{12}, let us call the event E_j that the difference has propagated up to the $(24+j)$th bit and not further. So, E_j's are disjoint and $\Pr(E_j) = \frac{1}{2^j} - \frac{1}{2^{j+1}} = \frac{1}{2^{j+1}}$.
So,

$$\Pr(A_i) = \sum_{j=0}^{i-1} \Pr(A_i|E_j) \cdot \Pr(E_j) + \Pr(A_i|\big(\bigcup_{j\ge i} E_j\big)) \cdot \Pr\big(\bigcup_{j\ge i} E_j\big).$$

From Lemma 1, $\Pr\left(\bigcup_{j\ge i} E_j\right) = \frac{1}{2^i}$. From Lemma 2, $\Pr(A_i|E_j) = \frac{1}{2^{j-i}}$ and $\Pr\left(A_i|\bigcup_{j\ge i} E_j\right) = \frac{1}{2}$. Thus we get,

$$\Pr(A_i) = \begin{cases} \frac{1}{2^i} & \text{for } i = 0, 1, \\ \frac{3}{8} & \text{for } i = 2, \\ \frac{1}{2^{i-1}} & \text{for } i > 2. \end{cases}$$

Now, we compute $\Pr(A_i^c)$ for all three cases. Then, after rotation by 7 bits and XOR with X_4, these probabilities transmit to the positions $X_4[0], \ldots, X_4[8]$ respectively and we get the result. □

Theorem 5 *After the 3rd round*, $\mathcal{P}_{(12,21)} \approx 0.023$.

Proof X_{12} is updated by the function:

$$d = d \oplus ((c + b) \lll 13),$$

where c and b are respectively X_8 and X_4. Since the rotation is by 13 bits, $X_{12}[21]$ is basically the XOR of $X_{12}[21]$ (after the second round) and 8th bit of $(X_8 + X_4)$. Here, X_8 and X_4 are already updated up to the third round.

$X_{12}[21]$ *after the second round* this cell is unaffected in the first round. In the second round it is updated by

$$b = b \oplus ((a + d) \lll 7),$$

where a, d are X_{15} and X_{14}, respectively. So, we focus on the 14th bit of $X_{15} + X_{14}$. The only difference can be created by the propagation of the difference of X_{15} at 0th bit (Lemma 3) and next few bits, whose effect is negligible at 14th bit. Therefore, the probability of difference is approximately 0. So, after the second round, $\mathcal{P}_{(12,21)} \approx 1$.
$(X_8 + X_4)[8]$ *after third round* up to the 2nd round, the 8th bit of X_8 and X_4 are not influenced by the difference. In the third round, X_8 is XORed with $X_4 + X_0$. Here, the 31st bit of X_4 is different from X'_4 with probability 1, because it was XORed with X_7 in the previous step. Therefore, 8th bit of X_8 differs from X'_8 with probability almost 1. This means, $\mathcal{P}_{(8,8)} = \Pr(X_8[8] = X'_8[8]) \approx 0$. On the other hand, the probabilities $\mathcal{P}_{(4,i)}$ for $0 \leq i \leq 8$ we have from Theorem 4. We ignore the carry difference coming from the bits on the right side of the 4th bit. So, here we apply Theorems 2 and 3. We have $p_i = \mathcal{P}_{(8,i)}$ for $i < 8$ and $p_8 = \mathcal{P}_{(8,8)} = 0$. On the other hand, $q_i = \mathcal{P}_{(4,i)} = 1 - \frac{1}{2^i}$ for $4 \leq i \leq 8$. So, as in Theorem 3, $s = X_4 + X_8$ and $s' = X'_4 + X'_8$. Now, using Theorem 2 we compute $\Pr(c_5 \neq c'_5)$. In this context, since we ignore the carry differences coming from the bits on the right side of the 4th bit, we assume $\Pr(c_4 = c'_4) = 1$. After computing this, we use this result to find $\Pr(c_6 \neq c'_6)$ again using Theorem 2. In this way we proceed upto computing $\Pr(c_8 \neq c'_8)$. Then we use Theorem 3 to compute $\Pr(s[8] \neq s'[8])$, which comes to be 0.023. We do not show the step by step calculation here. □

Theorem 6 *After the completion of the third round*, $\mathcal{P}_{(0,7)} \approx 0.94$.

Proof In the third round, X_0 is updated by the function:

$$a = a \oplus ((c + d) \lll 18),$$

where c and d are X_8 and X_{12}, respectively which are already updated in the third round. Since the rotation is by 18 bits, the 7th bit of updated X_0 can be given by: $X_0[7] \oplus (X_8 + X_{12})[21]$. Now, $X_0 = X'_0$ up to the second round. So, the difference in the updated X_0 and X'_0 can be caused only by the difference of $(X_8 + X_{12})[21]$ and $(X'_8 + X'_{12})[21]$. The probability of the equality of these two bits can be found by partitioning it into two events:

(1) If there is no difference of the carry from previous bit.
(2) If there is difference in carry.

In the first case, the probability of equality can be found by Theorem 1, where $p = \mathcal{P}_{(12,21)} = 0.023$ (Theorem 5). This value is approximately 0.975.

Next we focus on $P_{(8,21)}$ to get q. In the second round, this bit is updated as $d = d \oplus ((b + c) \lll 13)$, where b is X_0 and c is X_4. Since X_0 is not influenced by the difference so far, we focus on 8th bit of X_4 (since the rotation is by 13 bits). This bit has a direct influence of the

difference given in 31st bit of X_7, which after rotation by 9 bits, reaches this position. So, this bit is different with probability 1. So, in the 2nd round, $P_{(8,21)} = 0$. In the third round, one can check that the term $(b + c)$ has influence over this position with very negligible probability. Therefore, the probability q is approximately 0.
In the second case, the probability can be found by Theorem 3, which gives the probability as approximately 0.
Next we find the probability of the difference of the carry. Though there are 20 bits on the right side, we ignore the 19 bits of the right side since their influence is negligible and consider only the 20th bit. So, the probability of equality of the carries can be computed by Theorem 2, where $p = 0.996, q = 0.95$. Using the formula, we have the probability of equality of the carries approximately as 0.97.
So, the total probability: $0.97 \times 0.975 + 0.03 \times 0 \approx 0.94$. □

5.3.2 Second column

In this column, the tuple (a, b, c, d) is (X_5, X_9, X_{13}, X_1). We have one result in this column that we are going to use in the proof of the final theorem in the fourth round.

Theorem 7 *After the completion of the third round*, $\mathcal{P}_{(1,14)} = 0.96$.

Proof In third round, X_1 is updated by

$$d = d \oplus ((b + c) \lll 13).$$

Here, up to the second round, X_1 (or d in the second round) is not influenced by the difference. b and c are respectively X_9 and X_{13} from the 3rd round. So, we focus on the 1st bit of their sum (since the rotation is by 13 bits). Now, X_{13} is updated in the 2nd round by $c = c \oplus ((a+b) \lll 9)$. In that update, on the 24th bit we focus, due to the rotation. Since the 19th bit contains a difference during its update (can be checked very easily), this bit is affected with probability $\frac{1}{2^5}$ (Lemma 1). So, the probability of equality is $1 - \frac{1}{2^5} \approx 0.97$. The differences in the addition $a + b$ in the third round do not bring any significant change in this probability.

By similar argument one can very easily check that $\mathcal{P}_{(9,1)} \approx 0.99$. Therefore, the probability of the equality of 1st bit of $(X_9 + X_{13})$ can be given by $0.97 \times 0.99 \approx 0.96$ (here we ignore the probability of the difference in the carry from the previous bit, since it is negligible). Therefore, the final probability of the given event is also 0.96. □

5.3.3 Fourth column

Theorem 8 *After the third round*, $\mathcal{P}_{(3,7)} = 0.75$.

Proof $(a, b, c, d) = (X_{15}, X_3, X_7, X_{11})$. In the first step, X_3 is updated by the function $b = b \oplus ((a + d) \lll 7)$. After the first round, the first row is not influenced by the difference. Therefore in the second round, the rowround operation on the first row also does not bring any difference. So, X_3 and X'_3 are exactly the same after the second round. Also, from Lemma 4, we know that the 0th bit of X_{11} and X'_{11} are equal after the second round. Therefore, the difference at 0th bit of $a + d$ occurs iff the 0th bit of a has the difference. Therefore, the probability of the equality in the least significant bit of $a + d$ for X and X' is equal to the difference at the least significant bit of a i.e., $\Pr(X_{15}[0] = X_{15}[0])$, which is 0.75, from Lemma 3. Therefore, this probability is 0.75.

Now, after left rotation by 7 bits, this bit shifts to the 7th bit. And this is XORed with b, which does not have any difference. Therefore, the probability that the 7th bit of the updated b is equal for X_3 and X_3' is 0.75. □

5.4 Fourth round

Now we prove the final result which gives us the single bit output bias observed in the fourth round of Salsa.

Theorem 9 *After the fourth round*, $\mathcal{P}_{(1,14)} \approx 0.56$.

Proof In the rowround on the first row of X, we have:

$$(a, b, c, d) = (X_0, X_1, X_2, X_3).$$

So, $b = X_1$ is updated at first. We focus on the 14th bit of X_1 and X_1'. Let us denote the 14th bit of b as $b[14]$. So $b[14] = b[14] \oplus ((a + d)[7])$ (since the 7th bit of $a + d$ is left shifted by 7 bits and reaches the 14th bit position).

Let us first focus on the 7th bit of $a + d$. The probability of the equality of this bit can be computed for two separate cases:

(1) If there is no difference in the carry from the previous bit.
(2) If there is difference in the carry.

Event 1 if there is no difference in the carry, the probability of equality at $(a + d)[7]$ can be calculated by the formula derived from Theorem 1, which is $pq(1 - p)(1 - q)$. Here, $p = 0.94$ (Theorem 6). In the 3rd round, X_3 is updated by $b = b \oplus ((a + d) \lll 7)$. So, the 7th bit of X_3 can be given by XOR of the previous X_3 and the 0th bit of X_{11} and X_{15}. Since the 0th bit of X_3 and X_{11} are unaffected by the difference up to the second round, so the difference can only transmit from the difference at X_{15}. Now, the 0th bit of X_{15} possesses equality with probability 0.75 after the first round (Lemma 3). It remains the same after the second round since in this rowround all the other key cells involved are not influenced by the difference. So, the probability is $q = 0.75$. So, the probability of the equality if there is difference in the carry is $(0.94 \times 0.75 + 0.06 \times 0.25) = 0.72$.
Event 2 if there is difference in the carry, the probability can be calculated by the formula from Theorem 3, which is $p(1-q)+q(1-p)$. So, the probability is: $(0.94 \times 0.25 + 0.06 \times 0.75) = 0.28$.

Now, we find the probabilities of events 1 and 2. Here, we take into consideration the carry coming from the immediate previous bit (6th bit) and ignore the bits which are prior to the 6th bit. We experimentally observe that in d, the 6th bit does not possess any bias, i.e., $\Pr(X_3[6] = X_3'[6]) = \frac{1}{2}$. Therefore, as already mentioned in the beginning of this section, we do not attempt to prove this probability. Rather we use this result directly in our proof. Suppose q_6 is the probability that the 6th bits of d (X_3 and X_3') are equal. So, $q_6 = \frac{1}{2}$. Therefore, we can use Theorem 2 here. By a similar argument as in the proof of $\mathcal{P}_{3,7} = 0.94$, it can be proved that $p_6 = \mathcal{P}_{3,6} = 0.9$. Since we ignore the bits on the right side of of the 6th bit, $\Pr(c_6 \neq c_6') = 0$. Therefore, according to the formula of Theorem 2, the probability that there is difference in the carry in the 7th bit is: $\Pr(c_7 \neq c_7') = \frac{1}{2} - \frac{0.9}{4} \approx 0.28$. And the probability that there is no difference in the carry is $1 - 0.28 = 0.72$. So, calculating the total probability, we get 0.59.

Next, this bit is XORed with $b[14]$. The probability that $b[14]$ differs is 0.96, from Theorem 7. So, the final probability can be given by the formula $(pq+(1-p)(1-q))$ (Theorem 1), where $p = 0.96$ and $q = 0.59$. So, we have the probability approximately as 0.56. □

Table 1 Comparison between theoretical and experimental results

Output bits	Theoretical result	Experimental result	Chosen IV experimental result [14]
$X_{15}^1[0]$	0.75	0.75	1.0
$X_4^3[8]$	0.998	0.996	0.997
$X_{12}^3[21]$	0.023	0.022	0.021
$X_0^3[7]$	0.94	0.95	0.95
$X_1^3[14]$	0.96	0.98	0.98
$X_3^3[7]$	0.75	0.74	0.93
$X_1^4[14]$	0.56	0.56	0.62

Therefore, now we have the detailed theoretical structure of this bias. In Table 1, we provide a comparison between the probabilities achieved theoretically and by experiment side-by-side in second column and third column, respectively. We perform experiment over 10,000 random key-IV pairs to produce this result. In [4], a special technique has been used to construct multiple bit output bias in the fifth round from a single bit output bias of the fourth round. The technique of this construction itself is the theoretical justification of this extension of distinguisher in the 5th round. Therefore for the 5th round distinguisher, no further theoretical justification is required. However, the position of the input difference and the observed output bit was different in that work. They gave the input difference at the 0th bit of X_7. The single bit output difference in fourth round they observed at (1, 13), i.e., 13th bit of X_1. From that they constructed a multibit output difference in the fifth round by the XOR of $X_1[13]$, $X_9[0]$, $X_9[0]$. The theoretical explanation of that work can also be explained similarly in this way.

Chosen IV cryptanalysis in [14], Maitra improved the bias for the same input–output pair by suitably choosing the IV's. Using this idea, he improved the probability $\Pr(X_1[14] = X_1'[14])$ to 0.62. This probability can also be theoretically proved by our approach. The tracking of the biases will be exactly similar, the only difference being that the biases would be higher in this case. For example, in the first round, the probability $\Pr(X_{15}[0] = X_{15}'[0])$ would become 1, instead of 0.75 (Lemma 3). The propagation of the difference produced by the carry is restricted by suitably choosing the IV so that the propagation does not occur. In this way, all the proofs in the propagation of difference can be given by the similar approach as in our paper and can be shown that the probability becomes 0.62. In Table 1, we provide the experimental probabilities based on 10,000 random key-IV pairs for chosen IV in the last column, which gives a clear idea about how the difference propagates.

6 Proof of the forward bias of ChaCha

For ChaCha, the input difference is given at the position (13, 13) and the output difference is observed after 3 rounds at the position (11, 0). The observed probability is 0.51. In the similar manner as Salsa, we follow the propagation of the bias in each round and prove them.

6.1 First Round

Lemma 5 *After the completion of the first round, we have the following results.*

(1) $\mathcal{P}_{(9,29+i)} = 1 - \frac{1}{2^i}$, *for* $i = 0, 1, 2$,
(2) $\mathcal{P}_{(13,20+i)} = 1 - \frac{1}{2^{2+i}}$, *for* $0 \leq i \leq 11$,
(3) $\mathcal{P}_{(5,4+i)} \approx 1 - \frac{1}{2^i}$, *for* $i = 0, 1, 2$,
(4) $\mathcal{P}_{(5,24)} \approx 0.0$,
(5) $\mathcal{P}_{(5,27)} \approx 0.66$,
(6) $\mathcal{P}_{(5,28)} \approx 0.78$,
(7) $\mathcal{P}_{(5,12)} \approx 0.0$.

Proof Since the initial difference is at the second column at position (13, 13), after the first round only the second column gets affected. Now, in the quarterround function, we have:

$$(a, b, c, d) = (X_1, X_5, X_9, X_{13}).$$

In the first addition between X_1 and X_5, no term containing any difference is involved. Then, in the operation $X_{13} = ((X_{13} \oplus X_1) \lll 16)$, the difference shifts to the 29th bit of X_{13} due to the rotation. Afterwards, we track the propagation of the difference step by step.
$X_9 = X_9 + X_{13}$ the 29th bit of X_9 receives the difference from X_{13}. Also, the difference propagates to the next bits of X_9 with probability $\frac{1}{2^k}$ as given in Lemma 1. However, the probabilities of differences at these three positions are not independent. Any one of these would possess the difference only if the previous one do, since the difference can not propagate to the next bit if the previous bit does not have difference. So, we can't use Theorem 2 or 3 directly here. So we prove it as follows:

If we consider the differences at the three positions as a tuple ($\Delta_9[31]$, $\Delta_9[30]$, $\Delta_9[29]$), it can have only three possible values: (0, 0, 1), (0, 1, 1) and (1, 1, 1). The probabilities of occurrences of these values are as follows:

- (0, 0, 1) : $\frac{1}{2}$, since it occurs whenever the difference at the 29th bit does not propagate to the 30th bit, i.e, $(c_{30} = c'_{30})$.
- (0, 1, 1) : $\frac{1}{4}$. This occurs when $(c_{30} \neq c'_{30})$ but $(c_{31} = c'_{31})$. So, the difference propagates from the 29th to the 10th bit, which has probability $\frac{1}{2}$. And then the difference does not propagate to the 11th bit, which has again probability $\frac{1}{2}$. Since the values at the 30th bit are independent of the values at 29th bit, the probability is $\frac{1}{4}$.
- (1, 1, 1) : $\frac{1}{4}$. When both $(c_{30} \neq c'_{30})$ and $(c_{31} \neq c'_{31})$, i.e., the difference propagates from the 29th to the 30th and then to the 31st. Therefore, again the probability is $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$.

Therefore, we have: $Pr(X_9[29 + k] = X'_9[29 + k]) = 1 - \frac{1}{2^k}$ for $k = 0, 1, 2$.
$X_5 = (X_5 \oplus X_9) \lll 12$ due to the XOR and rotation, the differences are formed at the 9th, 10th and 11th bit of X_5 with the same probability, i.e., $\Pr(X_5[9+k] = X'_5[9+k]) = 1 - \frac{1}{2^k}$ for $k = 0, 1, 2$.

$$\Pr(X_5[9 + k] = X'_5[9 + k]) = 1 - \frac{1}{2^k} \quad \text{for } k = 0, 1, 2. \tag{1}$$

The propagation of the difference up to this stage, i.e., the half round of the first round of ChaCha is represented in Fig. 5. The colors of the boxes represent the value of the probability $\mathcal{P}$ as already given in Fig. 1.
$X_1 = X_1 + X_5$ X_1 is not yet influenced by the difference. On the other hand, X_5 has been affected by the difference on the 9th, 10th and 11th bit as we have just proved.

If we consider the differences at these three positions as a tuple ($\Delta_5[11]$, $\Delta_5[10]$, $\Delta_5[9]$), it can have only three possible values: (0, 0, 1), (0, 1, 1) and (1, 1, 1). We denote these three

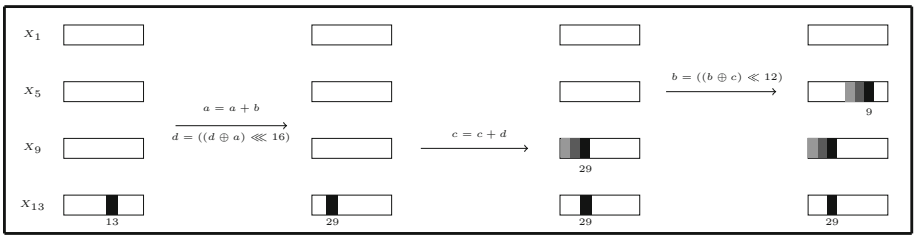

Fig. 5 Differential propagation in the first half round of ChaCha

events as E_1, E_2, E_3. The probabilities of occurrences of these values are as follows:

$$\left.\begin{array}{ll}(0,0,1): & \text{Probability=0.5,}\\ (0,1,1): & \text{Probability=0.25,}\\ (1,1,1): & \text{Probability=0.25.}\end{array}\right\} \tag{2}$$

Our aim is to find the probability of the updated $\Delta_1[i] = 0$ for $i = 9, 10, 11$ and 12. It is clear that $\Pr(\Delta_1[9] = 0) = 0$. Now, $(\Delta_1[12], \Delta_1[11], \Delta_1[10])$ can have $2^3 = 8$ possible values. For each of the three events E_1, E_2 and E_3 in Eq. (2), we find the probability for each of the eight possible values of the tuple $(\Delta_1[12], \Delta_1[11], \Delta_1[10])$. Let us denote them by $a_1, a_2, \ldots, a_8$ and the corresponding events that $(\Delta_1[12], \Delta_1[11], \Delta_1[10]) = a_i$ by A_i. Now, for each of the 8 values, we compute $\Pr(A_i)$ as $\sum_{j=1}^{3} \Pr(E_j) \cdot \Pr(A_i|E_j)$. We do not show the detailed calculations here, but directly write the result as given below (Fig. 4).

$$\left.\begin{array}{ll}0.3437 & \text{for } (0,0,0)\\ 0.2187 & \text{for } (0,0,1)\\ 0.0625 & \text{for } (0,1,0) \text{ and } (1,1,0)\\ 0.125 & \text{for } (0,1,1) \text{ and } (1,1,1)\\ 0.0313 & \text{for } (1,0,0) \text{ and } (1,0,1)\end{array}\right\} \tag{3}$$

From this, we get can the total probability of occurrence $\Delta_1[i] = 0$ for $i = 10, 11, 12$ by adding the probabilities of the suitable tuple values. So, we have: $\Pr(\Delta_1[i] = 0) = 0, 0.5, 0.625$ and 0.75 for $i = 9, 10, 11, 12$, respectively. From $i = 12$ onwards, since there is no more differences in $X_5[i]$, so the difference propagates according to Lemma 1. So $\Pr(\Delta_1[12 + i] = 0) = 1 - \frac{1}{2^{2+i}}$ for $i > 0$.

$X_{13} = (X_{13} \oplus X_1) \lll 8$ the corresponding bits (which are discussed above for X_1) of X_{13} did not have any influence of difference before. So, they receive the same probability as in X_1. After that, the bits are rotated by 8 bit positions. So, we have: $\Pr((\Delta_{13}[17]) = 0) = 0$, and the probabilities for all possible values of the tuple $(\Delta_{13}[20], \Delta_{13}[19], \Delta_{13}[18])$ are the same as $\Pr(\Delta_1[12], \Delta_1[11], \Delta_1[10])$ as given in the last proof (Eq. (1)). Therefore, we have $\mathcal{P}_{(13,18)} = 0.5$, $\mathcal{P}_{(13,19)} = 0.625$ and $\mathcal{P}_{(13,20+i)} = 1 - \frac{1}{2^{2+i}}$.

Also, the difference at the 29th bit of X_{13} reaches the 5th bit by rotation. So, after this step,

$$\mathcal{P}_{(13,5)} = 0. \tag{4}$$

$X_9 = X_9 + X_{13}$ we already know the probabilities of the possible values of the tuple $(\Delta_9[31], \Delta_9[30], \Delta_9[29])$ and therefore we have $\Pr(X_9[29 + k] = X'_9[29 + k]) = 1 - \frac{1}{2^k}$ for $k = 0, 1, 2$. Since the corresponding bits of X_{13} are not yet influenced by the difference propagation, in the updated X_9, the probabilities remain the same. So,

$$\mathcal{P}_{(9,29+i)} = 1 - \frac{1}{2^i} \quad \text{for } i = 0, 1, 2. \tag{5}$$

Also, the difference at $X_{13}[17]$ transmits to $X_9[17]$ after this addition. For the bits 18 to 21, we calculate the probabilities for X_9 by the same method as in the last one. We achieve: $\mathcal{P}_{(9,20)} \approx 0.66$, $\mathcal{P}_{(9,21)} \approx 0.78$.

And the difference at (13, 5) transmits to the 5th bit of X_9. So, $\mathcal{P}_{(9,5)} = 0$.
$X_5 = (X_5 \oplus X_9) \lll 7$ the differences at the 29th, 30th and 31st bit of X_9 transmits to X_5 and then after getting rotated by 7 bits, move respectively to the 4th, 5th and 6th bit of X_5 with the same probabilities. So, the probability distribution of the tuple $(\Delta_5[4], \Delta_5[5], \Delta_5[6])$ for different values is same as the distribution of $(\Delta_9[29], \Delta_9[30], \Delta_9[31])$, which is basically similar to (2).

$$\mathcal{P}_{(5,4+i)} \approx 1 - \frac{1}{2^i}, \quad \text{for } i = 0, 1, 2.$$

The difference at the 17th bit of X_9 transmits to the 24th bit of X_5. And the difference at the 20th and the 21st bit of X_9 produces difference at the 27th and the 28th bit of X_5, respectively. Therefore, $\mathcal{P}_{(5,24)} \approx 0.0$, $\mathcal{P}_{(5,27)} \approx 0.66$, $\mathcal{P}_{(5,28)} \approx 0.78$. The difference at $X_9[17]$ transmits to X_5 and then after getting rotated by 7 bit, moves to the 24th bit of X_5.

The difference at (9, 5), after the XOR and rotation by 7 bits, transmit to (5, 12). Therefore $\mathcal{P}_{(5,12)} = 0$. □

6.2 Second round

Theorem 10 *After the second round of ChaCha we have* $\mathcal{P}_{(3,16)} \approx 0.99$, $\mathcal{P}_{(3,15)} \approx 0.97$.

Proof We will prove the result for $X_3[16]$ only and the other one follows exactly in the similar manner.

$$(a, b, c, d) = (X_3, X_4, X_9, X_{14}).$$

In the second round, in the diagonal (X_3, X_4, X_9, X_{14}), at first the following operations take place:

$$X_3 = X_3 + X_4, \quad X_{14} = ((X_{14} \oplus X_3) \lll 16).$$

Here, since none of the cells contains difference, there is no propagation of difference. The next step is:
$X_9 = X_9 + X_{14}$ since X_{14} is not yet affected, $\mathcal{P}_{9,29+k}$ remains the same, i.e, $1 - \frac{1}{2^k}$ for $k = 0, 1, 2$. After that, the operation is:
$X_4 = (X_9 \oplus X_4) \lll 12$ since X_4 previously had no difference, the only differences in the updated X_4 transmit from X_9. Due to the rotation by 12 bits, those differences shift to the 9th, 10th and 11th bit. Therefore $\mathcal{P}_{(4,i)} = \Pr(X_4[i] = X'_4[i])$ is approximately equal to 0, $\frac{1}{2}$ and $\frac{3}{4}$ for $i = 9$, 10 and 11 respectively.
$X_3 = X_3 + X_4$ after this addition operation, the probabilities of the differences we will calculate using Theorem 2. Let p_j and q_j in Theorem 2 are respectively $\mathcal{P}_{(3,j)}$ and $\mathcal{P}_{(4,j)}$. Then $p_j = 1$ for all j since X_3 is not influenced by the difference yet. So, the formula of Theorem 2 can be written as

$$\Pr(c_{i+1} \neq c'_{i+1}) = \Pr(c_i \neq c'_i) \cdot \frac{q_j}{2} + \Pr(c_i = c'_i) \frac{(1 - q_j)}{2}.$$

And we have $q_9 = 0$, $q_{10} = \frac{1}{2}$, $q_{11} = \frac{3}{4}$ and $\Pr(c_9 \neq c'_9) = 0$. Using this data, we compute at first $\Pr(c_{10} \neq c'_{10}) = \frac{1}{2}$, then $\Pr(c_{11} \neq c'_{11}) = \frac{1}{4}$, and then $\Pr(c_{12} \neq c'_{12}) = \frac{3}{16}$. From $j = 12$ to 16, q_j is also 1 because these are not influenced by the difference propagation.

So, calculating $\Pr(c_{16} \neq c'_{16})$ we get $\frac{3}{256}$. Since p_{16} and q_{16} are 1, by using Theorem 3 we can compute the probability of $X_3[16] = X'_3[16]$ to be $\frac{253}{256}$, which is approximately $0.988 \approx 0.99$.

Using the exactly same method we can prove that $\mathcal{P}_{(3,15)} \approx 0.97$. □

Theorem 11 *After the second round,* $\mathcal{P}_{(15,16)} \approx 0.67$.

Proof In the diagonal $(X_0, X_5, X_{10}, X_{15})$, only X_5 is influenced by the difference after the first round. In the first step of this diagonalround, we have the following operations.

$$X_0 = X_0 + X_5; \quad X_{15} = (X_{15} \oplus X_0) \lll 16.$$

The 24th bit of X_5 contains the difference (Lemma 5), which transmits to the 24th bit of X_0 after this addition. Then it reaches the 8th bit of X_{15} after the XOR and rotation. So, after this step,

$$\mathcal{P}_{(15,8)} = 0. \tag{6}$$

From Lemma 5 in the first round, we know that the 4th bit of X_5 differs from X'_5. Also the 5th and 6th bits of X_5 and X'_5 are equal with probability 0.5 and 0.75, respectively. If we consider as a tuple, $(\Delta_5[4], \Delta_5[5], \Delta_5[6])$ has the probability distribution exactly similar to (2) as follows:

$$\left.\begin{array}{ll} (0,0,1): & \text{Probability=0.5,} \\ (0,1,1): & \text{Probability=0.25,} \\ (1,1,1): & \text{Probability=0.25.} \end{array}\right\} \tag{7}$$

For all three cases, we compute the probability of $(X_0[i] = X'_0[i])$ for $i = 4$ to 8 and then find the total probability for each X_i. Those are as follows:

$$\begin{array}{ll} 0.625 & \text{for 6th bit,} \\ 0.750 & \text{for 7th bit,} \\ 0.875 & \text{for 8th bit.} \end{array} \tag{8}$$

This propagation has been shown in Fig. 6.
On the other hand the 12th bit of X_5 has a difference, which transmits to the same bit of X_0. So, $\mathcal{P}_{(0,12)} = 0$. After XOR with X_{15} and rotation, it transmits to the 28th bit of X_{15}. The next step is:

$$X_{10} = X_{10} + X_{15}; \quad X_5 = (X_5 \oplus X_{10}) \lll 12.$$

On addition with X_{10}, the difference transmits to the 28th bit of X_{10}.

From Lemma 5 in first round, we know that $\mathcal{P}_{(5,27)} = 0.66$, $\mathcal{P}_{(5,28)} = 0.78$. After the XOR and rotation, the difference carries to the 8th bit of X_5. And we have

$$\left.\begin{array}{l} \mathcal{P}_{(5,7)} = 0.66, \\ \mathcal{P}_{(5,8)} = 1 - 0.78 = 0.22. \end{array}\right\}$$
$$X_0 = X_0 + X_5; \quad X_{15} = (X_{15} \oplus X_0) \lll 16. \tag{9}$$

After this addition, we find the probability of the event $(\Delta_0[8] = 0)$. We consider one previous bit, i.e., the 7th bits of X_0 and X_5. We have the corresponding probabilities from Eqs. (8) and (9). Similar to the technique in the beginning of this theorem, we can say that there can be three disjoint events:

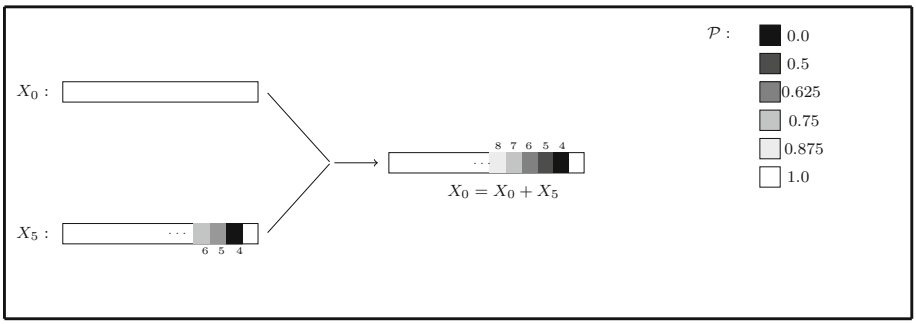

Fig. 6 One of the differential propagations during the operation $X_0 = (X_0 + X_5)$ according to Theorem 4

(1) $(\Delta_0[6] = 0$ and $\Delta_0[7] = 0)$: probability 0.75.
(2) $(\Delta_0[6] = 1$ and $\Delta_0[7] = 0)$: probability 0.125.
(3) $(\Delta_0[6] = 1$ and $\Delta_0[7] = 1)$: probability 0.125.

Computing the probability of equality of the 8th bit of the updated X_0 for each of these three cases and adding them, we get the final probability as 0.333. This, after the XOR with X_{16} and rotation by 8 bits, reaches the 16th bit of X_{16}. Since the 8th bit of X_{16} previously had difference (Eq. (6)), the final probability is $1 - 0.333 = 0.667 \approx 0.67$. □

Theorem 12 *After the second round*,

$$\mathcal{P}_{(7,i)} = \begin{cases} 0.94 & \text{for } i = 15, \\ 0.97 & \text{for } i = 16. \end{cases} \tag{10}$$

Proof In the diagonal (X_2, X_7, X_8, X_{13}), the first step is:

$X_2 = X_2 + X_7$: here, no cell are affected by the difference. The next step is:

$X_{13} = (X_{13} \oplus X_2) \lll 16$. from Lemma 5, we have the probabilities of $X_{13}[k] = X'_{13}[k]$ for $k = 22, 23, 24, 25$ by the formula $\mathcal{P}_{(13,20+i)} = 1 - \frac{1}{2^{2+i}}$. According to this formula, we have

$$\mathcal{P}_{(13,i)} = \Pr(X_{13}[i] = X'_{13}[i]) = \begin{cases} 0.937 & \text{for } i = 22, \\ 0.968 & \text{for } i = 23, \\ 0.984 & \text{for } i = 24, \\ 0.992 & \text{for } i = 25. \end{cases} \tag{11}$$

After rotation by 16 bits, these bits move to the 6th, 7th, 8th and 9th bit. In the second step, the following operation takes place:

$$X_8 = X_8 + X_{13}.$$

We ignore the bits on the right side (less significant) of the 6th bit.
X_8 is not influenced by the difference so far. Using Theorems 2 and 3, we find the probability of $(X_8[i] = X'_8[i])$ for $i = 8, 9$ as follows:

We have, $p_i = \mathcal{P}_{(8,i)} = 1$ for $i = 6, 7, 8, 9$.

$$q_i = \mathcal{P}_{(13,i)} = \begin{cases} 0.937 & \text{for } i = 6, \\ 0.968 & \text{for } i = 7, \\ 0.984 & \text{for } i = 8, \\ 0.992 & \text{for } i = 9. \end{cases} \tag{12}$$

Since we ignore the carry difference coming from the previous bits, $\Pr(c_6 = c_6') = 1$. So, using Theorem 2 we calculate $\Pr(c_i \neq c_i')$ for $i = 7, 8, 9$. These are 0.032, 0.03149 and 0.0235, respectively. Using this and the formula in Theorem 3, we calculate the probabilities $\mathcal{P}_{(8,i)}$ for $i = 8, 9$ to be approximately 0.95 and 0.97, respectively. After this, X_{13} is XORed with X_2 and rotation is performed. This X_8 is later added with X_{13}, but does not face any significant change in the probabilities, since after the rotation in X_{13}, the new bits of X_{13} at the 8th and 9th positions are not influenced by the difference with any significant probability. On the other hand $\mathcal{P}_{(7,i)}$ for $i = 8, 9$ at this stage can be proved to be approximately 0.99 and 1 respectively, in exactly similar way we proved the probabilities $\mathcal{P}_{(3,15)}$ and $\mathcal{P}_{(3,16)}$ in Theorem 10. We skip this proof here. Finally the XOR of X_7 and X_8 takes place. In this XOR, the probabilities at the 8th and 9th bit we compute by the formula $pq + (1-p)(1-q)$ as in Theorem 1. For the 8th bit, $p = \mathcal{P}_{(7,8)} = 0.99$ and $q = \mathcal{P}_{(8,8)} = 0.95$. So, $pq + (1-p)(1-q) = 0.94$. Similarly, for the 9th bit, the value is 0.97. These bits, after rotation by 7 bits, reach our desired positions 15th and 16th, respectively. □

Theorem 13 *After completion of the second round* $\mathcal{P}_{(7,12)} = 0.75$.

Proof In the diagonal (X_2, X_7, X_8, X_{13}), the first operation is:

$$X_2 = X_2 + X_7; \quad X_{13} = ((X_{13} \oplus X_2) \lll 16).$$

After the first round, there is a difference at (13, 5). After this XOR and rotation by 16 bits, that difference transmits to position (13, 21). After that, in the following addition,

$$X_8 = X_8 + X_{13},$$

the difference transmits to (8, 21) and propagates to the next bits with probability given in Lemma 1. So, $\mathcal{P}_{(8,23)} = 0.75$, $\mathcal{P}_{(8,24)} = 0.875$ and $\mathcal{P}_{(8,25)} = 0.9375$.
These differences, after the step

$$X_7 = ((X_7 \oplus X_8) \lll 12),$$

transmit to X_7. So, we have $\mathcal{P}_{(7,3)} = 0.75$, $\mathcal{P}_{(7,4)} = 0.875$, $\mathcal{P}_{(7,5)} = 0.9375$.

On the other hand, from Theorem 5 in first round, we have the probabilities $\mathcal{P}_{(13,20)}$ and $\mathcal{P}_{(13,21)}$. In the second round, during the operation $X_{13} = (X_{13} \oplus X_2) \lll 16$, due to the rotation by 16 bits, those differences with same probabilities transmit to the 4th and 5th bits of X_{13}. Then, the operation $X_8 = X_8 + X_{13}$ takes place. We compute the probability $\mathcal{P}_{(8,5)}$. Before this operation, $X_8[4]$ and $X_8[5]$ were not influenced by the difference. So, we have: $p_i = \mathcal{P}_{(8,i)} = 1$ for $i = 4, 5$.
Now, $q_4 = \mathcal{P}_{(13,4)} = 0.75$ and $q_5 = \mathcal{P}_{(13,5)} = 0.875$. We ignore the bits on the right of the 4th bit. Now, using Theorems 2 and 3 we compute: $\mathcal{P}_{(8,5)} = 0.79$. This bit is added with the same bit of X_7 in the operation $X_7 = ((X_7 \oplus X_8) \lll 7)$ and after rotation reaches the 12th bit position. So the corresponding probability can be found using the formula of Theorem 3 as:

$$p = \mathcal{P}_{(8,5)} = 0.79,$$

$q = \mathcal{P}_{(13,5)} = 0.9375.$

So, $pq + (1-p)(1-q) \approx 0.75$, which is the probability $\mathcal{P}_{(7,12)}$. □

6.3 Third round

In the third round, we focus on the last column $(X_3, X_7, X_{11}, X_{15})$. The first half round is as follows:

$$X_3 = X_3 + X_7, \quad X_{15} = ((X_{15} \oplus X_3) \lll 16),$$
$$X_{11} = X_{11} + X_{15}, \quad X_7 = ((X_7 \oplus X_{11}) \lll 12).$$

Theorem 14 *After the half round in the 3rd round, we have the following results*:

(1) $\mathcal{P}_{(3,16)} = 0.92$,
$\mathcal{P}_{(3,8)} = 0.86$,
$\mathcal{P}_{(3,24)} = 0.68$.
(2) $\mathcal{P}_{(15,0)} \approx 0.65$
$\mathcal{P}_{(15,24)} \approx 0.75$
$\mathcal{P}_{(15,i)} = 0.62 \quad for\ i = 11$
$= 0.77 \quad for\ i = 12.$

Proof (1) We prove the result for $X_3[16]$ only. The other two follow similarly. In this proof, we consider the carry difference coming from the just previous bit (15th) and ignore the previous bits. In the second round, from Theorem 10 we have the probabilities $\mathcal{P}_{(3,16)} = 0.99$, $\mathcal{P}_{(3,15)} = 0.97$. Also, from the last theorem (Theorem 12), we have the probabilities $\mathcal{P}_{(7,16)}$ and $\mathcal{P}_{(7,15)}$ to be 0.97 and 0.94, respectively. From these information, using Theorems 2 and 3, we can easily calculate this probability as shown below:

$$p_{15} = \mathcal{P}_{(3,15)} = 0.97,$$
$$q_{15} = \mathcal{P}_{(7,15)} = 0.94.$$

Since we ignore the carry difference from the previous bits, $\Pr(c_{15} = c'_{15})$ we assume to be 1. So, from Theorem 2, $\Pr(c_{16} \neq c'_{16}) = \frac{1-p_{15}q_{15}}{2} = 0.044$. Now, $p_{16} = 0.99$ and $q_{16} = 0.97$. Therefore, from Theorem 3, we have $\mathcal{P}_{(3,16)} = 0.92$.
Similarly we can compute the probabilities $\mathcal{P}_{(3,24)}$ and $\mathcal{P}_{(3,8)}$.

(2) The bit $X_{15}[0]$ is the XOR of previous $X_{15}[16]$ and $X_3[16]$ (because of the rotation by 16 bits). So, using Theorem 11, we can find the probability from Theorem 3 as follows:

$$p = \mathcal{P}_{(15,16)} = 0.67 \quad \text{and} \quad q = \mathcal{P}_{(3,16)} = 0.93.$$

Therefore, $pq + (1-p)(1-q) \approx 0.65$.
Similarly, in the second case, $X_{15}[24]$ is the XOR of $X_{15}[8]$ and $X_3[8]$. So, in the same manner, we find the probability using Theorem 11. We do not show the calculation again. Similarly we can prove the results for $X_{15}[11]$ and $X_{15}[12]$ also.

□

Theorem 15 *After the half round in the third round, we have*

(1) $\mathcal{P}_{(11,0)} = 0.65$,
(2) $\mathcal{P}_{(11,12)} = 0.68$,
(3) $\mathcal{P}_{(7,24)} = 0.61$.

Proof (1) First we focus on $X_{11}[0]$. Since this is the 0th bit, it is the XOR of the 0th bit of X_{15} and X_{11}. We have: $\mathcal{P}_{(11,0)} = 1$ (since it is not yet influenced by the difference). $\mathcal{P}_{(15,0)} = 0.65$. (Theorem 14).
Therefore, again by the formula in Theorem 3, we can calculate the probability as follows: $p = 0.99$, $q = 0.65$ (as denoted in Theorem 3). Therefore, $pq + (1-p)(1-q) = 0.648 \approx 0.65$.

(2) For the second case, $\Pr(X_{15}[i] = X'_{15}[i])$ for $i = 11, 12$ is known from Theorem 14. Ignoring the carry coming from the previous bits, we calculate the probability based on the 11th and 12th bit only. From Theorem 2, we have $p_i = \mathcal{P}_{(11,i)} = 1$ for $i = 11, 12$ and
$q_i = \mathcal{P}_{(15,i)} = 0.62$ for $i = 11$,
$= 0.77$ for $i = 12$.
Using Theorem 2 we get $\Pr(c_{12} \neq c'_{12}) = 0.19$ and then using Theorem 3 we calculate the desired probability $\mathcal{P}_{(11,12)} = 0.68$.

(3) $X_7[24]$ is the XOR of $X_7[12]$ and $X_{11}[12]$ which is rotated by 12 bits. We know the probability of $(X_7[12] = X'_7[12])$ from Theorem 14. Therefore we can find the probability of this event using Theorem 3 as: $p = \mathcal{P}_{(11,12)} = 0.68$ and $q = \mathcal{P}_{(7,12)} = 0.75$. So, $pq + (1-p)(1-q) \approx 0.61 = \mathcal{P}_{(7,24)}$.

□

The next step is:

$$X_3 = X_3 + X_7, \quad X_{15} = ((X_{15} \oplus X_3) \lll 8),$$
$$X_{11} = X_{11} + X_{15}, \quad X_7 = ((X_{\oplus} X_{11}) \lll 7).$$

Theorem 16 *After the completion of third round,* $\mathcal{P}_{(11,0)} = \Pr(X_{11}[0] = X'_{11}[0]) \approx 0.51$.

Proof We have the probabilities $\mathcal{P}(3, 24)$ from Theorem 14 and $\mathcal{P}(7, 24)$ from Theorem 15. Using this, we compute the the updated $\mathcal{P}(3, 24)$ by the formula of Theorem 3. The probability comes approximately as 0.55.

Next, X_{15} is updated. We focus on the 0th bit of X_{15}. This bit is the XOR of 24th bit of X_{15} and X_3, which then gets rotated by 8 bits. We already have $\mathcal{P}_{(3,24)} \approx 0.55$. Theorem 14 tells us that $\mathcal{P}_{(15,24)} \approx 0.75$. Using Theorem 3 we can calculate these probability of the updated $(X_{15}[0] = X'_{15}[0])$, i.e. $\mathcal{P}_{(15,0)}$, which is approximately 0.53.

In the final step, we have:

$$X_{11} = X_{11} + X_{15} \quad X_7 = ((X_7 \oplus X_{11}) \lll 7).$$

We focus only on the addition between X_{11} and X_{15} because this gives us the final result. Since we focus on the 0th bit, it is the XOR of 0th bit of X_{11} and X_{15}.

$$X_{11}[0] = X_{11}[0] \oplus X_{15}[0].$$

The probability for $X_{15}[0]$ we have just calculated. For $X_{11}[0]$ we calculated in the last theorem (Theorem 15). Therefore, we calculate $\mathcal{P}_{(11,0)}$, the probability of $(X_{11}[0] = X'_{11}[0])$ again in the same manner using Theorem 3. Here, $p = 0.65$, $q = 0.53$. Therefore, the desired probability is: $pq + (1-p)(1-q) \approx 0.51$. □

Therefore, now we have the detailed theoretical structure of the forward bias in ChaCha. In Table 2, we provide a comparison between the probabilities achieved theoretically and by experiment side-by-side in the second and third column. Since for ChaCha we have so

Table 2 Comparison between the theoretical and the experimental results

Output bits	Theoretical result	Experimental result	Chosen IV experimental result [14]
$X_3^2[15]$	0.97	0.969	0.969
$X_{15}^2[16]$	0.67	0.667	0.909
$X_7^2[16]$	0.97	0.967	0.992
$X_7^2[12]$	0.75	0.745	0.882
$X_3^{2.5}[16]$	0.92	0.923	0.964
$X_{15}^{2.5}[0]$	0.65	0.652	0.880
$X_{11}^{2.5}[0]$	0.65	0.652	0.875
$X_7^{2.5}[24]$	0.61	0.609	0.743
$X_{11}^3[0]$	0.51	0.513	0.568

many theorems, we put only the major ones in the table. Similar to Salsa, for ChaCha also we perform experiment over 10,000 random key-IV pairs to produce this result. In the last column we provide the biases observed experimentally for the chosen IV case [14], where the number of differences after the first round is minimum. These results can be proved in the similar manner.

7 Probabilistic neutral bit

Before proceeding to the theoretical explanation of a probabilistically neutral bit, let us briefly explain what a probabilistic neutral bit is and how it is used in the key recovery attack.

7.1 Key recovery attack and probabilistically neutral bits

The quarterround function used in Salsa and ChaCha is a reversible function. So, each round of Salsa and ChaCha is reversible and from the final round matrix $X^{(R)}$, the matrix at any intermediate round $X^{(r)}$ can be obtained by applying the reverse function of quarterround on $X^{(R)}$ by $R - r$ rounds. For example, in Salsa, the reverse function is known as ReverseSalsa, which is as follows:

$$\begin{aligned} a &= a \oplus ((d + c) \lll 18), \\ d &= d \oplus ((c + b) \lll 13), \\ c &= c \oplus ((b + a) \lll 9), \\ b &= b \oplus ((a + d) \lll 7). \end{aligned}$$

Here we discuss briefly the idea of probabilistically neutral bits and their role in the key recovery attack on the cipher.

Probabilistically neutral bits suppose, by putting the input difference at position (i, j) of X we produce X' and the forward bias ϵ_d is observed at round r $(r < R)$ at a position (p, q), i.e, $\Pr(\Delta_p^{(r)}[q] = 0) = \frac{1}{2}(1 + \epsilon_d)$. Now, Suppose $Z = X + X^{(R)}$ and $Z' = X' + X'^{(R)}$ are the outputs at the end of R $(R > r)$ rounds of the cipher. Now, let k be a key bit position. We complement the value assigned at k for both X and X', producing $\overline{X}$ and $\overline{X'}$, respectively.

We run the reverse algorithm on $Z - \overline{X}$ and $Z' - \overline{X'}$ by $R - r$ rounds and achieve Y and Y', respectively. We investigate the position (p, q) of Y and Y'. Let us call the difference $Y_p[q] \oplus Y'_p[q]$ to be $\Gamma_p[q]$. Now, if the event $\Gamma_p[q] = \Delta_p^{(r)}[q]$ occurs with high probability, we call the key bit position k a probabilistically neutral bit or a PNB for this input bit (i, j) and output bit (p, q) combination.

Actual attack after achieving a set of probabilistically neutral bits, the attacker assigns random values to those key bit positions and try to guess the accurate values of the remaining key bits. Suppose, for one such guess, we get the matrices $\tilde{X}$ and $\tilde{X}'$. Now, on $Z - \tilde{X}$ and $Z' - \tilde{X}'$, the reverse algorithm is applied by $R-r$ rounds let us denote the outputs by $\tilde{Y}$ and $\tilde{Y}'$ respectively. The difference is investigated at position (p, q). Suppose $\Gamma_p\tilde{[q]} = Y_p\tilde{[q]} \oplus Y'_p\tilde{[q]}$. We call the bias of the event $\Gamma_p\tilde{[q]} = \Delta_p^{(r)}[q]$ backward bias and denote by ϵ_a. Now If the event $\Gamma_p\tilde{[q]} = 0$ shows a high bias ϵ, the guess of the non-PNB key bits are correct. After that we find the values of the PNBs by exhaustive search. The bias ϵ can be approximated by the product of the forward and backward bias, i.e., $\epsilon = \epsilon_d \cdot \epsilon_a$.

Since our aim is to provide theoretical explanation only, we don't provide here the detailed procedure of computing the time complexity of the attack. For a detailed study of the attack procedure and complexity calculation, one can visit the works in [1,14].

7.2 Theoretical explanation

In this section, we attempt to investigate the backward bias in the differential attack. The set of probabilistic neutral bits is responsible for the backward bias. We do not prove the backward bias for the entire set of probabilistic neutral bits. Rather, we pick a single bit from the list of probabilistically neutral bits of Salsa and explain why it works as a probabilistic neutral bit in the attack. A similar proof can be obtained for any other bit for both Salsa and ChaCha.

In the differential attack with input difference at (7, 31) and output difference at (1, 14) in Salsa, the probabilistically neutral bit that comes first in the list is the bit at position (12, 5). In fact in the list of 42 PNBs given in [7], all the bits from (12, 5) to (12, 18) are there.

We show for a random matrix M that if we change the value of the key bit at position (12, 5) to construct a matrix M' and then apply ReverseSalsa on both of them by 4 rounds, then in the output, the probability that at position (1, 14) there is a difference between M and M' is very low. Similar to the case of Salsa 8 rounds, in the first round we apply ReverseSalsa at first on the rows of M.

First round in the last row, $(a, b, c, d) = (M_{15}, M_{12}, M_{13}, M_{14})$. The first step of the Reverse-quarterround, where M_{15} is updated, does not involve M_{12}. So there is no propagation of the difference. Then, M_{14} is updated, where M_{12} comes in to play as follows:

$$M_{14} = M_{14} \oplus ((M_{12} + M_{13}) \lll 13).$$

So, the difference transmits to the 18th bit of M_{14} and propagates to the next bits with probability $\frac{1}{2^n}$ as in Lemma 1. In the next step in the update of M_{13}, in a similar manner the difference transmits to the 14th bit of M_{13} and propagates to next bits with probability $\frac{1}{2^n}$. In the final step, M_{12} is updated

$$M_{12} = M_{12} \oplus ((M_{14} + M_{15}) \lll 7).$$

The difference already in the 5th bit remains there. Another difference transmits from M_{14}. Due to the rotation by 7 bits it moves to the 25th bit and propagates to the bits on the left.

Second round
First column in this column, $(a, b, c, d) = (M_0, M_4, M_8, M_{12})$. The first step is:

$$M_0 = M_0 \oplus ((M_8 + M_{12}) \lll 18).$$

The differences at M_{12} gets rotated by 18 bits and transmits to M_0. So, the 23rd bit contains a difference, which propagates towards left with probability $\frac{1}{2^n}$. And the 11th bit contains a difference along with the bits on its left with decreasing probability.

$$M_{12} = M_{12} \oplus ((M_8 + M_4) \lll 13).$$

Since M_8 and M_4 do not possess any difference yet, so the differences at M_{12} remains at the same positions with the same probabilities.

$$M_8 = M_8 \oplus ((M_0 + M_4) \lll 13).$$

Among the involved cells, only M_0 contains differences. Those get rotated by 13 bits. So, the 4th bit contains a difference along with the bits on its left with gradually decreasing probability. Also, the 24th bit contains a difference along with gradually decreasing probability on its left.
Second column in this column, $(a, b, c, d) = (M_5, M_9, M_{13}, M_1)$. In the first step, M_5 is updated as:

$$M_5 = M_5 \oplus ((M_1 + M_{13}) \lll 18).$$

The difference at the 14th bit of M_{13} generates a difference at the 0-th bit of M_5. And the probability of propagation of this difference to the next bits is similar to Lemma 1.

Then M_1 is updated as:

$$M_1 = M_1((M_9 + M_{13}) \lll 13).$$

In a similar manner as the last one, the difference transmits to the 27th bit of M_1 and propagates in similar manner.

Next,

$$M_{13} = M_{13}((M_5 + M_9) \lll 9).$$

Only M_5 contains some differences. Due to the rotation by 9 bits, the difference at the 0th bit of M_5 produces a difference at the 9th bit of M_{13}. This propagates further up to some bits. So, the 0th and 1st bit of M_{13} is not yet influenced by the difference. Then, in the final step:

$$M_9 = M_9((M_5 + M_1) \lll 7).$$

Here, M_5 has differences starting from the 0th bit up to few bits on the left. After rotation by 7 bits, these produce differences at the 7th bit of M_9 and on the bits on the left side of it. Similarly M_1 has differences at the 27th bit and at the bits on its left, which after being rotated by 7 bits, move to the 2nd bit and on the left of the 7th bit of M_9 So, the 0th and 1st bit of M_9 is also not influenced by difference.
Third round in the third row, $(a, b, c, d) = (M_{10}, M_{11}, M_8, M_9)$.

$d = M_9$ is updated with the operation

$$M_9 = M_9 \oplus ((M_{11} + M_8) \lll 13).$$

M_{11} is not influenced by the difference since it is in last column. The 0th and 1st bit of M_9 is also not influenced yet. So, the only difference can transmit from M_8. M_8 contains difference at position 24th and 4th along with the bits on the left side of these bits with

gradually decreasing probability. Due to the rotation by 13 bits, these move to the 5th and 17th position. Therefore, the 0th and 1st position of M_9 would have negligible effect from the propagation of difference.
M_{13} in a similar method we can show that the 0th and 1st bit of M_{13} also is influenced by the difference with negligible probability.
M_1 also the fact that the 14th bit of M_1 is not influenced by the difference can be shown similarly.
Fourth round in this round, on the second column, $(a, b, c, d) = (M_5, M_9, M_{13}, M_1)$. In the second step, M_1 is updated as:

$$M_1 = M_1 \oplus ((M_9 + M_{13}) \lll 13).$$

Up to the 3rd round there was no difference at position (1, 14). Since we know that the 0th and 1st bit of M_9 and M_{13} do not have any difference, therefore after being rotated by 13 bits, they do not produce any difference at the 14th bit of M_1. Therefore this bit is not influenced by the difference. Exactly similar argument is valid for the key bits on the right side of (12, 5) up to (12, 18). Therefore, if a few key consecutive bits starting from (12, 5) to the right side are complemented to construct M', there would not be any influence on (1, 14) after applying ReverseSalsa by four rounds.

Now, in Salsa, we apply reverseround on $Z - \overline{X}$ and $Z' - \overline{X'}$ by 4 rounds. Let us assume that $M = Z - X$. So, $M' = Z - \overline{X}$ would have a difference from $Z - X$ at (12, 5) and this difference may propagate to a few bits on right. So, we can take $Z' - \overline{X'}$ to be our M'. Therefore, as we have shown, running 4 rounds of the ReverseSalsa on $Z - X$ and $Z - \overline{X}$ would have the same value at the position (1, 14). Now, running ReverseSalsa on $Z - X = X^{(8)}$ basically produces $X^{(4)}$ and by Y we denote the same output for $Z - \overline{X}$. Therefore,

$$Y_1[14] = X_1^{(4)}[14]. \tag{13}$$

Similarly, for $Z' - X'$ and $Z' - \overline{X'}$ we can use the same argument to show that

$$Y_1'[14] = X_1'^{(4)}[14]. \tag{14}$$

From Eqs. (13) and (14), we can say that

$$Y_1[14] \oplus Y_1'[14] = X_1^{(4)}[14] \oplus X_1'^{(4)}[14].$$

This implies, $\Gamma_1[14] = \Delta_1^{(4)}[14]$. So, $\Pr(\Gamma_1[14] = \Delta_1^{(4)}[14]) = 1$. This justifies the reason of the bit to be a PNB.

8 Conclusion

This work provides a deep insight of the differential attack idea against Salsa and ChaCha. For both the ciphers, our work finds out the exact reason for the observed forward biases, which have been the primary tool in all the differential distinguishing and key recovery attacks for one decade. It has always been an aim of the researchers to increase the observed bias and if possible, extend it to the next round. This theoretical explanation will help to get a clear picture of the propagation of the bias from the beginning, which may help to find some better distinguisher by some suitable method. In this work the theoretical justification of a probabilistically neutral bit has also been provided. We believe that this theory may also be helpful to improve the backward bias as well by application of some suitable techniques.

References

1. Aumasson J.P., Fischer S., Khazaei S., Meier W., Rechberger C.: New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba. FSE 2008, LNCS 5086, pp. 470–488 (2008).
2. Bernstein D.J.: Salsa20 specification. eSTREAM Project algorithm description (2005). http://www.ecrypt.eu.org/stream/salsa20pf.html.
3. Bursztein E.: Speeding up and strengthening HTTPS connections for Chrome on Android (2014). https://security.googleblog.com/2014/04/speeding-up-and-strengthening-https.html.
4. Choudhuri A.R., Maitra S.: Significantly improved multi-bit differentials for Reduced Round Salsa and ChaCha. IACR Trans. Symmetric Cryptol. **2016**(2), 261–287 (2016). http://eprint.iacr.org/2016/1034.
5. Crowley P.: Truncated differential cryptanalysis of five rounds of Salsa20. IACR 2005. http://eprint.iacr.org/2005/375.
6. Deepthi K., Singh K.: Cryptanalysis of Salsa and ChaCha: revisited. In: International Conference on Mobile Networks and Management (2018).
7. Dey S., Sarkar S.: Improved analysis for reduced round Salsa and ChaCha. Discret. Appl. Math. **227**(2017), 58–69 (2017).
8. Dey S., Sarkar S.: Settling the mystery of $Z_r = r$ in RC4. Cryptogr. Commun. **11**(4), 697–715 (2019).
9. Dey S., Sarkar S.: Proving the forward bias of Salsa. In: Workshop on Coding and Cryptography (2019). https://www.lebesgue.fr/sites/default/files/proceedings_WCC/WCC_2019_paper_48.pdf.
10. Dey S., Roy T., Sarkar S.: Revisiting the design principles of Salsa and ChaCha. Adv. Math. Commun. **13**(3), 689–704 (2019).
11. Ding L.: Improved related-cipher attack on Salsa20 Stream Cipher. IEEE Access **7**, 30197–30202 (2019).
12. Fischer S., Meier W., Berbain C., Biasse J.F.: Non-randomness in eSTREAM Candidates Salsa20 and TSC-4. In: Indocrypt 2006, LNCS 4329, pp. 2–16 (2006).
13. Isobe T., Ohigashi T., Watanabe Y., Morii M.: Full plaintext recovery attack on broadcast RC4. In: FSE 2013, LNCS 8424, pp. 179–202 (2013).
14. Maitra S.: Chosen IV cryptanalysis on Reduced Round ChaCha and Salsa. Discret. Appl. Math. **208**, 88–97 (2016).
15. Maitra S., Paul G., Meier W.: Salsa20 Cryptanalysis: New Moves and Revisiting Old Styles. WCC (2015). http://eprint.iacr.org/2015/217.
16. Mantin I., Shamir A.: A practical attack on broadcast RC4. In: FSE, LNCS 2355, pp. 152–164 (2001).
17. Neves S., Araujo F.: An observation on NORX, BLAKE2, and ChaCha. Inf. Process. Lett. (2019). https://doi.org/10.1016/j.ipl.2019.05.001.
18. Sengupta S., Maitra S., Paul G., Sarkar S.: (Non-)random sequences from (non-)random permutations—analysis of RC4 stream cipher. J. Cryptol. **27**(1), 67–108 (2014). http://eprint.iacr.org/2011/448.
19. Shi Z., Zhang B., Feng D., Wu W.: Improved key recovery attacks on Reduced-Round Salsa20 and ChaCha. In: ICISC, LNCS 7839, pp. 337–351 (2012).
20. Tsunoo Y., Saito T., Kubo H., Suzaki T., Nakashima H.: Differential Cryptanalysis of Salsa20/8. SASC (2007). http://www.ecrypt.eu.org/stream/papersdir/2007/010.pdf.