IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 10 January 2024, accepted 25 February 2024, date of publication 1 March 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3372857

RESEARCH ARTICLE

Analyzing the Probability of Key Recovery in the Differential Attacks Against ChaCha

NITIN KUMAR SHARMA AND SABYASACHI DEY
Department of Mathematics, Birla Institute of Technology and Science Pilani, Hyderabad Campus, Hyderabad 500078, India
Corresponding author: Nitin Kumar Sharma (sharmanitinkumar685@gmail.com)

ABSTRACT The stream cipher ChaCha has been subjected to differential linear cryptanalysis since 2008. Aumasson et al. (2008) laid the groundwork for this attack, employing the concept of probabilistically neutral bits for key recovery. Subsequently, various enhancements have been made to this attack over the last few decades. These improvements are essentially refinements to the probabilistically neutral bit-based attack approach. Despite the proposed modifications in these improvements, which increase attack complexity, the consequential changes in the associated probability of key recovery have not been thoroughly examined. A comprehensive analysis of the probability of key recovery is lacking in all attacks within this domain. No systematic process is available in the existing works for analyzing the probability of key recovery. This paper addresses this gap by proposing a method for estimating the probability of key recovery in these attacks. Employing this method, we calculate an estimated interval for the probability of key recovery for both the original idea presented by Aumasson et al. (2008) and the subsequent modifications to this idea. This analysis allows us to understand the variations in probability associated with these modifications.

INDEX TERMS Probability of key recovery, ChaCha, right pair-based attack, error probability.

I. INTRODUCTION

Symmetric encryption is a major branch of cryptography used for security, confidentiality, and data privacy. Symmetric ciphers mostly use mathematical functions in their encryption and decryption algorithms. Modular addition ($\boxplus$), bitwise rotation ($\lll$), and exclusive-OR ($\oplus$) (together known as ARX) operations-based algorithms are introduced in several symmetric-key cryptosystems. ChaCha is one such stream cipher that was developed by D. J. Bernstein in 2008, improving diffusion as compared to the Salsa cipher. ChaCha is highly used in software implementations as it is fast because of its machine-friendly operations. ChaCha is also used as a base function in the BLAKE hash function. The BLAKE hash function was a finalist in the NIST hash function competition. Google uses ChaCha with Poly1305 MAC as a cipher suite for the new TLS1.3. ChaCha20-Poly1305 is used in the QUIC protocol, which is used by HTTP/3. ChaCha is also used in many protocols like SSH, Noise, and S/MIME4.0. ChaCha uses Pseudo-Random Functions (PRFs), which are used as random number generators in FreeBSD, OpenBSD, and NetBSD operating systems. ChaCha is a preferred algorithm for mobile devices that use ARX-based CPUs. Other uses of ChaCha are mentioned in [2]. The Adiantum cipher [3], used by Google for disk encryption, uses ChaCha for length-preserving mode in low-end devices.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

A. PREVIOUS WORKS

Since the introduction of ChaCha [6] in 2008, many cryptanalytic techniques have been applied to the reduced-round versions of the cipher. In 2008, Aumasson et al. [1] presented the first cryptanalysis of 256-bit key version of ChaCha6 and ChaCha7 with time complexity 2^{139} and 2^{248}, respectively (ChaCha$R$ represents R-round ChaCha). The authors also introduced an attack on the 128-bit key version of ChaCha6 with time complexity 2^{107}.

In 2012, Shi et al. [7] improved the time complexity of attacks given by Aumasson et al. [1]. For the 256-bit key version of ChaCha6 and ChaCha7, the time complexity was reduced to 2^{136} and $2^{246.5}$, respectively. The time complexity of the 128-bit key version of ChaCha6 was also

TABLE 1. Estimated probability of key recovery for all three attack techniques.

Attack	Author's Claim		Our Estimated
	Applicable On	Probability to recover key	probability to recover key
Aumasson et al. [1]	100%	[0.50-1]	[0.63-0.71]
Beierle et al. [4]	70%	Not Mentioned	[0.81-0.85]
Dey et al. [5]	62%	Not Mentioned	[0.86-0.90]

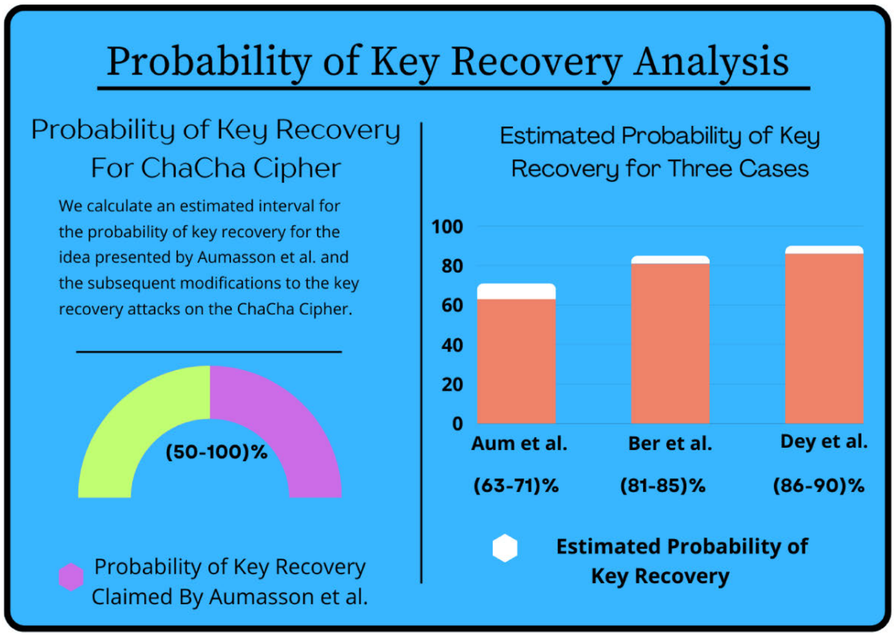

FIGURE 1. Probability of key recovery for ChaCha cipher.

reduced to 2^{105}. The probability of key recovery for the attack on ChaCha6 (128 and 256-bit key versions) is 45%. The attack mentioned on ChaCha7 has only a 43% probability of key recovery. In 2015, Maitra [8] provided an attack on the 256-bit key version of ChaCha7 with time complexity 2^{239}. The authors mentioned that complexity works for more than half of the keys, i.e., the probability of key recovery is approximately 50%. The author introduced the concept of Chosen $\mathcal{IV}$ to improve the attack techniques against the cipher. In 2016, Choudhuri and Maitra [9] provided an attack on the 256-bit key version of ChaCha$R$ ($4 \leq R \leq 7$). For ChaCha6 and ChaCha7 the time complexity is reduced 2^{116} and $2^{237.7}$, respectively. In 2017, Dey and Sarkar [10] improved the time complexity of the 256-bit key version of ChaCha7. The time complexity was $2^{235.2}$.

In 2019, Dey et al. [11] revisited the design principles of the ChaCha. In 2020, Coutinho and Neto [12] provided a new multi-bit differential and improved the time complexity for the 256-bit key version of ChaCha7, reducing the complexity to $2^{231.9}$. Beierle et al. [4] presented a paper in CRYPTO 2020, in which they further reduced the time complexity to $2^{230.86}$. They also reduce the time complexity of the 256-bit key version of ChaCha6. In addition, the authors provide a single-bit distinguisher of 3.5 rounds. In 2021, Coutinho and Neto [13] submitted a paper on modified attack procedures in Eurocrypt 2021. They improved the complexity by providing improved linear approximations to ChaCha and reducing the time complexity to $2^{228.51}$ for the 256-bit key version of ChaCha7. In 2021, Dey et al. [14] revisited the attack techniques mentioned in CRYPTO 2020 [4] and Eurocrypt 2021 [13], addressed some incorrect results, and provided a justification for the correct result.

In 2021, Dey and Sarkar [15] provided a mathematical proof for the observed probabilities of the distinguishers mentioned for the Salsa and ChaCha ciphers. In Eurocrypt 2022, Dey et al. [5] provided an improved attack that reduced the time complexity of the 256-bit key version of ChaCha7 to $2^{221.95}$. They also provided the time complexity for the 128-bit key version of ChaCha6. They also provided the first-ever attack on the 128-bit key version of ChaCha6.5 with a time complexity of $2^{123.04}$. In 2022, Coutinho et al. ([16], [17]) provided a 7-round distinguisher for ChaCha with time complexity 2^{214}. In 2022, Miyashita et al. [18] provided a differential attack on the 256-bit key version of ChaCha7 by improving the PNB technique. In 2023, Dey et al. [19] proposed the first-ever attacks on 7.25-round ChaCha256 with time complexity $2^{244.85}$. They also present an enhanced PNB-searching algorithm. In FSE 2023, Dey et al. [20] provided multiple $\mathcal{ID}-\mathcal{OD}$ differential and reduced the time complexity for ChaCha6 to $2^{99.48}$. In CRYPTO 2023, Wang et al. [21] improved the time complexity for the 256-bit key version of ChaCha6 and ChaCha7.

After introducing the concept of probabilistically neutral bits in differential attacks against Salsa and ChaCha, these modifications have led to clear improvements in terms of time complexity, but little attention has been given to the changes in the probabilities of key recoveries of the attacks. In [1], the authors claimed that the probability of key recovery was at least 50% but did not provide a better estimation. In our work, we show that the probabilities of key recoveries have indeed changed significantly.

B. OUR CONTRIBUTION

In the initial attack approach introduced by Aumasson et al. [1], the authors demonstrated its ability to recover at least 50% of the keys. Most of the subsequent works focused on enhancing the distinguishers or improving the probabilistically neutral bits, while the core attack idea remained unchanged. But, some modifications to the attack techniques were proposed by Beierle et al. [4] and Dey et al. [5]. However, neither of these works conducted a thorough analysis to determine whether the probability of key recovery differs from the original approach. In general, none of the existing attacks has sincerely addressed the probability of key recovery, which creates a gap in the proper evaluation of any modifications proposed in the attack technique.

Firstly, this paper introduces a systematic procedure to compute an estimated interval for the probability of successful key recovery in the context of differential attack techniques applied to ChaCha. This procedure can be used not only for existing attacks but also for potential future attacks in this line. With the help of this, the introduction of any new idea can be thoroughly assessed, taking into account not only the improvement in complexity but also the change in the probability of key recovery.

Secondly, we apply our method to obtain the probability of key recovery for all three attack approaches mentioned above ([1], [4], [5]), utilizing the same distinguisher. Upon analyzing and comparing the probabilities, we have several findings related to the probability of key recovery. We mention below our findings for each of these works.

1) According to the claim of Aumasson et al. [1], their attack idea, which is applicable to all keys, can recover the key with a probability of at least 0.5. They did not mention any tighter estimation of the probability. In our computation, we found a more precise estimate for the probability of key recovery, which is [0.630, 0.713].
2) Next, in the attack by Beierle et al. [4], the authors claimed that their technique is applicable only for 70% of the keys, which they call "weak keys." They did not analyze the probability of successful key recovery among those weak keys. Neither do they comment on the performance of this approach against the remaining 30% "strong keys". Our analysis shows that this attack approach can be used for all keys (instead of only "weak keys"), with the probability of key recovery in the interval [0.816, 0.849]. Therefore, the efficiency of the attack is not only significantly higher than the author's claim but also higher than the probability of the previous attack approach by [1].
3) Lastly, we analyzed the work of Dey et al. [5]. In their work, they provided an attack technique that is applicable for only 62% of the keys, which they call "exploitable keys". Again, the authors did not analyze the probability of successful key recovery of the approach on its application on the exploitable keys, nor did they comment anything on the scenario when the key is not "exploitable". We compute and show that the attack can be used for all keys with a probability of key recovery lying in [0.866, 0.900], which is higher than both of the previous two approaches.

We have mentioned all the observations in detail in Table 1. We provide the results claimed by authors in the previous three works. The table contains information about the percentage of keys for which the mentioned attack technique is applicable, the probability of key recovery claimed by the authors, and our estimated value of the same for all three attack procedures.

1) PAPER ORGANIZATION

The paper is organized as follows:

- Section II comprises the preliminary details of ChaCha and its cryptanalysis techniques. We explain the detailed structure of the ChaCha in Subsection II-A. In Subsection II-B, we discuss the differential cryptanalysis and attack procedure. The list of notations is given in Table 2.
- Section III revisits the complexity calculation technique of the existing attack approaches and derives some mathematical relations associated with the data complexity used in the attack. These relations are to be used afterwards in the computation of the probability of key recovery.
- In Section IV, we discuss the distribution of biases of the distinguisher for different keys. This distribution plays a part in the computation of the probability of key recovery.
- In Section V, we introduce a technique to compute the probabilities of key recoveries of differential-linear attacks on ChaCha. We compute estimated intervals of the probabilities of key recoveries of the existing attack approaches. Using this, in Subsection V-A, we find the probability of key recovery for the attack procedure by Aumassaon et al. [1], which appears to be in the interval [0.63, 0.71]. Subsection V-B and Subsection V-C compute the probability of key recovery computation for

the attack procedures mentioned in [4] and [5], respectively, whose estimated intervals are [0.81, 0.85] and [0.86, 0.90], respectively.

- Finally, we conclude our work in Section VI.

II. PRELIMINARIES

In this section, we explain the basic structure of the ChaCha cipher, the differential cryptanalysis, and the attack procedure. In this section, we explain the fundamentals of forward bias ϵ_d and backward bias ϵ_a.

A. ChaCha CIPHER

ChaCha works on sixteen 32-bit words represented as a 4×4 matrix. The cipher design has two key versions, 128-bit and 256-bit. The 256-bit key version of cipher takes 8 key words $(k_0, k_1, \ldots, k_7)$, 4 constants words (c_0, c_1, c_2, c_3), 1 $\mathcal{IV}$ word t_0 and 3 counter words (v_0, v_1, v_2) as input and generates a 512-bit output. The first row of matrix consists of 4 words or constants $c_0, c_1, c_2,$ and c_3 derived from word "expand 32-byte k". The second and third row of the matrix consists of keywords $(k_0, k_1, \ldots, k_7)$. Fourth row consists of $\mathcal{IV}$ word t_0 and counter words (v_0, v_1, v_2). The four constants for 256-bit key structure are

$$c_0 = \texttt{0x61707865}, \ c_1 = \texttt{0x3320646e},$$
$$c_2 = \texttt{0x79622d32}, \ c_3 = \texttt{0x6b206574}.$$

For the 128-bit key structure, the second and third rows of the matrix are the same. There is a slight change in the constants for the 128-bit key structure. The four constants for the 128-bit key structure are

$$c_0 = \texttt{0x61707865}, \ c_1 = \texttt{0x3120646e},$$
$$c_2 = \texttt{0x79622d36}, \ c_3 = \texttt{0x6b206574}.$$

The matrix looks as follows:

$$X = \begin{pmatrix} X_0 & X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} \\ X_{12} & X_{13} & X_{14} & X_{15} \end{pmatrix} = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & v_0 & v_1 & v_2 \end{pmatrix}.$$

In ChaCha, the round function is a nonlinear operation consisting of three operations viz: XOR operation between the bits ($\oplus$), addition modulo 2^{32} ($\boxplus$), and left cyclic rotation operation for each round ($\lll$). In this round function (a, b, c, d) is the initial vector that transforms into vector (a'', b'', c'', d'') as shown below:

$$\begin{aligned} a' &= a \boxplus b; \quad & d' &= ((d \oplus a') \lll 16); \\ c' &= c \boxplus d'; \quad & b' &= ((b \oplus c') \lll 12); \\ a'' &= a' \boxplus b'; \quad & d'' &= ((d' \oplus a'') \lll 8); \\ c'' &= c' \boxplus d''; \quad & b'' &= ((b' \oplus c'') \lll 7); \end{aligned} \tag{1}$$

For an initial state matrix X, after applying for n round functions, we obtain a state matrix $X^{(n)}$. For odd rounds, the round function acts along the columns of the matrix. This is called the columnround function. The four columns of state matrix X, *viz.* $(X_0, X_4, X_8, X_{12}), (X_1, X_5, X_9, X_{13}), (X_2, X_6, X_{10}, X_{14})$ and $(X_3, X_7, X_{11}, X_{15})$. For even rounds, the round function acts along the diagonals of the matrix. This is called the diagonalround function. The four diagonals are $(X_0, X_5, X_{10}, X_{15}), (X_1, X_6, X_{11}, X_{12}), (X_2, X_7, X_8, X_{13})$ and (X_3, X_4, X_9, X_{14})

The keystream block Z is obtained by addition of matrices $X^{(0)}$ and $X^{(n)}$ as shown below:

$$Z = X^{(0)} \boxplus X^{(n)},$$

where $X^{(0)}$ is denoted as the initial state and $X^{(n)}$ is the state after n-rounds of X.

Every ChaCha round function is reversible. In reverse round function the vector (a'', b'', c'', d'') acts as initial vector and changes into vector (a, b, c, d) as follows:

$$\begin{aligned} b'' &= ((b' \oplus c'') \lll 7); \quad & c' &= c'' \boxminus d''; \\ d'' &= ((d' \oplus a'') \lll 8); \quad & a' &= a'' \boxminus b'; \\ b' &= ((b \oplus c') \lll 12); \quad & c &= c' \boxminus d'; \\ d' &= ((d \oplus a') \lll 16); \quad & a &= a' \boxminus b; \end{aligned} \tag{2}$$

B. DIFFERENTIAL CRYPTANALYSIS AND ATTACK PROCEDURE

We explain the differential cryptanalysis introduced by Aumasson et al. [1] in 2008 for ChaCha. We consider X to be the initial state matrix. We introduce input differential Δ_{in} to generate another state matrix X', where $X' = X \oplus \Delta_{in}$. The differential Δ_{in} indicates value 1 at j-th bit of the i-th word and 0 at the remaining bits. Now, we look for the output differential after r rounds. Let $\Delta X_i^{(r)}[j] = X_i^{(r)}[j] \oplus X_i'^{(r)}[j]$ denotes the difference obtained between two states X and X' at j-th bit of the i-th word after r-rounds. The bias obtained after r-rounds is known as forward bias and is denoted by ϵ_d.

$$\Pr\left[\Delta X_i^{(r)}[j] = 0 | \Delta X = \Delta_{in}\right] = \frac{1}{2}(1 + \epsilon_d). \tag{3}$$

Here, the bias ϵ_d holds for all the outputs. In the attacks, we consider an estimated median value of the biases over all the keys. In Section IV, we have shown the distribution of forward bias ϵ_d.

After extending the state matrices X and X' up to r-rounds, we also find the final state matrices $X^{(n)}$ and $X'^{(n)}$ after n rounds. We also generate the keystream blocks corresponding to both the state matrices X and X' given as $Z = X^{(0)} \boxplus X^{(n)}$ and $Z' = X' \boxplus X'^{(n)}$. Differential cryptanalysis is not only used to analyse the security of ARX ciphers but also used to do the performance analysis of audio [22] and image encryption [23], [24] algorithms.

Now, we discuss the concept of probabilistically neutral bits (PNBs) to find the value of the key bits. Aumasson et al. [1] introduced this concept by dividing the key bits into two sets, PNBs and non-PNBs, on the basis of their influence on the output difference bit. The m bits that have a high influence on the output difference bit are considered as non-PNBs, and the remaining $(256 - m)$ as PNBs.

Next, we discuss how to find the values of non-PNB bits. We fix some guessed values at the m positions of the state matrices X and X' to obtain two new states $\bar{X}$ and $\bar{X}'$. We use N pairs of keystream blocks for each guessed key. Using the value of keystream blocks Z and Z', we obtain the two new matrices $\bar{M} = Z - \bar{X}$ and $\bar{M}' = Z' - \bar{X}'$. Now we run the reverse round function for $n-r$ rounds and obtain the difference $\Delta\bar{M}_i^{(n-r)}[j] = \bar{M}_i^{(n-r)}[j] \oplus \bar{M}_i'^{(n-r)}[j]$.

Hence, if $\left(\Delta\bar{M}_i^{(n-r)}[j] = \Delta X_i^{(r)}[j]\right)$ with a high probability, we consider the bias to be backward bias and is denoted by ϵ_a. Therefore, the final bias mentioned by Aumasson et al. [1], for $\left(\Delta\bar{M}_i^{(n-r)}[j] = 1\right)$ is given by $\frac{1}{2}(1+\epsilon)$, where $\epsilon = \epsilon_d \cdot \epsilon_a$ under reasonable independency assumptions.

1) ATTACK USING RIGHT-PAIRS

Beierle et al. [4] introduced the idea of right pair-based attack. The authors observed that for approximately 70% of the keys, there exists at least one $\mathcal{IV}$, which produces a minimum difference of 10 in the first round. They called such keys "weak keys" and the pairs of keys and $\mathcal{IV}$s right pair. They observed that, for a weak key, on average, out of 2^5 random guesses, one $\mathcal{IV}$ appears to form a right pair. Having a right pair helps to improve the bias of the distinguisher. One can visit Section V-B of [4] for details.

2) MEMORY-BASED ATTACK USING RIGHT-PAIRS

Dey et al. [5] further introduced memory into this attack. They decomposed the key space into no-memory key space and Memory key space. For each member of memory key space, the attacker can find a $\mathcal{IV}$, which forms a right pair for each of the keys belonging to that coset. This pair can be stored in memory beforehand. Therefore, during the attack, the 2^5 random guesses can be avoided. For details, one can visit Section 7.1 of [5].

To distinguish between the initial attack approach (by Aumasson et al. [1]), where $\mathcal{IV}$s were chosen randomly and the right pair-based attack approach, we call the first one as "random-$\mathcal{IV}$ based attack."

III. MATHEMATICAL FORMULATION OF THE TIME AND DATA COMPLEXITY

Let $\mathcal{X}$ be the normal distribution. The parameters for the distribution $\mathcal{X}$ are:

$$E(\mathcal{X}) = \frac{N}{2}(1+\epsilon) \text{ and } \sigma_{\mathcal{X}} = \sqrt{V(\mathcal{X})} = \sqrt{\frac{N}{4}(1+\epsilon)(1-\epsilon)}.$$

The mean and standard deviation for the normal distribution $\mathcal{X}_1$ are denoted by μ_1 and σ_1 respectively and are given by:

$$\mu_1 = \frac{N}{2}(1+\epsilon) \text{ and } \sigma_1 = \sqrt{\sigma_{\mathcal{X}}} = \sqrt{\frac{N}{4}(1+\epsilon)(1-\epsilon)}.$$

$\mathcal{X}_0$ is the distribution generated from random output. The parameters for $\mathcal{X}_0$ distribution are:

$$\mu_0 = \frac{N}{2} \quad \text{and} \quad \sigma_0 = \frac{\sqrt{N}}{2}.$$

TABLE 2. Table of notations.

Notation	Meaning
X	The state matrix of the cipher consisting of 16 words
$X^{(0)}$	Initial state matrix
$X^{(r)}$	State matrix after r rounds
$\Delta X_i^{(r)}[j]$	XOR difference after the r-th round of the j-th bit of the i-th word of two states X and X'
Z	Key stream block obtained by adding X and $X^{(n)}$
ChaCha$R$	R rounds of ChaCha
$\mathcal{ID}$	Input Difference position
$\mathcal{OD}$	Output Difference position
ϵ_d	Bias obtained in forward direction
ϵ_a	Bias obtained in backward direction
ϵ	Approximate value of $\epsilon_d \times \epsilon_a$
Θ	Threshold Value
P_j	Percentage of key bits lying in an interval $[\epsilon_{d_j}, \epsilon_{d_{j+1}}]$
Φ	Cumulative Distribution Function of standard normal distribution

Let $H_0 : \mathcal{X} = \mathcal{X}_0$ be the hypothesis if the distribution is generated from the random output and $H_1 : \mathcal{X} = \mathcal{X}_1$ be the hypothesis if the distribution is the normal distribution. Let Θ be the threshold and should follow the error probability conditions.

For the non-detection error, we have the

$\Pr(Y \leq \Theta | H_1) < 1.3 \times 10^{-3}$ and for the false alarm error the $\Pr(Y \geq \Theta | H_0) < 2^{-\alpha}$ for some suitable α.

A. FALSE ALARM ERROR

For the distribution $\mathcal{X}_0$ the probability $\Pr(\mathcal{X}_0 \geq \Theta)$ is less than $2^{-\alpha}$ i.e., $\Pr(\mathcal{X}_0 \geq \Theta) \leq 2^{-\alpha}$. Similarly $\Pr\left(\frac{\mathcal{X}_0 - \mu_0}{\sigma_0} \geq \frac{\Theta - \mu_0}{\sigma_0}\right) \leq 2^{-\alpha}$, where $\frac{\mathcal{X}_0 - \mu_0}{\sigma_0}$ denotes the standard normal and is given as:

$$Z = \frac{\mathcal{X}_0 - \mu_0}{\sigma_0} = \frac{\mathcal{X}_0 - \frac{N}{2}}{\frac{\sqrt{N}}{2}}.$$

This implies

$$\Pr\left(Z \geq \frac{\Theta - \frac{N}{2}}{\frac{\sqrt{N}}{2}}\right) \leq 2^{-\alpha} \tag{4}$$

We know, $\Pr(Z \geq x) \leq e^{-x^2/2}$. Therefore we have $e^{-x^2/2} = 2^{-\alpha}$. This implies

$$\left(\frac{\Theta - \frac{N}{2}}{\frac{\sqrt{N}}{2}}\right)^2 = 2 \times \alpha \log 2$$
$$\implies \frac{\Theta - \frac{N}{2}}{\frac{\sqrt{N}}{2}} = \sqrt{\alpha \log 4}. \tag{5}$$

B. NON-DETECTION ERROR

The standard normal form of $\mathcal{X}_1$ is given as:

$$Z = \frac{\mathcal{X}_1 - \mu_1}{\sigma_1} = \frac{\mathcal{X}_1 - \frac{N}{2}(1+\epsilon)}{\frac{\sqrt{N}}{2}\left(\sqrt{1-\epsilon^2}\right)}.$$

As we mentioned, the non-detection error probability is less than 1.3×10^{-3}. So, $\Pr(\mathcal{X}_1 \leq \Theta) \leq 1.3 \times 10^{-3}$. Changing the above inequality in standard form we have $\Pr\left(Z \leq \frac{\Theta - \mu_1}{\sigma_1}\right) \leq 1.3 \times 10^{-3}$.

Since $\Pr(Z \leq -3) = 1.3 \times 10^{-3}$. This implies

$$\frac{\Theta - \frac{N}{2}(1+\epsilon)}{\frac{\sqrt{N}}{2}\left(\sqrt{1-\epsilon^2}\right)} = -3.$$
$$\Theta = \frac{N}{2}(1+\epsilon) - 3 \times \frac{\sqrt{N}}{2}\left(\sqrt{1-\epsilon^2}\right). \tag{6}$$

To compute the data complexity value, we compare the Equation 5 and Equation 6 to compute the data complexity value. Rewriting Equation 6 we get

$$\frac{\Theta - \frac{N}{2}}{\frac{\sqrt{N}}{2}} = \sqrt{N}\epsilon - 3\sqrt{1-\epsilon^2} \tag{7}$$

Comparing the L.H.S of Equation 5 and Equation 7 we get

$$\sqrt{N}\epsilon - 3\sqrt{1-\epsilon^2} = \sqrt{\alpha \log 4} N = \left(\frac{\sqrt{\alpha \log 4} + 3\sqrt{1-\epsilon^2}}{\epsilon}\right)^2. \tag{8}$$

The time complexity of the attack is given as

$$C = 2^m N + 2^{(256-\alpha)}. \tag{9}$$

IV. PARTITIONING THE RANGE OF THE BIASES

In the existing attacks, as mentioned in Subsection II-B, the median value of the bias is considered for the computation of the attack complexity. Therefore, the formula works for approximately 50% of all the keys. In this work, we aim to compute the probability when, for a given key, the bias is lower than the median value. The overall probability of key recovery depends on the distribution of biases for different keys.

Throughout our analysis, we use the 3.5-round distinguisher $\left(\Delta_{12}^{(0)}[6] - \Delta_1^{(3.5)}[0]\right)$ given by Beierle et al. [4], where $\Delta_{12}^{(0)}[6]$ denotes the $\mathcal{ID}$ position and $\Delta_1^{(3.5)}[0]$ is the $\mathcal{OD}$ position. Here, we divide the bias values into a fixed number of intervals, and for each interval, find out the percentage of key bits whose bias lies in that particular interval.

We consider only the median value of the forward bias ϵ_d for an $\mathcal{ID} - \mathcal{OD}$ position, but the bias is distributed over all the keys. Let $[0, \mathcal{E}]$ be the range of the forward bias value. We divide this interval into k sub-intervals $[\epsilon_{d_j}, \epsilon_{d_{j+1}}]$, $0 \leq j \leq k-1$. Here, $\epsilon_{d_0} = 0$ and $\epsilon_{d_k} = \mathcal{E}$ denotes the minimum and maximum bias value respectively. The value ϵ_d denotes the median value of the k values $\epsilon_{d_0}, \epsilon_{d_1}, \ldots, \epsilon_{d_{k-1}}$ and lies in an interval $[\epsilon_{d_j}, \epsilon_{d_{j+1}}]$. Let P_j denote the percentage of key bits lying in the interval $[\epsilon_{d_j}, \epsilon_{d_{j+1}}]$. The source code for computing the bias distribution for the Right pair-based attack and the Random $\mathcal{IV}$ based attack cases is available in Code Ocean [25].

A. ANALYSIS OF BIAS DISTRIBUTION FOR RIGHT PAIR-BASED ATTACK

In this, we have presented the forward bias distribution for the Right pair-based attack case. The median bias value ϵ_d in the Right pair-based attack case for the $\mathcal{ID} - \mathcal{OD}$ pair is 0.00317. In Figure 2, the histogram represents the percentage of key bits in intervals. The percentage value for the respective interval is mentioned in Table 3. The x-axis represents the values of biases (scaled by a factor of 10^{-4}). Y-axis represents the percentage of keys lying in an interval. Note that the intervals are not necessarily equal. We have chosen the intervals in such a way that we can provide a tight estimation (small interval).

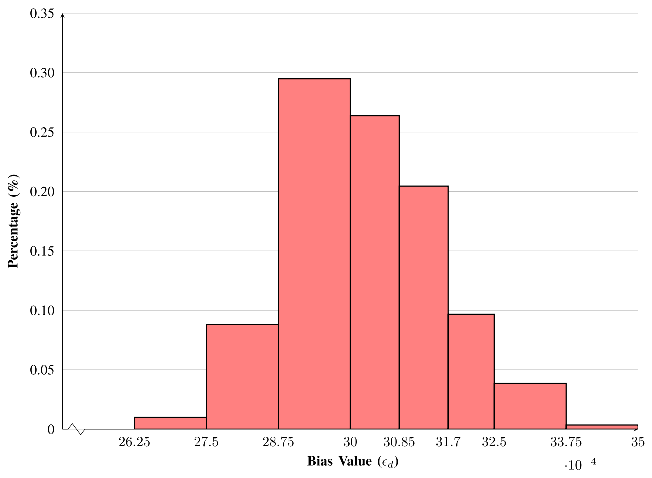

FIGURE 2. Bias distribution for right pair-based attack.

B. ANALYSIS OF BIAS DISTRIBUTION FOR RANDOM $\mathcal{IV}$ BASED ATTACK

In this, we have presented the Random $\mathcal{IV}$ based attack case for the same pair of $\mathcal{ID} - \mathcal{OD}$. The experimentally observed median bias value ϵ_d in the Random $\mathcal{IV}$ based attack case for the $\mathcal{ID} - \mathcal{OD}$ pair is 0.00050. In Figure 3, the histogram represents the percentage of key bits in intervals. The percentage value for the respective interval is mentioned in Table 4.

TABLE 3. Percentage of bias distribution for right pair-based attack.

Bias Range ($\times 10^{-4}$)	26.25-27.5	27.50-28.75	28.75-30.00	30.00-30.85	30.85-31.70	31.70-32.50	32.50-33.75	33.75-35.00
Percentage (P_j)	0.010	0.0882	0.2948	0.2637	0.2045	0.0967	0.0386	0.0035

TABLE 4. Percentage of bias distribution for random $\mathcal{IV}$ based attack.

Bias Range ($\times 10^{-4}$)	0-1	1-1.5	1.5-2	2-2.5	2.5-3	3-3.5	3.5-4	4-5	5-6.25	6.25-7.5	7.5-8.75	8.75-10	10-11.25	11.25-12.5
Percentage (P_j)	0.055	0.040	0.045	0.040	0.080	0.080	0.085	0.105	0.155	0.090	0.090	0.070	0.040	0.040

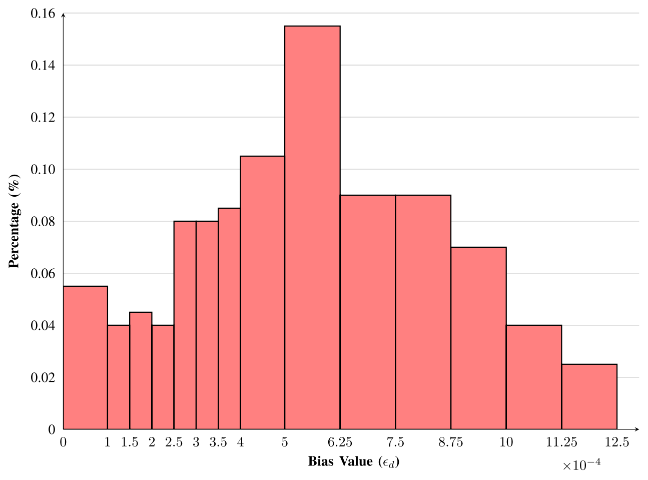

FIGURE 3. Bias distribution for random $\mathcal{IV}$ based attack.

V. CALCULATION OF PROBABILITIES OF KEY RECOVERIES

In 2006, Fischer et al. [26] introduced the concept of probability of key recovery in the attack on the Salsa cipher. Aumasson et al. [1] tried to improve the probability of key recovery for the Salsa cipher and obtain the same for ChaCha. In 2012, Shi et al. [7] provided the probability of key recovery ideas but did not provide the proper formulation. Similarly, the probability of key recovery computation is ignored in most previous attacks on ChaCha. In this section, we provided the formula for obtaining the probability of key recovery in the differential-linear attack on ChaCha. This computation works for all the ciphers with similar design algorithms. Now, we discuss how to compute the probability of key recovery.

The distribution $\mathcal{X}_1$ is approximated as normal distribution with mean $\frac{N}{2}(1+\epsilon)$ and standard deviation $\sqrt{\frac{N}{4}(1-\epsilon^2)}$. The form $\frac{\mathcal{X}_1-\frac{N}{2}(1+\epsilon)}{\sqrt{\frac{N}{4}(1-\epsilon^2)}}$ represents the standard normal distribution, where $\epsilon = \epsilon_d \times \epsilon_a$.

As mentioned in Section III, $\Pr(\mathcal{X}_1 \leq \Theta)$ denotes the non-detection error probability. So, $\Pr(\mathcal{X}_1 \geq \Theta)$ is the probability that the correct guess of significant key bits is detected. Since $\frac{\mathcal{X}_1-\frac{N}{2}(1+\epsilon)}{\sqrt{\frac{N}{4}(1-\epsilon^2)}}$ follows standard normal distribution, $\Pr(\mathcal{X}_1 \geq \Theta)$ can be given by

$$1 - \Phi\left(\frac{\Theta - \frac{N}{2}(1+\epsilon)}{\sqrt{\frac{N}{4}(1-\epsilon^2)}}\right). \tag{10}$$

We know, Φ denotes the Cumulative Distribution Function of standard normal distribution. For simplification, let us consider $\mathcal{F}(\epsilon) = \frac{\Theta-\frac{N}{2}(1+\epsilon)}{\sqrt{\frac{N}{4}(1-\epsilon^2)}}$.

Now, using Equation 6, we observe that, $\Theta - \frac{N}{2}(1+x) < 0$, as $\left(\sqrt{1-x^2}\right)$ is positive for $0 < x < 1$.

For $x < y$, we have.

$$\frac{1}{\sqrt{\frac{N}{4}(1-x^2)}} < \frac{1}{\sqrt{\frac{N}{4}(1-y^2)}}$$
$$\implies \frac{\Theta - \frac{N}{2}(1+x)}{\sqrt{\frac{N}{4}(1-x^2)}} > \frac{\Theta - \frac{N}{2}(1+x)}{\sqrt{\frac{N}{4}(1-y^2)}}.$$

Also, $\Theta - \frac{N}{2}(1+x) < \Theta - \frac{N}{2}(1+y)$. This implies

$$\frac{\Theta - \frac{N}{2}(1+x)}{\sqrt{\frac{N}{4}(1-x^2)}} > \frac{\Theta - \frac{N}{2}(1+y)}{\sqrt{\frac{N}{4}(1-y^2)}}$$
$$i.e., \quad \mathcal{F}(x) > \mathcal{F}(y).$$

Hence, $\mathcal{F}$ is a decreasing function.

In Section IV, we mentioned that the forward bias value of all the keys lies in the intervals $[\epsilon_{d_j}, \epsilon_{d_{j+1}}]$, $0 \leq j \leq k-1$. Therefore, the approximate value of bias $\epsilon_j = \epsilon_{d_j} \times \epsilon_a$ for $0 < j < k-1$.

Now, we have to find the probability for the attack procedure to be successful if the bias lies in the interval $[\epsilon_j, \epsilon_{j+1}]$. Hence, for $\epsilon_j < \epsilon < \epsilon_{j+1}$, we have $\mathcal{F}(\epsilon_j) > \mathcal{F}(\epsilon) > \mathcal{F}(\epsilon_{j+1})$.

Φ, being an increasing function, Φ reserves the sign of the inequality. Therefore,

$$\Phi\left(\mathcal{F}(\epsilon_j)\right) > \Phi\left(\mathcal{F}(\epsilon)\right) > \Phi\left(\mathcal{F}(\epsilon_{j+1})\right).$$

Now the probability that the correct guess of the significant key bits is not detected given that the bias value lies in the interval $[\epsilon_j, \epsilon_{j+1}]$ is denoted as error probability and is given by

$$\Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j > \Pr(Y \leq \Theta | j) > \Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j$$

where, Y denotes the case when $\epsilon_j < \epsilon < \epsilon_{j+1}$ for a particular value of j.

Hence, the total probability over the interval $[0, \mathcal{E}]$ is as

$$\sum_{j=0}^{k-1} \Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j > \Pr(Y \leq \Theta|j) > \sum_{j=0}^{k-1} \Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j.$$

From the above inequality, we obtain the sum of error probability values of all the k sub-intervals of bias interval $[0, \mathcal{E}]$. The sum of error probability values can be represented in the interval form as

$$\left[\sum_{j=0}^{k-1} \Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j, \sum_{j=0}^{k-1} \Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j\right].$$

Now, to compute the probability of key recovery, we subtract the error probability values from 1. Hence, the probability of key recovery lies in the interval.

$$\left[(1 - \sum_{j=0}^{k-1} \Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j), (1 - \sum_{j=0}^{k-1} \Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j)\right].$$

A. PROBABILITY OF KEY RECOVERY IN THE ATTACK PROCEDURE BY AUMASSON ET AL. [1]

Aumasson et al. [1] obtained the forward bias for Random $\mathcal{IV}$ based attacks as the idea of right pair-based attack was introduced by Beierle et al. [4] much later. Aumasson et al. obtained the data and time complexity value for ChaCha7 using the $\mathcal{ID} - \mathcal{OD}$ pair $\left(\Delta_{14}^{(0)}[7] - \Delta_{10}^{(3)}[0]\right)$. But in 2020 Beierle et al. [4], provided a 3.5 round differential $\left(\Delta_{12}^{(0)}[6] - \Delta_{1}^{(3.5)}[0]\right)$. We use this 3.5 round $\mathcal{ID} - \mathcal{OD}$ pair throughout for uniformity.

Now, to compute the probability of key recovery, we find the value of $\mathcal{F}(\epsilon_j)$ for each $\epsilon_j = \epsilon_{d_j} \times \epsilon_a$ for $0 < j < 13$. Next, we compute the value of Φ function for each $\mathcal{F}(\epsilon_j)$. To obtain the value of the error probability for the bias values ϵ_0 and ϵ_1, we multiply the probability value P_0 with $\Phi\left(\mathcal{F}(\epsilon_0)\right)$ and $\Phi\left(\mathcal{F}(\epsilon_1)\right)$ respectively. Similarly, we compute the value of error probability for ϵ_j and ϵ_{j+1}, $0 < j < 13$. After adding the error probabilities of all ϵ_j's, we compute the probability of key recovery.

Using the attack technique given by Aumasson et al. [1], the bias ϵ_d observed for 3.5 rounds is equal to 0.00050. This bias is also known as forward bias. Also, bias observed in a backward direction from 7 rounds to 3.5 rounds is known as backward bias and is given by $\epsilon_a = 0.00057$. In this procedure, we have considered 79 key bits as PNBs. Substituting the value of $\epsilon = \epsilon_d \cdot \epsilon_a$ in Equation 8 we evaluated the data complexity $N = 2^{49.96} = 1095112114681510$ for $\alpha = 30$. Using the value of N, we get the threshold value $\Theta = 547556163755514.06$ from Equation 6.

In *Table* 5, we compute the error probability value for each ϵ_{d_j}, $0 < j < 13$. The sum of error probability values can be represented in interval form as $[2864.6 \times 10^{-4}, 3696.1 \times 10^{-4}]$. So, the probability of key recovery is computed as follows:

$$[(1-3696.1\times10^{-4}), (1-2864.6\times10^{-4})] = [0.630, 0.713].$$

B. PROBABILITY OF KEY RECOVERY IN THE ATTACK PROCEDURE BY BEIERLE ET AL. [2]

In CRYPTO 2020, Beierle et al. [4] introduced the new $\mathcal{ID} - \mathcal{OD}$ pair $\left(\Delta_{12}^{(0)}[6] - \Delta_{1}^{(3.5)}[0]\right)$. They used the concept of Right pair-based attack for finding the key-$\mathcal{IV}$ for all keys and observed that there exists no such pair for some keys. They found the bias ϵ_d for 3.5 rounds for the Right pair-based attack case. In the attack procedure for 7 round ChaCha, the bias observed in the backward direction from 7 rounds to 3.5 rounds is known as backward bias and is given by $\epsilon_a = 0.000610$. They have considered 74 key bits as PNBs. Using Equation 8, we have value of data complexity $N = 2^{43.83} = 15636683811300$. Using Equation 6, we have obtained the value of Θ as 7818352092492.251.

1) CRITICISM OF THE CLAIM

In [4], the authors have categorised the keys into "weak keys" (70%) and "strong keys" (30%). The authors proposed the attack only for the weak keys. To find an $\mathcal{IV}$ which forms a right pair for a particular weak key, we have to evaluate 2^5 iterations. However, this was an average value and did not completely fit with the previously existing complexity computation formula since the existing formula represented the maximum number of iterations, not the average. Aligning it with the existing formula, if we assume that the attacker would perform 2^5 iterations at most, we have experimentally examined that out of all keys, only for 47% of the keys we find an $\mathcal{IV}$ to form a right pair among 2^5 $\mathcal{IV}$s, and for the remaining 53% of the keys, we do not find an $\mathcal{IV}$ among the 2^5 iteration.

In *Table* 6, we have shown the percentage of keys for which we can get a suitable $\mathcal{IV}$ to form a right pair. To reach 70%, we need 2^{11} iterations.

2) COMPUTING THE PROBABILITY

Therefore, in order to compute the probability of key recovery, instead of the 70% − 30% breakdown, we have to use the 47%−53% breakdown. For the 53% keys for which we do not find a suitable $\mathcal{IV}$ to form a right pair, automatically, it would be a random $\mathcal{IV}$ based attack. So, in order to compute the probability of key recovery, we have to compute the same for both these two cases separately. Therefore, we consider both the attack approaches: the right pair and random $\mathcal{IV}$ based.

$$[(1-66.4\times10^{-4}), (1-37.1\times10^{-4})] = [0.993, 0.996].$$

a: RIGHT PAIR-BASED ATTACK (FOR 47% OF THE KEYS)

In this case, we consider only those key-$\mathcal{IV}$ pairs for which the minimum difference after the first round is 10, also called right pairs. The observed forward bias is $\epsilon_d = 0.00317$.

Here, we divide the bias value into 8 sub-intervals. After substituting the value of P_j and ϵ_j, we obtain the error

TABLE 5. Calculation of error probability for attack procedure by aumasson et al. in [1].

j	Pj	$\epsilon_{d_j} \times 10^4$	$\Phi(\mathcal{F}(\epsilon_j)) \times P_j \times 10^4$	$\epsilon_{d_{j+1}} \times 10^4$	$\Phi(\mathcal{F}(\epsilon_{j+1})) \times P_j \times 10^4$
0	0.055	0	549.9	1.0	549.9
1	0.040	1.0	399.9	1.5	399.9
2	0.045	1.5	449.9	2.0	448.1
3	0.040	2.0	398.3	2.5	382.5
4	0.080	2.5	765.0	3.0	623.4
5	0.080	3.0	623.4	3.5	346.0
6	0.085	3.5	367.6	4.0	113.4
7	0.105	4.0	140.1	5.0	1.4
8	0.155	5.0	2.0	6.25	0.0028
9	0.090	6.25	0.000036	7.5	0.000006
10	0.090	7.5	≈ 0	8.75	≈ 0
11	0.070	8.75	≈ 0	10.0	≈ 0
12	0.040	10.0	≈ 0	11.25	≈ 0
13	0.025	11.25	≈ 0	12.50	≈ 0
			Sum = 3696.1		Sum = 2864.6

TABLE 6. Percentage of keys for which we get $\mathcal{IV}$ to form a right pair.

Number of Iterations	32	64	128	256	512	1024	2048
Percentage	47	58	64	67.5	68.5	69	70

probability value and hence compute the probability of key recovery for the right pair-based attack.

From *Table* 7, we obtain the error probability for all 7 values of ϵ_{d_j} for the right pair-based attack. The sum of the error probabilities of these values lies in the interval $[37.1 \times 10^{-4}, 66.4 \times 10^{-4}]$. Using the formula mentioned in *Section V*, the interval of the probability of key recovery for the right pair-based attack can be computed as

b: RANDOM $\mathcal{IV}$ BASED ATTACK (FOR 53% OF THE KEYS)

In this case, a Random $\mathcal{IV}$ based attack is considered for every key. The observed forward bias is $\epsilon_d = 0.00050$. We compute the probability of key recovery for random $\mathcal{IV}$ based attack by the procedure explained in Subsection V-A.

In *Table* 8, we compute the error probability for a Random $\mathcal{IV}$ based attack. In this case, the bias value is divided into 14 sub-intervals. The interval of error probability is $[2795.4 \times 10^{-4}, 3381.7 \times 10^{-4}]$. So the probability of key recovery for the Random $\mathcal{IV}$ based attack lies in the interval

$$[(1-3381.7\times10^{-4}), (1-2795.4\times10^{-4})] = [0.661, 0.720].$$

As mentioned above, 47% of the keys form a right pair, and the remaining 53% keys do not find a suitable $\mathcal{IV}$ to form a right pair. Therefore, we compute the probability for all keys by adding the probabilities of key recovery of the right pair (47%) and random $\mathcal{IV}$ (53%) based attacks. The interval of the probability of key recovery is

$$[0.47 \times 0.993 + 0.53\times0.661, 0.47\times0.996+0.53\times0.720] = [0.816, 0.849].$$

C. PROBABILITY OF KEY RECOVERY IN THE ATTACK PROCEDURE BY DEY ET AL. [9]

For the 7-round attack, Dey et al. [5] obtained the value of forward bias ϵ_d for the same 3.5-round distinguisher. In the attack procedure, the backward bias observed in the direction from 7 rounds to 3.5 rounds is given by $\epsilon_a = 0.00057$. They have considered 79 key bits as PNBs. The obtained value of data complexity is N= $2^{44.89} = 32601419142621$. The value of Θ obtained from Equation 6 is 16300730460414.76.

In [5], Dey et al. have mentioned that out of all the keys, only 62% keys are exploitable, i.e., only for 62% of the keys we get a favorable $\mathcal{IV}$, which we store in memory. For the remaining 38% keys, we have to choose a Random $\mathcal{IV}$ based attack. Therefore, similar to Subsection V-B, we find the probabilities for both the right pair and random $\mathcal{IV}$ based attack.

1) RIGHT PAIR-BASED ATTACK (FOR 62% OF KEYS)

They observed the bias for the right pairs only. The observed forward bias is $\epsilon_d = 0.00317$.

TABLE 7. **Calculation of error probability for the right pair-based attack case in attack procedure by Beierle et al. in [4].**

j	P_j	$\epsilon_{d_j} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j \times 10^4$	$\epsilon_{d_{j+1}} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j \times 10^4$
0	0.0100	26.25	4.64	27.50	2.38
1	0.0882	27.50	20.9	28.75	9.94
2	0.2948	28.75	33.0	30.00	14.5
3	0.2637	30.00	1.30	30.85	6.9
4	0.2045	30.85	5.0	31.70	2.7
5	0.0967	31.70	1.30	32.50	0.68
6	0.0386	32.50	0.27	33.75	0.092
7	0.0035	33.75	0.0084	35.00	0.0028
			Sum = 66.4		Sum = 37.1

TABLE 8. **Calculation of error probability for the random $\mathcal{IV}$ based attack case in attack procedure by Beierle et al. in [4].**

j	P_j	$\epsilon_{d_j} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j \times 10^4$	$\epsilon_{d_{j+1}} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j \times 10^4$
0	0.055	0	398.7	1.0	352.0
1	0.040	1.0	256.0	1.5	237.6
2	0.045	1.5	267.3	2.0	246.1
3	0.040	2.0	218.8	2.5	200.0
4	0.080	2.5	400.0	3.0	361.6
5	0.080	3.0	361.6	3.5	324.0
6	0.085	3.5	344.2	4.0	305.1
7	0.105	4.0	376.9	5.0	287.7
8	0.155	5.0	424.7	6.25	285.2
9	0.090	6.25	165.6	7.5	103.5
10	0.090	7.5	103.5	8.75	59.4
11	0.070	8.75	46.2	10.0	24.5
12	0.040	10.0	14.0	11.25	6.8
13	0.025	11.25	4.2	12.5	1.9
			Sum = 3381.7		Sum = 2795.4

Table 9 demonstrates the computation of error probability for the right pair-based attack by Dey et al. From the table, we obtain the result that the error probability value lies in the interval $[103.64 \times 10^{-4}, 246.76 \times 10^{-4}]$. In a similar manner, the probability of key recovery in the right pair-based attack is calculated as

$$[(1-246.76\times10^{-4}), (1-103.64\times10^{-4})] = [0.975, 0.989].$$

2) RANDOM $\mathcal{IV}$ BASED ATTACK (FOR 38% OF KEYS)

Here, they also observed the bias for all the key-$\mathcal{IV}$ pairs. The observed forward bias is $\epsilon_d = 0.00050$. The computation of the probability of key recovery is the same as mentioned in Subsection V-B.

The calculation of obtaining the error probability for Random $\mathcal{IV}$ based attack by Dey et al. is explained in *Table* 10. The interval of error probability is given as $[2438.0 \times 10^{-4}, 3100.23 \times 10^{-4}]$. So the probability of key recovery for the Random $\mathcal{IV}$ based attack lies in the interval

$$[(1-3100.23\times10^{-4}), (1-2438.0\times10^{-4})] = [0.69, 0.756].$$

Now, to compute the probability of key recovery of all the keys, we add the values of probabilities of key recovery of both 62% exploitable (right pair) and 38% random $\mathcal{IV}$ keys.

TABLE 9. Calculation of error probability for the right pair-based attack case for attack procedure by Dey et al. in [5].

j	P_j	$\epsilon_{d_j} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j \times 10^4$	$\epsilon_{d_{j+1}} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j \times 10^4$
0	0.0100	26.25	112.3	27.50	51.55
1	0.0882	27.50	45.36	28.75	18.6
2	0.2948	28.75	62	30.00	21.6
3	0.2637	30.00	19	30.85	8.5
4	0.2045	30.85	6.6	31.70	2.8
5	0.0967	31.70	1.3	32.50	0.54
6	0.0386	32.50	0.2	33.75	0.050
7	0.0035	33.75	0.0045	35.00	0.0007
			Sum = 246.76		Sum = 103.64

TABLE 10. Calculation of error probability for the random $\mathcal{IV}$ based attack case in attack procedure by Dey et al. in [5].

j	P_j	$\epsilon_{d_j} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_j)\right) \times P_j \times 10^4$	$\epsilon_{d_{j+1}} \times 10^4$	$\Phi\left(\mathcal{F}(\epsilon_{j+1})\right) \times P_j \times 10^4$
0	0.055	0	435.0	1.0	356.4
1	0.040	1.0	273.6	1.5	250.0
2	0.045	1.5	281.2	2.0	253.3
3	0.040	2.0	225.2	2.5	200.0
4	0.080	2.5	400.0	3.0	348.8
5	0.080	3.0	348.8	3.5	299.2
6	0.085	3.5	31.7.9	4.0	267.7
7	0.105	4.0	330.7	5.0	218.4
8	0.155	5.0	322.4	6.25	172.0
9	0.090	6.25	99.9	7.5	47.3
10	0.090	7.5	47.34	8.75	18.9
11	0.070	8.75	14.77	10.0	5.1
12	0.040	10.0	2.92	11.25	0.8
13	0.025	11.25	0.50	12.5	0.1
			Sum = 3100.23		Sum = 2438.0

The interval of the probability of key recovery is

$$[0.62\times0.975+0.38\times0.69, 0.62\times0.989+0.38\times0.756] = [0.866, 0.900].$$

VI. CONCLUSION

In this paper, we analyze the probability of key recovery in differential attacks against ChaCha. Until now, the assessment of new ideas in this line of attacks has predominantly focused on the improvement in complexity. However, our contribution introduces another parameter for the comprehensive evaluation of any new idea. In future attacks on ChaCha, incorporating a section in their contributions that outlines the probability of key recovery adds another layer to their approach.

One research direction in this domain is the analysis of how the choice of keys influences the backward bias ϵ_a. A study on the distribution of ϵ_a and its integration with the distribution of ϵ_d to obtain an overall distribution of ϵ is an important problem that can be explored in the future.

REFERENCES

[1] J.-P. Aumasson, S. Fischer, S. Khazaei, W. Meier, and C. Rechberger, "New features of Latin dances: Analysis of Salsa, ChaCha, and Rumba," in *Fast Softw. Encryption*. Berlin, Heidelberg: Springer, 2008, pp. 470–488.

[2] *ChaChaApplications.*

[3] (2019). *AdiantumCipher*. [Online]. Available: https://www.johndcook.com/blog/2019/03/02/chacha-adiantum/

[4] C. Beierle, M. Broll, F. Canale, N. David, A. Flórez-Gutiérrez, G. Leander, M. Naya-Plasencia, and Y. Todo, "Improved differential-linear attacks with applications to ARX ciphers," *J. Cryptol.*, vol. 35, no. 4, pp. 329–358, Oct. 2022, doi: 10.1007/s00145-022-09437-z.

[5] S. Dey, H. K. Garai, S. Sarkar, and N. K. Sharma, "Revamped differential-linear cryptanalysis on reduced round ChaCha," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Cham: Springer, 2022, pp. 86–114.

[6] D. Bernstein. (2008). *ChaCha, a Variant of Salsa20*. [Online]. Available: https://cr.yp.to/chacha/chacha-20080120.pdf

[7] Z. Shi, B. Zhang, D. Feng, and W. Wu, "Improved key recovery attacks on reduced-round Salsa20 and ChaCha," in *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 337–351.

[8] S. Maitra, "Chosen IV cryptanalysis on reduced round ChaCha and Salsa," *Discrete Appl. Math.*, vol. 208, pp. 88–97, Jul. 2016, doi: 10.1016/j.dam.2016.02.020.

[9] A. R. Choudhuri and S. Maitra, "Significantly improved multi-bit differentials for reduced round Salsa and ChaCha," in *Proc. IACR Trans. Symmetric Cryptol.*, 2017, pp. 261–287.

[10] S. Dey and S. Sarkar, "Improved analysis for reduced round Salsa and ChaCha," *Discrete Appl. Math.*, vol. 227, pp. 58–69, Aug. 2017, doi: 10.1016/j.dam.2017.04.034.

[11] S. Dey, T. Roy, and S. Sarkar, "Revisiting design principles of Salsa and ChaCha," *Adv. Math. Commun.*, vol. 13, no. 4, pp. 689–704, 2019, doi: 10.3934/amc.2019041.

[12] M. Coutinho and T. C. S. Neto, "New multi-bit differentials to improve attacks against ChaCha," *Cryptol. ePrint Arch.*, pp. 1–16, 2020.

[13] M. Coutinho and T. C. Souza Neto, "Improved linear approximations to ARX ciphers and attacks against ChaCha," in *Lecture Notes in Computer Science*, A. Canteaut and F.-X. Standaert, Eds. Springer, 2021, pp. 711–740.

[14] S. Dey, C. Dey, S. Sarkar, and W. Meier, "Revisiting cryptanalysis on ChaCha from crypto 2020 and eurocrypt 2021," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 6114–6133, Sep. 2022, doi: 10.1109/TIT.2022.3171865. https://doi.org/10.1109/TIT.2022.3171865

[15] S. Dey and S. Sarkar, "A theoretical investigation on the distinguishers of salsa and ChaCha," *Discrete Appl. Math.*, vol. 302, pp. 147–162, Oct. 2021, doi: 10.1016/j.dam.2021.06.017.

[16] M. Coutinho, I. Passos, J. C. Grados Vásquez, F. L. L. de Mendonça, R. T. de Sousa, and F. Borges, "Latin dances reloaded: Improved cryptanalysis against Salsa and ChaCha, and the proposal of Forró," in *Advances in Cryptology—ASIACRYPT*. Cham, Switzerland: Springer, 2022, pp. 256–286.

[17] M. Coutinho, I. Passos, J. C. G. Vásquez, S. Sarkar, F. L. L. de Mendonça, R. T. de Sousa, and F. Borges, "Latin dances reloaded: Improved cryptanalysis against Salsa and ChaCha, and the proposal of Forró," *J. Cryptol.*, vol. 36, no. 3, p. 18, Jul. 2023, doi: 10.1007/s00145-023-09455-5.

[18] S. Miyashita, R. Ito, and A. Miyaji, "PNB-focused differential cryptanalysis of ChaCha stream cipher," in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2021, pp. 46–66.

[19] S. Dey, H. K. Garai, S. Sarkar, and N. K. Sharma, "Enhanced differential-linear attacks on reduced round ChaCha," *IEEE Trans. Inf. Theory*, vol. 69, no. 8, pp. 5318–5336, Aug. 2023, doi: 10.1109/TIT.2023.3269790. https://doi.org/10.1109/TIT.2023.3269790

[20] S. Dey, H. K. Garai, and S. Maitra, "Cryptanalysis of reduced round ChaCha—New attack & deeper analysis," *IACR Trans. Symmetric Cryptol.*, pp. 89–110, Mar. 2023.

[21] S. Wang, M. Liu, S. Hou, and D. Lin, "Moving a step of ChaCha in syncopated rhythm," in *Advances in Cryptology—CRYPTO*, Cham, Switzerland: Springer, 2023, pp. 273–304.

[22] R. Wu, S. Gao, X. Wang, S. Liu, Q. Li, U. Erkan, and X. Tang, "AEA-NCS: An audio encryption algorithm based on a nested chaotic system," *Chaos, Solitons Fractals*, vol. 165, Dec. 2022, Art. no. 112770. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077922009493

[23] S. Gao, R. Wu, X. Wang, J. Wang, Q. Li, C. Wang, and X. Tang, "A 3D model encryption scheme based on a cascaded chaotic system," *Signal Process.*, vol. 202, Jan. 2023, Art. no. 108745. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168422002845

[24] S. Gao, R. Wu, X. Wang, J. Liu, Q. Li, and X. Tang, "EFR-CSTP: Encryption for face recognition based on the chaos and semi-tensor product theory," *Inf. Sci.*, vol. 621, pp. 766–781, Apr. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025522014323

[25] N. K. Sharma. (2024). *Analyzing the Probability of Key Recovery in the Differential Attacks Against ChaCha*. [Online]. Available: https://codeocean.com/capsule/0745999/tree/v1

[26] S. Fischer, W. Meier, C. Berbain, J.-F. Biasse, and M. J. B. Robshaw, "Non-randomness in eSTREAM candidates Salsa20 and TSC-4," in *Progress in Cryptology—INDOCRYPT*. Cham, Switzerland: Springer, 2006, pp. 2–16.

NITIN KUMAR SHARMA received the B.Sc. (Hons.) and M.Sc. degrees in mathematics from Himachal Pradesh University, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Birla Institute of Technology and Science (BITS) Pilani, Hyderabad Campus, India. His research interest includes symmetric key cryptanalysis.

SABYASACHI DEY received the B.Sc. degree from the University of Calcutta, in 2013, and the M.Sc. and Ph.D. degrees in mathematics from the Indian Institute of Technology Madras, Chennai, India, in 2015 and 2018, respectively. He received the INSPIRE Scholarship (B.Sc.-M.Sc.) from the Department of Science and Technology (DST), India. He also received the National Board for Higher Mathematics Fellowship (Ph.D.) from the Department of Atomic Energy (DAE), India. In 2019, he joined the Birla Institute of Technology and Science (BITS) Pilani, Hyderabad Campus, India, as an Assistant Professor. His main research interest includes symmetric key cryptology.