Scilight

Pragmatic Cybersecurity
https://www.sciltp.com/journals/pc

Article

The Related Key Boomerang Attack on ChaCha Permutation Function †

Nasratullah Ghafoori [1] and Atsuko Miyaji [2,*]

[1] Opening Line Inc., Minato-ku, Tokyo 105-0003, Japan
[2] Graduate School of Engineering, The University of Osaka, Suita, Osaka 565-0871, Japan
* Correspondence: miyaji@comm.eng.osaka-u.ac.jp

† Extended from a non-archival presentation titled: The Related Key Boomerang Attack on ChaCha Permutation Function. In Proceedings of the 31st Australasian Conference on Information Security and Privacy Conference, Perth, WA, Australia, 6–9 July 2026.

How To Cite: Ghafoori, N.; Miyaji, A. The Related Key Boomerang Attack on ChaCha Permutation Function. *Pragmatic Cybersecurity* **2026**, *1*(2), 16. https://doi.org/10.53941/pc.2026.100016

Received: 30 April 2026
Revised: 29 July 2026
Accepted: 31 July 2026
Published: 7 September 2026

Abstract: The ChaCha stream cipher is widely employed in both hardware and software implementations. Although its security has been extensively studied through differential and differential-linear cryptanalysis, its vulnerability to variants of differential cryptanalysis remains largely unaddressed. This paper investigates the related-key boomerang attack on the ChaCha permutation function. The boomerang attack leverages differential analysis by decomposing the primitive into two sub-components, an approach that proves particularly efficient in cases where identifying high-probability differentials over the entire primitive is infeasible. The related-key boomerang attack incorporates four distinct yet related keys alongside the input differentials to mount the attack. This paper presents a related-key boomerang analysis of the ChaCha 6 and ChaCha 7 permutation functions. Additionally, an attack algorithm targeting reduced-round variants of ChaCha is presented, establishing a correlation among input differences, output differences, and key-bit positions. Consequently, we propose a distinguisher for the ChaCha 6 permutation function with a data complexity of approximately $2^{2.03}$. Furthermore, we evaluate the ChaCha 7 variant and demonstrate that, for certain attack configurations, it exhibits no detectable bias, thereby establishing a well-defined security boundary for this attack.

Keywords: symmetric key cryptography; ChaCha permutation; related-key cryptanalysis; boomerang cryptanalysis; differential cryptanalysis

1. Introduction

Differential cryptanalysis [1] was originally proposed to study the security of the DES [2] block cipher. Subsequent research introduced several variants of differential cryptanalysis, including truncated differentials [3], impossible differentials [4], related-key attacks [5], and boomerang attacks [6]. The boomerang attack treats the cipher E as a cascade composition $E = E_0 \cdot E_1$. It combines the differentials of the sub-ciphers E_0 and E_1 into a unified differential to distinguish the entire cipher E using adaptively chosen plaintexts and ciphertexts. The boomerang attack demonstrates that the absence of a high-probability differential over the full cipher E does not guarantee security against differential cryptanalysis. Researchers have introduced several variants of the boomerang attack, including the related-key boomerang and rectangle attacks [7], the retracing boomerang attack [8], and the amplified boomerang attack [9]. The related-key differential attack assumes that the attacker controls the key-bit differences, plaintexts, and ciphertexts.

The related-key boomerang attack combines related-key differentials with the boomerang framework. According to [10], the related-key attack enables the exploitation of high-probability related-key differentials in the sub-ciphers E_0 and E_1 across both components. The related-key boomerang attack is generally more effective than other combined attack strategies, such as the related-key impossible differential attack [11] and the related-key differential-linear attack [12].

Nevertheless, the ChaCha stream cipher [13] has been extensively studied against differential and differential-

linear attacks, as shown in Tables 1 and 2. However, no study has examined its security against variants of differential cryptanalysis, such as the related-key boomerang attack. This gap may expose potential vulnerabilities in the underlying permutation. For instance, COCONUT98 [14] was initially shown to be resistant to standard differential cryptanalysis, yet was later compromised by the boomerang attack.

Table 1. Comparison of attacks on ChaCha 6.

Method (Model)	Rounds	Time	Data	Ref.
Diff. Distinguisher (Stream)	6	2^{116}	2^{116}	[15]
Diff. Distinguisher (Stream)	6	2^{51}	2^{51}	[16]
Diff. KR (Stream)	6	2^{139}	2^{30}	[17]
Diff. KR (Stream)	6	$2^{127.5}$	$2^{37.5}$	[15]
Diff. KR (Stream)	6	$2^{77.4}$	2^{58}	[18]
Diff. KR (Stream)	6	2^{136}	2^{28}	[19]
Amplified Boomerang (Stream)	6	2^{36}	2^{36}	[20]
RK Boomerang (Perm.)	6	$\approx 2^{2.03}$	$\approx 2^{2.03}$	This work

* All prior attacks in this table target the ChaCha stream cipher, whereas this work targets the ChaCha permutation function. This comparison is intended to survey existing studies, not to directly compare with our attack.

Table 2. Comparison of attacks against ChaCha 7.

Method (Model)	Rounds	Time	Data	Ref.
Diff. Distinguisher (Stream)	7	2^{224}	2^{224}	[16]
Diff. Distinguisher (Stream)	7	2^{214}	2^{214}	[21]
Diff. Distinguisher (Stream)	7	2^{207}	2^{207}	[22]
Diff. Distinguisher (Stream)	7	$2^{166.89}$	$2^{166.89}$	[23]
Diff. Distinguisher (Stream)	7	$2^{120.9}$	$2^{120.9}$	[24]
Diff. Distinguisher (Stream)	7	$2^{122.5}$	$2^{122.5}$	[25]
Diff. KR (Stream)	7	2^{248}	2^{27}	[17]
Diff. KR (Stream)	7	$2^{237.7}$	2^{96}	[15]
Diff. KR (Stream)	7	$2^{230.86}$	$2^{48.8}$	[18]
Diff. KR (Stream)	7	$2^{221.95}$	$2^{48.83}$	[26]
Diff. KR (Stream)	7	$2^{231.63}$	$2^{49.58}$	[27]
Diff. KR (Stream)	7	$2^{210.3}$	$2^{103.3}$	[28]
Diff. KR (Stream)	7	$2^{206.8}$	$2^{110.81}$	[23]
Diff. KR (Stream)	7.25	$2^{254.01}$	$2^{51.81}$	[29]
Amplified Boomerang (Stream)	7	$2^{35.9}$	$2^{35.9}$	[20]
RK Boomerang (Perm.)	7	-	Bound Found	This work

* All prior attacks in this table target the ChaCha stream cipher, whereas this work targets the ChaCha permutation function. This comparison is intended to survey existing studies, not to directly compare with our attack.

ChaCha is known to resist conventional differential attacks up to 3.5 rounds. Existing studies have employed differentials spanning either 3 or 3.5 rounds in combination with 2.5 or 3 rounds of linear approximation to attack ChaCha 6, ChaCha 7, and ChaCha 7.5 as reported in Tables 1 and 2. Despite nearly fifteen years of cryptanalytic effort, these attacks have remained limited in scope. A recent study [30,31] extensively documented the attacks of the preceding fourteen years, all of which were either differential or differential-linear in nature. Aumasson [32] also noted this trend, characterising it as applying the same technique with progressively refined analysis, and suggested adopting ChaCha 8 in place of ChaCha 20 to improve performance without compromising security. This observation highlights a new avenue for the cryptanalysis of ChaCha, aimed at deepening the understanding of its security properties and potential vulnerabilities under different adversarial models.

It is important to note that, while the prior works listed in Tables 1 and 2 primarily target the ChaCha stream cipher, the present work targets the ChaCha permutation function. This stronger adversarial model admits lower

data complexity, thereby providing insight into the structural properties of the permutation itself, rather than those of the mode of operation. The structure of ChaCha remains unchanged in the proposed adversarial model.

The effectiveness and versatility of the related-key boomerang attack [10], together with [33] work on KATAN32/48/64, motivated the present work to investigate the related-key boomerang attack and present novel attacks on ChaCha 6 and ChaCha 7. Both theoretical and experimental analyses are conducted to evaluate the results comprehensively.

2. Contributions

In this paper, we present a related-key boomerang attack on the ChaCha permutation function. We define an attack strategy for selecting key-bit and output differential positions, and establish a distinguisher for ChaCha 6 and ChaCha 7.

Outline of the Paper

This paper is organised as follows. Section 3 introduces the structure of the ChaCha permutation, the notation and abbreviations employed, an overview of existing studies, and an explanation of the boomerang and related-key boomerang attacks. Section 4 presents the proposed attacks and the corresponding results for ChaCha 6 and ChaCha 7. Section 5 concludes the paper and discusses directions for future research.

3. Preliminaries

This section provides an overview of the ChaCha stream cipher and introduces the notation and abbreviations used throughout this paper. In addition, the relevant research background is summarised in Tables 1 and 2. The methods underlying the proposed attack are also presented.

3.1. ChaCha Permutation vs. Stream Cipher

It is crucial to distinguish between the ChaCha Stream Cipher and the Permutation.

- Stream Cipher Mode: The input state $X^{(0)}$ is constructed from a 256-bit Key, 64-bit Nonce, 64-bit Counter, and 128-bit Constants. The attacker controls only the Nonce and Counter.
- Permutation Mode: The attacker has full control over the 512-bit initial state $X^{(0)}$. In this work, we treat the traditionally defined "Key" words simply as part of the editable input state. This model evaluates the pure structural security of the permutation P.

Because the permutation-level adversary considered in this work strictly subsumes the capabilities of a stream-cipher adversary (the latter can vary only the nonce and counter, a small subset of the full 512-bit input), a distinguisher obtained under the stronger model bounds the structural resistance of the ChaCha permutation itself, but does not by itself imply a corresponding attack against ChaCha when used as a stream cipher. In the stream-cipher construction, the keystream is computed as $Z = X^{(0)} + X^{(R)}$, so an attacker who cannot freely choose $X^{(0)}$, and in particular the key words, observes only a feed-forward-masked version of the internal permutation difference. The complexities reported in this paper should therefore be read as characterising the security margin of the ChaCha permutation as a structural primitive, rather than as a direct claim about the practical security of the ChaCha stream cipher. This is also why the comparisons in Tables 1 and 2 are presented as background context on the existing literature rather than as a head-to-head ranking against prior stream-cipher results.

3.2. ChaCha Stream Cipher

The ChaCha stream cipher works through the following three steps to generate a keystream block of 16 words, with each word having a size of 32 bits:

Step 1. To generate a 512-bit key stream, the ChaCha initial state matrix $X^{(0)}$ of order 4×4 is initialized from a 256-bit secret key $k = (k_0, k_1, \ldots, k_7)$, a 128-bit initial vector $v = (v_0, v_1, v_2, v_3)$, and four constants $c = (c_0, c_1, c_2, c_3)$, such that $c_0 = 0\text{x}61707865$, $c_1 = 0\text{x}3320646e$, $c_2 = 0\text{x}79622d32$, and $c_3 = 0\text{x}6b206574$. After initialization, we get the following initial state matrix:

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

Step 2. The round function of ChaCha consists of four simultaneous computations of the quarterround

function. According to the procedure, a vector $(x_a^{(r)}, x_b^{(r)}, x_c^{(r)}, x_d^{(r)})$ in the internal state matrix $X^{(r)}$ is updated by sequentially computing the following:

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

For odd rounds, called columnrounds, the quarterround function is applied to the following four column vectors: $(x_0^{(r)}, x_4^{(r)}, x_8^{(r)}, x_{12}^{(r)})$, $(x_1^{(r)}, x_5^{(r)}, x_9^{(r)}, x_{13}^{(r)})$, $(x_2^{(r)}, x_6^{(r)}, x_{10}^{(r)}, x_{14}^{(r)})$, and $(x_3^{(r)}, x_7^{(r)}, x_{11}^{(r)}, x_{15}^{(r)})$. For even-numbered rounds, which are called diagonalrounds, the quarterround function is applied to the following four diagonal vectors: $(x_0^{(r)}, x_5^{(r)}, x_{10}^{(r)}, x_{15}^{(r)})$, $(x_1^{(r)}, x_6^{(r)}, x_{11}^{(r)}, x_{12}^{(r)})$, $(x_2^{(r)}, x_7^{(r)}, x_8^{(r)}, x_{13}^{(r)})$, and $(x_3^{(r)}, x_4^{(r)}, x_9^{(r)}, x_{14}^{(r)})$.

Step 3. A 512-bit keystream block is computed as $Z = X^{(0)} + X^{(R)}$, where R is the last round. The original version of ChaCha has $R = 20$ rounds, and the ChaCha20/R denotes the reduced-round version.

The round function of ChaCha is reversible. In other words, an input vector $(x_a^{(r+1)}, x_b^{(r+1)}, x_c^{(r+1)}, x_d^{(r+1)})$ in the internal state matrix $X^{(r+1)}$ is backdated by sequentially computing the following:

$$\begin{cases} x_{b'''}^{(r)} = x_b^{(r+1)} \lll 25, & x_{b''}^{(r)} = x_{b'''}^{(r)} \oplus x_c^{(r+1)}, & x_{c'}^{(r)} = x_c^{(r+1)} - x_d^{(r+1)}, \\ x_{d'''}^{(r)} = x_d^{(r+1)} \lll 24, & x_{d''}^{(r)} = x_{d'''}^{(r)} \oplus x_a^{(r+1)}, & x_{a'}^{(r)} = x_a^{(r+1)} - x_{b''}^{(r)}, \\ x_{b'}^{(r)} = x_{b''}^{(r)} \lll 20, & x_b^{(r)} = x_{b'}^{(r)} \oplus x_{c'}^{(r)}, & x_c^{(r)} = x_{c'}^{(r)} - x_{d''}^{(r)}, \\ x_{d'}^{(r)} = x_{d''}^{(r)} \lll 16, & x_d^{(r)} = x_{d'}^{(r)} \oplus x_{a'}^{(r)}, & x_a^{(r)} = x_{a'}^{(r)} - x_b^{(r)} \end{cases} \tag{2}$$

3.3. Differential Cryptanalysis

Biham and Shamir introduced differential cryptanalysis [1] as a systematic method for analysing the security of DES [2]. This approach subsequently became a prominent technique for evaluating the security of various cryptographic primitives. Differential cryptanalysis is a statistical chosen-plaintext attack that exploits differences between plaintext pairs and their corresponding ciphertext pairs. More precisely, it aims to utilise the effect of an input difference ($\mathcal{ID}$) over n rounds to determine the probability of a particular output difference. Consider an iterative cipher $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ consisting of r rounds. An r-round differential characteristic with probability p for a cipher E is defined as follows: given an initial $\mathcal{ID}$ denoted α, such that $X \oplus X' = \alpha$, there exists a probability p that the resulting output difference ($\mathcal{OD}$) after r rounds satisfies $X^{(r)} \oplus X'^{(r)} = \beta$. Formally,

$$\Pr\big(X^{(r)} \oplus X'^{(r)} = \beta \mid X^{(0)} \oplus X'^{(0)} = \alpha\big) = p,$$

which is compactly denoted as $\alpha \to \beta$.

3.4. Related-Key Attack

As discussed in Section 3.3, differential attacks analyse the difference between a plaintext difference ΔP (or α) and a ciphertext difference ΔC (or β), such that

$$\Pr_{P,\,K}\big[E_K(P) \oplus E_K(P \oplus \Delta P) = \Delta C\big],$$

where the probability p deviates from that of a random permutation, that is, it is either anomalously high or zero [4]. A related-key differential cryptanalysis [5] is characterised by a triplet comprising a plaintext difference α, a ciphertext difference β, and a key difference ΔK, such that

$$\Pr_{P,\,K}\big[E_K(P) \oplus E_{K\oplus\Delta K}(P \oplus \Delta P) = \Delta C\big],$$

where the probability p deviates from that of a random permutation, namely when it is either anomalously high or zero [4]. Keys bearing special structural relationships are employed to encrypt the plaintexts. The attacker is assumed to know the relationship between the two keys, but not the keys themselves. The attacker then exploits

the resulting ciphertexts to recover the keys. In this paper, the XOR difference is employed to relate the keys, i.e., $K \oplus \lambda = \Delta K$.

3.5. The Boomerang Attack

Wagner introduced the boomerang attack [6] as a technique to demonstrate that the absence of extended high-probability differentials does not guarantee a cipher's security against differential cryptanalysis. The motivation for the boomerang attack is evident, as in many ciphers it is more straightforward to search for short differentials with high probability than for a long differential with low probability. The concept involves encrypting two plaintexts with a single-bit difference through the cipher, introducing a single-bit difference in the resulting ciphertext, and studying their return path (see Figure 1). The second differential within the cipher connects the differentials between the upper and lower parts of the cipher, allowing the attack to be mounted. We assume that a cipher $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ can be described as $E = E_1 \cdot E_0$, where E_0 exhibits a differential characteristic $\alpha \xrightarrow{p} \beta$ and E_1 exhibits a differential characteristic $\gamma \xrightarrow{q} \delta$. As depicted in Figure 1, the existence of δ is guaranteed by the condition $pq \geq 2^{-n/2}$. Algorithm 1 enables the differentiation of the cipher E from a random permutation. In the case of a random permutation, the probability of occurrence of step 7 in Algorithm 1 is $2^{-n/2}$, where n is the block size. In contrast, for a cipher E, the attack can be executed simultaneously for all possible values of β and γ, as long as the condition $\beta \neq \gamma$ is satisfied. Consequently, a right quartet for E is encountered with a probability of no less than $(pq)^2$, where

$$p = \sqrt{\sum_{\beta} \Pr[\alpha \to \beta]^2}, \qquad q = \sqrt{\sum_{\gamma} \Pr[\gamma \to \delta]^2}.$$

In the context of the boomerang process described in Algorithm 1, the target cipher E can be distinguished from a random oracle given $\mathcal{O}((pq)^{-2})$ adaptively chosen plaintexts and ciphertexts, where $pq \gg 2^{-n/2}$ [10].

Algorithm 1 The Boomerang Distinguisher Attack Algorithm [8]

Input: The initial states (X_1, X_2) with input difference α
Output: Identification of the cipher E or a random permutation

1: Initialize a counter **ctr** $\leftarrow 0$
2: Generate $(pq)^{-2}$ unique plaintext pairs (X_1, X_2) with input difference α
3: **for all** pairs (X_1, X_2) **do**
4: Ask for the encryption of (X_1, X_2) to (C_1, C_2)
5: Compute $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$
6: Ask for the decryption of (C_3, C_4) to (X_3, X_4)
7: **if** $(X_3 \oplus X_4) = \alpha$ **then**
8: Increment **ctr**
9: **end if**
10: **end for**
11: **if ctr** ≥ 1 **then**
12: **Return:** This is the cipher E
13: **else**
14: **Return:** This is a random permutation
15: **end if**

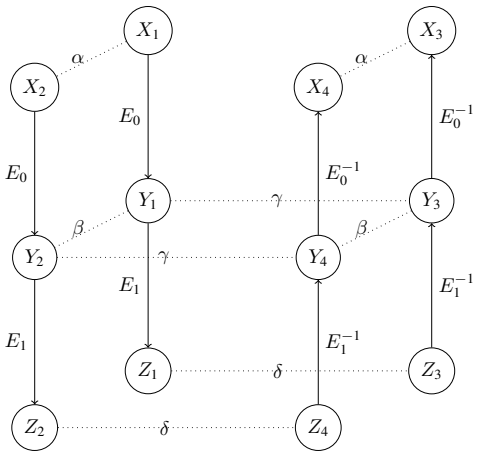

Figure 1. A schematic of the basic boomerang attack.

3.6. Related-Key Boomerang Attack

The related-key boomerang attack [5] employs the key difference λ alongside the plaintext difference. To illustrate, consider the differential $\alpha \xrightarrow{E_0} \beta$ with a key difference λ_1 and probability $\hat{p}$, and the differential $\gamma \xrightarrow{E_1} \delta$ with a key difference λ_2 and probability $\hat{q}$. A related-key boomerang distinguisher can be constructed using the plaintext differences $\alpha \xrightarrow{E_0} \beta$ and $\gamma \xrightarrow{E_1} \delta$, together with four secret but related keys. The keys are defined as follows: $K_1,\ K_2 = K_1 \oplus \Delta K_0,\ K_3 = K_1 \oplus \Delta K_1,\ K_4 = K_1 \oplus \Delta K_0 \oplus \Delta K_1$, or alternatively expressed in terms of $\lambda_1, \lambda_2, \lambda_3$. The attacker may employ multiple differentials for E_0 and E_1; however, this is subject to the stringent requirement that all related-key differentials used in E_0 share the same key difference ΔK_0 and the same input difference α. Moreover, the related-key differentials employed in E_1 must share the same key difference ΔK_1 and the same output difference δ. Algorithm 2 can distinguish between a cipher and a random permutation. For a random permutation, the condition $X_3 \oplus X_4 = \alpha$ in Algorithm 2 occurs with probability 2^{-n}. The probability that the condition $X_3 \oplus X_4 = \alpha$ is satisfied for a cipher E is established by [10] as follows: consider a quartet (X_1, X_2, X_3, X_4) constructed by Algorithm 2; we have

$$\begin{aligned} Pr[X_3 \oplus X_4 = \alpha] = \sum_{\beta_1 \oplus \beta_2 \oplus \gamma_1 \oplus \gamma_2 = 0} & Pr[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1] \cdot Pr[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2] \\ & \cdot Pr[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta] \cdot Pr[\gamma_2 \xrightarrow[\gamma_2 K_{ab}]{E_1} \delta] \end{aligned} \tag{3}$$

In particular,

$$Pr[X_3 \oplus X_4 = \alpha] \geq (\hat{p}\hat{q})^2 \tag{4}$$

where

$$\hat{p} = \sqrt{\sum_{\beta'} Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta'\right]^2}, \quad \hat{q} = \sqrt{\sum_{\gamma'} Pr\left[\gamma' \xrightarrow[\Delta K_{ac}]{E_1} \delta\right]^2}.$$

If $\hat{p}\hat{q} > 2^{-n/2}$, then the likelihood of the event $X_3 \oplus X_4 = \alpha$ occurring is greater for the set E compared to a random permutation. In other words, we anticipate more quartets for E.

Algorithm 2 The Related-key Boomerang Distinguisher Algorithm

Input: The initial states (X_1, X_2) with input difference α and key difference λ_1
Output: Obtaining related-key boomerang distinguisher

1: Initialize a counter **ctr** $\leftarrow 0$.
2: Generate $(pq)^{-2}$ unique plaintext pairs (X_1, X_2) with an input difference α and key difference λ_1.
3: **for all** pairs (X_1, X_2) **do**
4: Encrypt the X_1 with K_1 and X_2 with $K_2 = K_1 \oplus \Delta K_0$ to obtain (C_1, C_2).
5: Compute $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$.
6: Decrypt C_3 with $K_3 = K_1 \oplus \Delta K_1$ to X_3 and C_4 with $K_4 = K_1 \oplus \Delta K_0 \oplus \Delta K_1$ to X_4.
7: **if** $X_3 \oplus X_4 = \alpha$ **then**
8: Increment **ctr**
9: **end if**
10: **end for**
11: **if ctr** ≥ 1 **then**
12: **Return:** This is the ChaCha Permutation.
13: **else**
14: **Return:** This is a random permutation.
15: **end if**

4. The Related-Key Boomerang Attack on the ChaCha Permutation Function

Considering the structure of ChaCha and the related-key boomerang attack framework, it is essential to mount the attack with a careful and selective strategy. The attack is divided into two phases. In the first phase, the positions of attack (i.e., $\mathcal{ID}$, $\mathcal{OD}$, and key-bit positions) are identified. In the second phase, Algorithm 3 is developed and executed based on the results of the first phase. It should be noted that, in the context of attacking the ChaCha permutation function, the adversary is granted access to the full 512-bit input. As the attack exploits differences involving key-bit positions within the ChaCha matrix in conjunction with the $\mathcal{ID}$, we adopt the terminology of a

related-key attack. Nevertheless, since the attack operates at the permutation level rather than the stream cipher level, the related-input attack may be considered equally valid. Please refer to Figure 2.

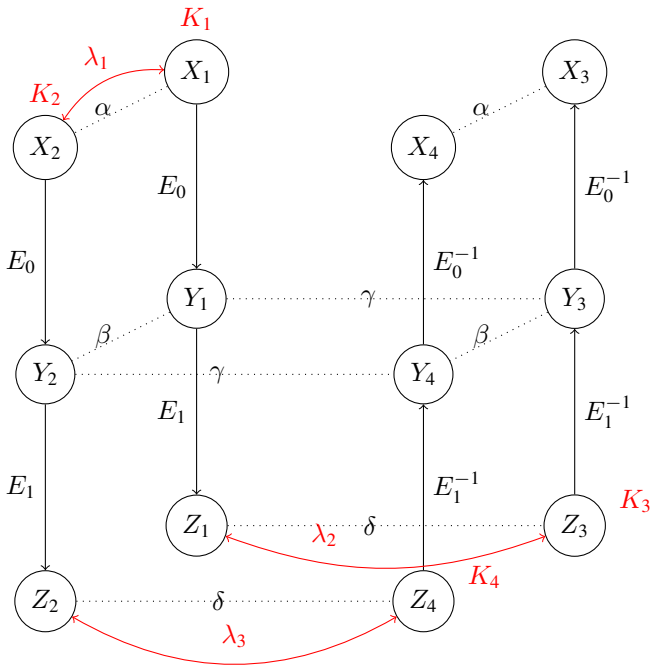

Figure 2. A schematic of the basic related-key boomerang attack.

4.1. Attack Points Selection Phase

In this section, the ChaCha stream cipher structure is used to identify the $\mathcal{ID}$, $\mathcal{OD}$, and key-bit positions for the related-key part of the attack. The ChaCha stream cipher algorithm provides 128 $\mathcal{ID}$ positions, 256 key-bit positions, and 512 $\mathcal{OD}$ positions for selection. More precisely, for every single $\mathcal{ID}$ position, there are 256 possible key-bit positions and 512 potential $\mathcal{OD}$ positions to choose from. As ChaCha is based on an Addition, Rotation, and XOR (ARX) structure, forming a difference distribution table (DDT) is not feasible. Given ChaCha's non-linearity and diffusion properties, a single-bit input difference in the key or IV may cause numerous output differences. In [10], the author employed key structures to mitigate the impact of this occurrence on the differentials. In the present work, the concept of the neutrality measure is adopted, as it is an effective method for measuring the impact of key-bits on the ciphertext within the current ChaCha research domain. Although the neutrality measure was originally introduced in the context of differential attacks on ChaCha [17], it serves as a natural selection criterion in the related-key boomerang framework. As noted above, the DDT is not feasible for ARX-based ciphers, and ChaCha's lack of a key schedule further eliminates key schedule analysis as an alternative. The neutrality measure therefore provides the most practical and justified means of identifying key-bit positions.

The neutrality measure analyses how much each key-bit in ChaCha influences the final output $\mathcal{OD}$. This influence is quantified by the neutrality measure, ranging from 0 to 1. Bits with higher neutrality measures contribute less to the overall change in $\mathcal{OD}$, thereby revealing their diminished significance in the output of ChaCha. This insight is valuable in cryptanalysis, as attackers can prioritise manipulating bits with lower impact, potentially simplifying their attack strategies. The neutrality measure is further utilised to identify input bits that have a limited propagation effect on the state, thereby slowing down the avalanche effect and maintaining high-probability differential characteristics for more rounds. For further details on neutrality measures, refer to [17,29].

In addition, the $\mathcal{OD}$ positions that are least affected by the $\mathcal{ID}$ and key-bit positions are identified. For this purpose, Algorithm 4 from [29] is executed and the results are presented as follows. To pinpoint the exact positions of $\mathcal{ID}$, key-bits, and $\mathcal{OD}$, the effect among an $\mathcal{ID}$, 256 key-bits, and 512 potential $\mathcal{OD}$ positions is evaluated. This analysis was conducted for both $r = 3$ and $r = 3.5$, please refer to Table 3. It was found that the key-bit positions $159, 191, 223, 255$, or equivalently $K_8[31], K_9[31], K_{10}[31], K_{11}[31]$, representing the least significant bits of the aforementioned key-word positions, had minimal impact on $\mathcal{OD}$ (refer to Table 4). Consequently, these positions were selected as key-bit positions to flip for the related-key part of the attack. Additionally, for the $\mathcal{OD}$ positions at $r = 3$, the positions $Y_4^{(3)}[0], Y_5^{(3)}[0], Y_6^{(3)}[0], Y_7^{(3)}[0]$ were the least affected $\mathcal{OD}$ positions by the key-bits, while for $r = 3.5$, the positions $Y_8^{(3.5)}[0], Y_9^{(3.5)}[0], Y_{10}^{(3.5)}[0], Y_{11}^{(3.5)}[0]$ were the least impacted $\mathcal{OD}$ positions by the difference

in IV and key-bit positions (refer to Table 4). Based on these findings, the positions identified above were selected as attack points to execute the related-key boomerang attack on ChaCha. The standard ChaCha constants ($c_0 \dots c_3$) are used across all experiments to ensure applicability to the standard permutation definition. Figure 3 depicts the distribution of the key-bit neutrality measure for $r = 3$ and $r = 3.5$ intermediate rounds. As shown in Figure 3, the distribution of the key-bit impact on the ciphertext remains consistent regardless of the number of rounds. This is attributed to the inverse quarter-round of the ChaCha stream cipher (see Equation (2)).

Algorithm 3 Experimental Verification of Related-Key Boomerang Probability

Input: The initial states (X_1, X_2) with an $\mathcal{ID}$ α and a keybit difference λ
Output: Verification of the boomerang distinguisher probability
Note: This algorithm describes the white-box verification process.

Initialize counters $\mathbf{ctr}_1 \leftarrow 0$, $\mathbf{ctr}_2 \leftarrow 0$ $\mathbf{ctr}_3 \leftarrow 0$, and $\mathbf{ctr}_4 \leftarrow 0$
for all pairs (X_1, X_2) **do**
    Initiate $X_1^{(0)}$ and $X_2^{(0)} = X_1^{(0)} \oplus \Delta_i^{(0)}[j]$.
    Encrypt X_1 with K_1 and X_2 with K_2 where $K_2 = K_1 \oplus \Delta K_0$ to obtain Y_1 and Y_2 using ChaCha E_0.
    if $Y_1 \oplus Y_2 = \beta$ **then**
        Increment $\mathbf{ctr}_1$
    end if
    Encrypt the Y_1 and Y_2 with ChaCha E_1 to get Z_1 and Z_2.
    Compute $Z_3 = Z_1 \oplus \delta \oplus \lambda_1$ and $Z_4 = Z_2 \oplus \delta \oplus \lambda_1 \oplus \lambda_2$
    Decrypt the Z_3 and Z_4 with ChaCha E_1^{-1}.
    if $Y_1 \oplus Y_3 = \alpha$, $Y_2 \oplus Y_4 = \alpha$, and $Y_3 \oplus Y_4 = \beta$. **then**
        Increment $\mathbf{ctr}_2$ $\mathbf{ctr}_3$, and $\mathbf{ctr}_4$, respectively.
    end if
end for
if ctr ≥ 1 **then**
    Return: Valid Distinguisher Found
else
    Return: No Distinguisher
end if

Algorithm 4 Computing $\mathcal{OD}$ bit with best neutral measure [29]

Require: Random(X, X', Z, Z')
Ensure: The $\mathcal{OD}$ bit with best neutral measure

1. Generate random keywords $k = (k_0, \dots, k_7)$.
2. Decide the single $\mathcal{ID}$ $\Delta_i^{(0)}[j]$ position, and generate random $v = (v_0, v_1)$ and $t = (t_0, t_1)$, initiate $X^{(0)}$ and $X'^{(0)} = X^{(0)} \oplus \Delta_i^{(0)}[j]$.
3. From $(X^{(0)}, X'^{(0)})$, compute $(X^{(r)}, X'^{(r)})$ and $(X^{(R)}, X'^{(R)})$.
4. From $(X^{(r)}, X'^{(r)})$ compute $\mathcal{OD}$ $\Delta_p^{(r)}[q] = X_p^{(r)}[q] \oplus X_p'^{(r)}[q]$ for all possible p and q.
5. From $(X^{(R)}, X'^{(R)})$ obtain the keystream $Z = X^{(0)} + X^{(R)}$ and $Z' = X'^{(0)} + X'^{(R)}$.
6. Complement a key bit κ ($\kappa \in \{0, \dots, 255\}$) and compute $\overline{X}^{(0)}$ and $\overline{X'}^{(0)}$ from initial states $(X^{(0)}, X'^{(0)})$.
7. Compute $(Y^{(r)}, Y'^{(r)})$ with $Z - \overline{X}^{(0)}$ and $Z' - \overline{X'}^{(0)}$ as inputs to the inversed round function of ChaCha.
8. Derive $\Gamma_p^{(r)}[q] = Y_p^{(r)}[q] \oplus Y_p'^{(r)}[q]$ for all possible choices of p and q.
9. Increment the sum for each p, q, and κ only if $\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q]$.
10. Divide the sum of $\Delta_p^{(r)}[q] = \Gamma_p^{(r)}[q]$ by key trail and $\mathcal{ID}$ samples to get the probability.

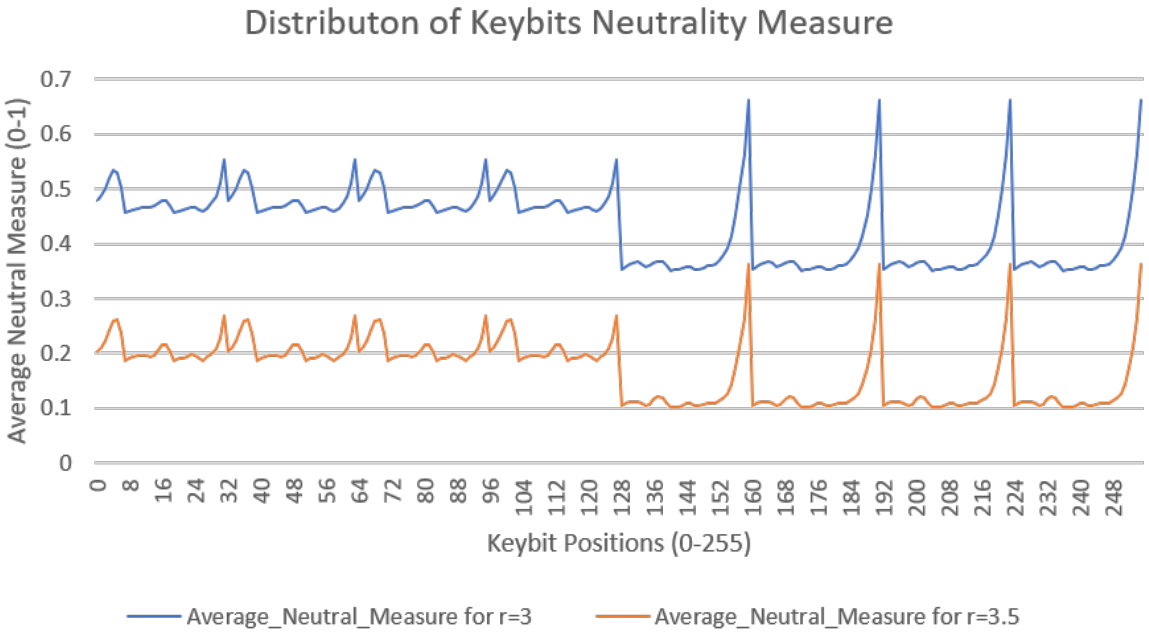

Figure 3. Distribution of keybits neutrality measure.

Table 3. The keybits neutrality measure.

Internal Rounds	Keybit Position	Average Neutral Measure
$r = 3$	159	0.662237799
	191	0.662253018
	223	0.662239797
	255	0.662243926
$r = 3.5$	159	0.362119217
	191	0.362123078
	223	0.362120682
	255	0.362121207

Table 4. The $\mathcal{OD}$ positions with highest average neutrality measure.

Internal Rounds	$\mathcal{OD}$	Average Neutral Measure
$r = 3$	$Y_4^{(3)}[0]$	0.648515617
	$Y_6^{(3)}[0]$	0.648516008
	$Y_7^{(3)}[0]$	0.648516234
$r = 3.5$	$Y_8^{(3.5)}[0]$	0.382019633
	$Y_9^{(3.5)}[0]$	0.382027766
	$Y_{10}^{(3.5)}[0]$	0.382022637
	$Y_{11}^{(3.5)}[0]$	0.382026773

4.2. Mounting the Attack

After completing 3.5 rounds, ChaCha becomes more resilient against conventional differential attacks, making it an attractive target for the boomerang attack and its variants. The differential properties of ChaCha's smaller components, comprising either 3 or 3.5 rounds, are analysed accordingly. In the case of ChaCha 6, it is divided into two sub-ciphers, each containing 3 rounds, and their respective differences are merged. Likewise, for ChaCha 7, it is partitioned into two sub-ciphers comprising 3.5 rounds each, and their respective differentials are combined to mount a distinguisher. However, splitting ChaCha's sub-ciphers beyond 3.5 rounds may lead to unreliable bias. Hence, a restriction is enforced on the subdivision of sub-ciphers, limiting it to a maximum of 3.5 sub-rounds to ensure accuracy and reliability of the analysis.

Algorithm 3 outlines the steps involved in the related-key boomerang attack. Initially, the attack employs $\alpha \to \beta$ as a difference in the initialisation vector (IV) and $\Delta K_9[31]$ or λ_1 as a key difference prior to encrypting (X_1, X_2) by E_0. Subsequently, it utilises $\gamma \to \delta$ along with $\lambda_2, \lambda_3, \lambda_4$ as differences in key-bits prior to decryption by E_1^{-1}. The detailed attack procedure is described as follows. Let $X_1^{(0)}$ and $X_2^{(0)}$ be two initial states with a single-bit difference in IV, such that $X_1^{(0)} \oplus X_2^{(0)} = \Delta X_{12}[0]$ and $X_1^{(0)} \oplus X_2^{(0)} = \Delta X_9[31]$ (i.e., key-bit 191) as differences in IV and key-bits. After encrypting $X_1^{(0)}$ and $X_2^{(0)}$ by the first half of the cipher E_0, two new states Y_1 and Y_2 are obtained, respectively. Using the differential characteristic of E_0, we have

$$Y_1 \oplus Y_2 = \beta \tag{5}$$

For Equation (5), the possible 512-bit $\mathcal{OD}$ positions are narrowed down based on the analysis presented in Section 4.1. For ChaCha 6, Equation (5) is examined for the $\mathcal{OD}$ positions $\Delta Y_4^{(3)}[0]$, $\Delta Y_6^{(3)}[0]$, and $\Delta Y_7^{(3)}[0]$. The corresponding probabilities are presented in Table 5. Similarly, for ChaCha 7, Equation (5) is assessed for the $\mathcal{OD}$ positions $\Delta Y_8^{(3.5)}[0]$, $\Delta Y_9^{(3.5)}[0]$, $\Delta Y_{10}^{(3.5)}[0]$, and $\Delta Y_{11}^{(3.5)}[0]$, with their corresponding probabilities shown in Table 6. The results produced by the E_0 cipher are retained to match the differentials with the outcomes from E_1^{-1}. For both ChaCha 6 and ChaCha 7, Equation (5) holds with probability p. The intermediate states $Y_i^{(r)}[j]$ are then encrypted by the E_1 sub-cipher to obtain Z_1 and Z_2. Subsequently, the new states to be input into E_1^{-1} are computed and the subsequent operations are performed.

$$Z_3 = Z_1 \oplus \delta \oplus \lambda_2 \tag{6}$$

$$Z_4 = Z_2 \oplus \delta \oplus \lambda_1 \oplus \lambda_2 \tag{7}$$

where $\delta = \Delta Z_{12}[0]$, representing the difference used to form the new states Z_3 and Z_4, and $\lambda_1 = \Delta Z_9[31]$ and $\lambda_2 = \Delta Z_{10}[31]$ are the related-key differences. Next, we input the sub-cipher Z_3 and Z_4 to E_1^{-1} to obtain Y_3 and Y_4. Equation (8) must be satisfied to compute the related-key boomerang probability.

$$(Y_1 \oplus Y_3 = \gamma) \wedge (Y_2 \oplus Y_4 = \gamma) \tag{8}$$

The Equation (8) occurs with a $\hat{q}^2$ probability. If both Equations (5) and (8) occur, it will result in $Y_3 \oplus Y_4 = \beta$. We can express this as:

$$Y_3 \oplus Y_4 = (Y_2 \oplus Y_4) \oplus (Y_1 \oplus Y_3) \oplus (Y_2 \oplus Y_1) = \gamma \oplus \gamma \oplus \beta = \beta \tag{9}$$

Therefore, according to the differential characteristic of E_0, we deduce that $X_3 \oplus X_4 = \alpha$ with a probability of $\hat{p}$. Assuming these events are independent (which we confirmed via experimental verification), we can thus state:

$$Pr[X_3 \oplus X_4 = \alpha \mid X_1^{(0)} \oplus X_2^{(0)} = \alpha] = \hat{p}^2\hat{q}^2. \tag{10}$$

To validate this independence assumption, we compared, over the 2^{40} trials described in Section 4.2, the empirically observed joint probability $Pr[X_3 \oplus X_4 = \alpha \wedge Y_1 \oplus Y_2 = \beta]$, obtained directly from the counters $\mathbf{ctr}_2$–$\mathbf{ctr}_4$ recorded by Algorithm 3, against the value predicted under independence, $\hat{p}^2\hat{q}^2$, computed from the individually measured biases in Tables 5 and 6.

Table 5. The related-key boomerang attack probability for ChaCha 6.

Intermediate $\mathcal{OD}$	$\hat{p_1^*}$	$\hat{p_2^*}$	$\hat{q_1^*}$	$\hat{q_2^*}$	$\hat{p^{*2}q^{*2}}$	Data Complexity
$\Delta X_4[0]^{(3)}$	0.49999	0.500001	0.984138	0.015862	0.0039029	$\approx 2^8$
$\Delta X_6[0]^{(3)}$	0.500001	0.500001	0.988814	0.988814	0.244439	$\approx 2^{2.03}$
$\Delta X_7[0]^{(3)}$	0.500001	0.5	0.50000	0.50000	0.0625002	$\approx 2^4$

Table 6. The related-key boomerang attack probability for ChaCha 7.

Intermediate $\mathcal{OD}$	$\hat{p_1^*}$	$\hat{p_2^*}$	$\hat{q_1^*}$	$\hat{q_2^*}$	$\hat{p^{*2}q^{*2}}$	Data Complexity
$\Delta X_8[0]^{(3.5)}$	0.500001	0.5	0.5	0.5	0.0625	$\approx 2^4$
$\Delta X_{10}[0]^{(3.5)}$	0.49999	0.49999	0.5	0.5000005	0.0624	$\approx 2^4$

On the Independence Assumption

The related-key boomerang probability $Pr[X_3 \oplus X_4 = \alpha] = \hat{p}^2\hat{q}^2$ used throughout this paper rests on one simplifying assumption: that the differential behaviour of the first sub-cipher E_0 and the differential behaviour of the second sub-cipher E_1 do not influence one another. In plain terms, once the intermediate states satisfy $Y_1 \oplus Y_2 = \beta$ with probability $\hat{p}$, we assume that whether E_1 (and its inverse) subsequently produces the required difference γ with probability $\hat{q}$ does not depend on how that particular β was reached inside E_0. The two halves behave like two separate coin flips rather than two linked ones. This is exactly the assumption underlying Wagner's original boomerang construction [6], where the distinguishing probability is likewise computed as $(pq)^2$ by treating the two halves of the cipher as independent, and it is the same assumption carried into the related-key boomerang framework of [10] that this paper builds upon.

4.3. Attack on ChaCha 7

To target ChaCha 7, two matrices $X_1^{(0)}$ and $X_2^{(0)}$ are initialised. A single-bit difference is introduced as $X_1^{(0)} \oplus X_2^{(0)} = \Delta X_{12}^{(0)}[0]$, along with a key-bit difference at $X_1^{(0)} \oplus X_2^{(0)} = \Delta X_9^{(0)}[31]$, which corresponds to the 191-st key-bit determined in Section 4.1. Subsequently, $X_1^{(0)}$ and $X_2^{(0)}$ are encrypted with E_0, yielding two intermediate states $Y_1^{(3.5)} = E_0(X_1^{(0)})$ and $Y_2^{(3.5)} = E_0(X_2^{(0)})$. The $\mathcal{OD}$ positions for $Y_1^{(3.5)}$ and $Y_2^{(3.5)}$ are selected based on the analysis presented in Section 4.1, namely $Y_8^{(3.5)}[0], Y_9^{(3.5)}[0], Y_{10}^{(3.5)}[0], Y_{11}^{(3.5)}[0]$ for ChaCha 7. The value $\hat{p}$ is computed for

$$Y_1^{(3.5)} \oplus Y_2^{(3.5)} = \Delta_{8,9,10,11}^{(3.5)}[0].$$

The results are presented in Table 6. According to Algorithm 3, the matrices generated by E_0 are preserved prior to being passed to E_1. These results from E_0 are subsequently utilised to compute the differences with the outcomes of E_1^{-1}, thereby yielding new probabilities. The resulting matrices are then encrypted with E_1 to obtain $Z_1^{(7)} = E_1(Y_1^{(3.5)})$ and $Z_2^{(7)} = E_1(Y_2^{(3.5)})$. The steps described in Section 4.2 are followed to obtain the new matrices for the E_1^{-1} cipher.

$$Z_3^{(7)} = Z_1^{(7)} \oplus \Delta X_{12,[0]}^{(R)} \oplus \Delta X_{10,[31]}^{(R)} \tag{11}$$

$$Z_4^{(7)} = Z_2^{(7)} \oplus \Delta X_{12,[0]}^{(R)} \oplus \Delta X_{10,[31]}^{(R)} \oplus \Delta X_{9,[31]}^{(R)} \tag{12}$$

The key-bit difference positions are determined by the analysis presented in Section 4.1. Consequently, the newly generated states $Z_3^{(7)}$ and $Z_4^{(7)}$ are input into the inverse round E_1^{-1}.

$$Y_3^{(3.5)} = E_1^{-1}(Z_3^{(7)}) \tag{13}$$

$$Y_4^{(3.5)} = E_1^{-1}(Z_4^{(7)}) \tag{14}$$

At this point, to construct the distinguisher, the difference of the following states is computed. As per Equation (8), the difference of the following states is computed:

$$Y_1^{(3.5)} \oplus Y_3^{(3.5)} = \gamma \tag{15}$$

$$Y_2^{(3.5)} \oplus Y_4^{(3.5)} = \gamma \tag{16}$$

The probabilities of $Y_1^{(3.5)} \oplus Y_3^{(3.5)}$ and $Y_2^{(3.5)} \oplus Y_4^{(3.5)}$ are denoted by q_1 and q_2, respectively. Given Equations (9), (15) and (16) impose the following difference on $Y_3^{(3.5)}$ and $Y_4^{(3.5)}$.

$$Y_3^{(3.5)} \oplus Y_4^{(3.5)} = \beta \tag{17}$$

The differences for Equation (17) over 3.5 rounds are computed for the $\mathcal{OD}$ positions $\Delta_{8,9,10,11}^{(3.5)}[0]$. To determine the probabilities, Algorithm 3 is executed with a complexity of 2^{40} (for each of 2^{10} key trials, 2^{30} IVs are evaluated). The results are presented in Table 6. The data complexity of the related-key boomerang distinguisher for ChaCha 7, 2^4, was experimentally verified.

A maximum-length random number (m-sequence) generator is used to generate random keys and IVs. The experiments are conducted on an Intel(R) Xeon(R) CPU E7-4830 v4 @ 2.00GHz machine, with a total complexity of 2^{40} (i.e., for each random key, 2^{30} IVs are evaluated). For consistency, the hexadecimal value 0xAFFFFFFF is used as the seed for IV generation and 0xAEEEEEEE as the seed for key generation. No notable impact on the

results is observed despite varying the seeds across the same attack scenarios.

Security Bound Analysis

The observed probability for the related-key boomerang return for ChaCha 7 is ≈ 0.0625 (Table 6). Since a 4-bit truncated differential probability is monitored, the probability of a random permutation satisfying this condition is exactly $0.5^4 = 0.0625$. Consequently, the observed probability is indistinguishable from that of a random permutation. This indicates that, while the attack successfully distinguishes 6 rounds (as demonstrated in the following section), ChaCha 7 provides sufficient diffusion to resist these specific distinguisher attacks. It should be noted that this security bound is a conditional empirical bound rather than a formal security proof. It holds exclusively under the specific $\mathcal{ID}$, $\mathcal{OD}$, and key-bit configurations examined in this work, and does not preclude the existence of other configurations that may yield a distinguishable bias against the ChaCha 7 permutation.

4.4. Attack on ChaCha 6

To conduct a related-key distinguisher attack on ChaCha 6, the same steps as for ChaCha 7 are followed, focusing on the 3-round intermediate state. Algorithm 3 is implemented with a total of 2^{40} trials for the positions $\Delta X_4[0]^{(3)}$, $\Delta X_5[0]^{(3)}$, $\Delta X_6[0]^{(3)}$, and $\Delta X_7[0]^{(3)}$. It is found that the position $\Delta X_5[0]^{(3)}$ yields a probability of 0 in E^{-1}; consequently, it is excluded from the computation. The data complexity of the related-key boomerang distinguisher for ChaCha 6, $\approx 2^{2.03}$ ($\approx$4 pairs), is experimentally verified with a high success probability.

5. Conclusions and Future Work

In this paper, the security of the ChaCha permutation function against the related-key boomerang attack is studied. An attack strategy for selecting key-bit and output differential positions is defined, and an attack on ChaCha 6 is mounted. As a result, a total of $\approx 2^{2.03}$ adaptively chosen plaintext and ciphertext pairs are required to distinguish the ChaCha permutation from a random permutation. This work introduces an adaptively chosen plaintext and ciphertext attack. Further work is required to extend the attack to chosen plaintext settings on the ChaCha stream cipher rather than the ChaCha permutation function, and to further investigate variants of the boomerang attack on ChaCha.

Author Contributions

N.G.: Conceptualization, methodology, investigation, formal analysis, writing—original draft preparation, writing—reviewing and editing; A.M.: Conceptualization, supervision, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI, Grant Number 26H02496.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

Grammarly (AI-powered writing assistant) was used for text editing, and the text was reviewed and corrected by the authors. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Notation

Notation	Description
$X^{(0)}$	The initial state matrix of ChaCha with 16 words of 32 bits each
$X^{'(0)}$	The associated matrix with a single bit difference.
$X^{(r,R)}$	The matrix after ChaCha r, R rounds where $R > r$
$x_i^{(r,R)}[j]$	The j^{th} bit of i^{th} word of matrix $X^{(r,R)}$
$x + y$	The word-wise addition of word x and y modulo 2^{32}
$\mathcal{ID}, \mathcal{OD}$	Input and output difference
$x \oplus y$	Bit-wise XOR operation of the word x and y
$x \lll n$	The left rotation of word x by n bits
Δx	The XOR difference of word x and $x^{'}$ defined as $\Delta x = x \bigoplus x^{'}$
ε_d	The forward bias of ChaCha
ctr	Differential condition occurrence counter
ACPCT	Adaptive Chosen Plaintext Ciphertext
CP	Chosen Plaintext
KR	Key Recovery Attack
Stream	Attack on the ChaCha Stream Cipher Mode
Perm	Attack on the ChaCha Permutation Mode (This Work)

References

1. Biham, E.; Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72.
2. National Bureau of Standards. *Federal Information Processing Standards Publication (FIPS PUB) 46*; Data Encryption Standard (DES); U.S. Department of Commerce: Washington, DC, USA, 1977.
3. Knudsen, L.R. Truncated and Higher Order Differentials. In Proceedings of the Second International Workshop, Leuven, Belgium, 14–16 December 1994; Preneel, B., Ed.; Springer: Berlin, Heidelberg, Germany, 1995; pp. 196–211. https://doi.org/10.1007/3-540-60590-8_16.
4. Biham, E.; Biryukov, A.; Shamir, A. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. *J. Cryptol.* **2005**, *18*, 291–311. https://doi.org/10.1007/s00145-005-0129-3.
5. Biham, E. New Types of Cryptanalytic Attacks Using Related Keys. *J. Cryptol.* **1994**, *7*, 229–246. https://doi.org/10.1007/BF00203965.
6. Wagner, D. The Boomerang Attack. In Proceedings of the 6th International Workshop, FSE'99, Rome, Italy, 24–26 March 1999; Knudsen, L., Ed.; Springer: Berlin, Heidelberg, Germany, 1999; pp. 156–170. https://doi.org/10.1007/3-540-48519-8_12.
7. Biham, E.; Dunkelman, O.; Keller, N. Related-Key Boomerang and Rectangle Attacks. In Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Cramer, R., Ed.; Springer: Berlin, Heidelberg, Germany, 2005; pp. 507–525. https://doi.org/10.1007/11426639_30.
8. Dunkelman, O.; Keller, N.; Ronen, E.; et al. The Retracing Boomerang Attack. In Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 10–14 May 2020; Canteaut, A., Ishai, Y., Eds.; Springer: Cham, Switzerland, 2020; pp. 280–309. https://doi.org/10.1007/978-3-030-45721-1_11.
9. Kelsey, J.; Kohno, T.; Schneier, B. Amplified Boomerang Attacks against Reduced-Round MARS and Serpent. In Proceedings of the 7th International Workshop, FSE 2000, New York, NY, USA, 10–12 April 2000; Schneier, B., Ed.; Springer: Berlin, Heidelberg, Germany, 2001; pp. 75–93. https://doi.org/10.1007/3-540-44706-7_6.
10. Kim, J.; Hong, S.; Preneel, B.; et al. Related-Key Boomerang and Rectangle Attacks: Theory and Experimental Analysis. *IEEE Trans. Inf. Theory* **2012**, *58*, 4948–4966. https://doi.org/10.1109/TIT.2012.2191655.
11. Jakimoski, G.; Desmedt, Y. Related-Key Differential Cryptanalysis of 192-bit Key AES Variants. In Proceedings of the 10th Annual International Workshop, SAC 2003, Ottawa, Canada, 14–15 August 2003; Matsui, M., Zuccherato, R.J., Eds.; Springer: Berlin, Heidelberg, Germany, 2004; pp. 227–237. https://doi.org/10.1007/978-3-540-24654-1_15.
12. Biham, E.; Dunkelman, O.; Keller, N. New Cryptanalytic Results on IDEA. In Proceedings of the 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, 3–7 December 2006; Lai, X., Chen, K., Eds.; Springer: Berlin, Heidelberg, Germany, 2006; pp. 412–427. https://doi.org/10.1007/11935230_27.
13. Bernstein, D.J. ChaCha, a Variant of Salsa20. In *Workshop Record of SASC*; Springer: Heidelberg, Germany, 2008; Vol. 8, pp. 3–5.
14. Vaudenay, S. Provable Security for Block Ciphers by Decorrelation. In Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science, Paris, France, 25–27 February 1998; Morvan, M., Meinel, C., Krob, D., Eds.; Springer: Berlin, Heidelberg, Germany, 1998; pp. 249–275. https://doi.org/10.1007/BFb0028566.

15. Choudhuri, A.R.; Maitra, S. Significantly Improved Multi-bit Differentials for Reduced Round Salsa and ChaCha. *IACR Trans. Symmetric Cryptol.* **2017**, *2016*, 261–287. https://doi.org/10.13154/tosc.v2016.i2.261-287.
16. Coutinho, M.; Souza Neto, T.C. Improved Linear Approximations to ARX Ciphers and Attacks against ChaCha. In Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 17–21 October 2021; Canteaut, A., Standaert, F.X., Eds.; Springer: Cham, Switzerland, 2021; pp. 711–740. https://doi.org/10.1007/978-3-030-77870-5_25.
17. Aumasson, J.P.; Fischer, S.; Khazaei, S.; et al. New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba. In Proceedings of the 15th International Workshop, FSE 2008, Lausanne, Switzerland, 10–13 February 2008; Nyberg, K., Ed.; Springer: Berlin, Heidelberg, Germany, 2008; pp. 470–488. https://doi.org/10.1007/978-3-540-71039-4_30.
18. Beierle, C.; Broll, M.; Canale, F.; et al. Improved Differential-Linear Attacks with Applications to ARX Ciphers. *J. Cryptol.* **2022**, *35*, 29. https://doi.org/10.1007/s00145-022-09437-z.
19. Shi, Z.; Zhang, B.; Feng, D.; et al. Improved Key Recovery Attacks on Reduced-Round Salsa20 and ChaCha. In Proceedings of the 15th International Conference, ICISC 2012, Seoul, Korea, 28–30 November 2012; Kwon, T., Lee, M.K., Kwon, D., Eds.; Springer: Berlin, Heidelberg, Germany, 2013; pp. 369–380. https://doi.org/10.1007/978-3-642-37682-5_24.
20. Ghafoori, N.; Miyaji, A. The Amplified Boomerang Attack on ChaCha. In Proceedings of the 8th International Conference, MobiSec 2024, Sapporo, Japan, 17–19 December 2024; Jo, H., Shin, S., Merlo, A., Eds.; Springer: Singapore, 2026; pp. 118–132. https://doi.org/10.1007/978-981-95-0172-4_8.
21. Coutinho, M.; Passos, I.; Vásquez, J.C.G.; et al. Latin Dances Reloaded: Improved Cryptanalysis Against Salsa and ChaCha, and the Proposal of Forró. *J. Cryptol.* **2023**, *36*, 18. https://doi.org/10.1007/s00145-023-09455-5.
22. Dey, C.; Sarkar, S. A New Distinguishing Attack on Reduced Round ChaCha Permutation. *Sci. Rep.* **2023**, *13*, 13958. https://doi.org/10.1038/s41598-023-39849-1.
23. Bellini, E.; Gerault, D.; Grados, J.; et al. Boosting Differential-Linear Cryptanalysis of ChaCha7 with MILP. *IACR Trans. Symmetric Cryptol.* **2023**, *2023*, 189–223. https://doi.org/10.46586/tosc.v2023.i2.189-223.
24. Watanabe, R.; Ghafoori, N.; Miyaji, A. Improved Differential-Linear Cryptanalysis of Reduced Rounds of ChaCha. In Proceedings of the 24th International Conference, WISA 2023, Jeju Island, South Korea, 23–25 August 2023; Kim, H., Youn, J., Eds.; Springer: Singapore, 2024; pp. 269–281. https://doi.org/10.1007/978-981-99-8024-6_21.
25. Okada, Y.; Watanabe, R.; Ghafoori, N.; et al. Improved Differential-Linear Cryptanalysis of Reduced Rounds of ChaCha Permutation. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2025**, *E108.A*, 1175–1195. https://doi.org/10.1587/transfun.2024DMP0008.
26. Dey, S.; Garai, H.K.; Sarkar, S.; et al. Revamped Differential-Linear Cryptanalysis on Reduced Round ChaCha. In Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, 30 May–3 June 2022; Dunkelman, O., Dziembowski, S., Eds.; Springer: Cham, Switzerland, 2022; pp. 86–114. https://doi.org/10.1007/978-3-031-07082-2_4.
27. Miyashita, S.; Ito, R.; Miyaji, A. PNB-Focused Differential Cryptanalysis of ChaCha Stream Cipher. In Proceedings of the 27th Australasian Conference, ACISP 2022, Wollongong, NSW, Australia, 28–30 November 2022; Nguyen, K., Yang, G., Guo, F., et al., Eds.; Springer: Cham, Switzerland, 2022; pp. 46–66. https://doi.org/10.1007/978-3-031-22301-3_3.
28. Wang, S.; Liu, M.; Hou, S.; et al. Moving a Step of ChaCha in Syncopated Rhythm. In Proceedings of the 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, 20–24 August 2023; Handschuh, H., Lysyanskaya, A., Eds.; Springer: Cham, Switzerland, 2023; pp. 273–304. https://doi.org/10.1007/978-3-031-38548-3_10.
29. Ghafoori, N.; Miyaji, A.; Ito, R.; et al. PNB Based Differential Cryptanalysis of Salsa20 and ChaCha. *IEICE Trans. Inf. Syst.* **2023**, *106*, 1407–1422. https://doi.org/10.1587/transinf.2022ICP0015.
30. Ghafoori, N.; Miyaji, A. Higher-Order Differential-Linear Cryptanalysis of ChaCha Stream Cipher. *IEEE Access* **2024**, *12*, 13386–13399. https://doi.org/10.1109/ACCESS.2024.3356868.
31. Ghafoori, N. A Study on Differential Cryptanalysis of Salsa20 and ChaCha Stream Ciphers. Ph.D. Thesis, Osaka University, Suita, Osaka, Japan, 2024. https://doi.org/10.18910/101457.
32. Aumasson, J.P. Too Much Crypto. *Cryptology ePrint Archive* **2019**, Paper 2019/1492.
33. Isobe, T.; Sasaki, Y.; Chen, J. Related-key Boomerang Attacks on KATAN32/48/64. In Proceedings of the 18th Australasian Conference, ACISP 2013, Brisbane, Australia, 1–3 July 2013; Boyd, C., Simpson, L., Eds.; Springer: Berlin, Heidelberg, Germany, 2013; pp. 268–285.