# Revisiting the Differential-Linear Attacks on ChaCha from IEEE TIT and INDOCRYPT 2024 (Extended Abstract)

Xinhai Wang, Lin Ding[†], Zhengting Li, Jiang Wan, Bin Hu

Information Engineering University, Zhengzhou 450001, China
dinglin_cipher@163.com

**Abstract.** The ChaCha stream cipher has become one of the best known ARX-based ciphers because of its widely use in several systems, such as in TLS, SSH and so on. In this paper, we find some errors in the attacks on ChaCha256 from IEEE TIT and INDOCRYPT 2024, and then corrected cryptanalytic attacks on ChaCha256 are given. However, the corrected attacks have extremely large time and data complexities. The corrected results show that the technique proposed in IEEE TIT may not be able to obtain improved differential-linear attacks on ChaCha.

**Keywords:** Cryptanalysis; differential-linear attack; ChaCha; stream cipher.

## 1 Introduction

ChaCha [1] is an ARX-based stream cipher designed by Bernstein in 2008. It has become one of the best known ARX-based ciphers because of its widely use in several systems, such as in TLS, SSH and so on. The cipher consists of 20 rounds totally, and has a 256-bit key version called ChaCha256 and a 128-bit key version called ChaCha128. Since ChaCha was proposed, there have been many cryptanalytic attacks on it. In IEEE TIT, Dey [2] proposed improved differential-linear attacks on reduced rounds of ChaCha. Recently, based on the approach proposed in IEEE TIT, new attacks on 7-, 7.25- and 7.5-round ChaCha256 were published at INDOCRYPT 2024 [3]. In this paper, we revisit the attacks on ChaCha256 from IEEE TIT and INDOCRYPT 2024. Unfortunately, we find some errors in all these attacks, and then give corrected cryptanalytic attacks on ChaCha256, as listed in Table 1. As shown in Table 1, the corrected attacks have extremely large time and data complexities, compared with the claimed attacks in [2, 3]. The corrected results show that the technique proposed in [2] may not be able to obtain improved differential-linear attacks on ChaCha.

## 2 A Brief Review of the Approach Proposed in IEEE TIT

IEEE TIT, Dey [2] proposed a divide-and-conquer approach to improve the existing PNB-based key recovery attack. Unlike previous approaches that focus on finding PNBs for the entire linear combination of multiple bits of

**Table 1.** Corrected complexities of the attacks on ChaCha256 from IEEE TIT and INDOCRYPT 2024

| Rounds | Time | Data | Memory | Attack | Correctness |
|---|---|---|---|---|---|
| 7 | $2^{192.89}$ | $2^{93.79}$ | $2^{68}$ | [2] | **No** |
| | $2^{178.12}$ | $2^{101.09}$ | – | [3] | **No** |
| | $2^{217.48}$ | $2^{127.45}$ | $2^{155.45}$ | **This paper** | **Yes** |
| 7.25 | $2^{228.24}$ | $2^{100.9}$ | $2^{91}$ | [2] | **No** |
| | $2^{212.43}$ | $2^{100.56}$ | – | [3] | **No** |
| | $2^{260.18}$ | $2^{135.19}$ | $2^{178.19}$ | **This paper** | **Yes** |
| 7.5 | $2^{255.24}$ | $2^{32.64}$ | $2^{217}$ | [2] | **No** |
| | $2^{253.23}$ | $2^{34.47}$ | – | [3] | **No** |
| | $2^{338.55}$ | $2^{117.36}$ | $2^{301.36}$ | **This paper** | **Yes** |

output difference, the new approach separately identifies PNBs for each output difference bit that constitutes the linear combination. Let us consider a PNB-based key recovery attack on ChaCha, which utilizes a linear combination $\mathcal{OD} = \mathcal{OD}_1 \oplus \mathcal{OD}_2 \oplus \cdots \oplus \mathcal{OD}_k$ involving $k\,(>1)$ output difference bits $\mathcal{OD}_1, \mathcal{OD}_2, \cdots, \mathcal{OD}_k$. The new approach is based on an observation that if, for some threshold $\theta$, $\kappa$ is a PNB for the linear combination $\mathcal{OD}$, then for the same threshold $\theta$, $\kappa$ is a PNB for each output difference bit $\mathcal{OD}_i (i \in \{1, 2, \ldots, k\})$. Let $PNB_{\mathcal{OD}}$ denotes the set of PNBs for the linear combination $\mathcal{OD}$, and $S$ denotes the set of remaining key bits, i.e., $S = PNB^c_{\mathcal{OD}}$. Denote by $PNB_{\mathcal{OD}_i}$ the set of PNBs for the output difference bit $\mathcal{OD}_i$. By Theorem 1 of [2], it has $PNB_{\mathcal{OD}} \subset PNB_{\mathcal{OD}_i}$ and then $PNB_{\mathcal{OD}_i} = PNB_{\mathcal{OD}} \bigcup PNB'_{\mathcal{OD}_i}$, where $PNB'_{\mathcal{OD}_i}$ is the set of the key bits that belong to $PNB_{\mathcal{OD}_i}$ and do not belong to $PNB$. Let $S_i = PNB^c_{\mathcal{OD}_i}$, and obviously it has $S_i \subset S$ and $\bigcup_{i=1}^{k} S_i = S$.

The PNB-based key recovery attack of [2] consists of two phases, i.e., the preprocessing phase and the online phase. In the preprocessing phase, the attacker has to find the primary PNB set for the linear combination $\mathcal{OD}$ as in the existing PNB-based attacks. Meanwhile, the attacker should find the PNB set $PNB_{\mathcal{OD}_i}$ for each output difference bit $\mathcal{OD}_i$, $i = 1, 2, \ldots, k$. In the online phase, the attacker should at first collect $N$ output keystream pairs $(Z, Z')$. For each possible guess of the significant key bits $S_i$, the output differential should be computed for each of $N$ output keystream pairs $(Z, Z')$, and then a tuple consisting of $N$ bits is obtained and stored. Finally, for each guessed value of the significant key bits $S$, the attacker verifies the correctness of the guessed significant key bits using the prepared tuples. By the proposed divide-and-conquer approach, Dey proposed improved key recovery attacks on 7- and 7.25-round ChaCha256, and gave the first-ever attack on 7.5-round ChaCha256. Recently, based on the approach proposed in IEEE TIT, new attacks on 7-, 7.25- and 7.5-round ChaCha256 were published at INDOCRYPT 2024 [3].

## 3 The Errors in the Attacks from IEEE TIT and INDOCRYPT 2024

In this paper, we will revisit the attacks on ChaCha from IEEE TIT and INDOCRYPT 2024, and point out the errors in them. The key problem that leads to the errors is that the attacker utilizes different thresholds for the linear combination $\mathcal{OD}$ and the output difference bit $\mathcal{OD}_i$. As shown above, for some threshold $\theta$, $\kappa$ is a PNB for the linear combination $\mathcal{OD}$, then for the same threshold $\theta$, $\kappa$ is a PNB for each output difference bit $\mathcal{OD}_i (i \in \{1, 2, \ldots, k\})$. However, if $\kappa$ is a PNB for the linear combination $\mathcal{OD}$ and some threshold $\theta$, for a larger threshold $\theta' (> \theta)$, $\kappa$ is not necessarily a PNB for the output difference bit $\mathcal{OD}_i$. Now, we take the attack on 7-round ChaCha256 as example and point out the errors existing in the attack from IEEE TIT.

In the proposed attack on 7-round ChaCha256, the author of [2] used the 5-round distinguisher given in [4]. The input difference consists of two active bits $\Delta x^{(0)}_{15,7}$ and $\Delta x^{(0)}_{15,19}$, and the output difference at round 5 is a linear combination of 5 active bits, i.e., $\mathcal{OD} = \Delta x^{(5)}_{2,0} \oplus \Delta x^{(5)}_{6,7} \oplus \Delta x^{(5)}_{6,19} \oplus \Delta x^{(5)}_{10,12} \oplus \Delta x^{(5)}_{14,0}$. For the linear combination $\mathcal{OD}$, by assigning a threshold of 0.435, they had found 156 PNBs as follows:

$PNB_{\mathcal{OD}}$ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 19, 20, 21, 31, 32, 33, 34, 35, 36, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 71, 72, 73, 74, 77, 78, 79, 80, 83, 84, 85, 86, 89, 90, 91, 95, 99, 100, 103, 104, 105, 106, 107, 108, 109, 110, 123, 124, 125, 126, 127, 128, 129, 130, 140, 141, 142, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 191, 192, 193, 194, 198, 199, 200, 201, 204, 205, 206, 207, 210, 211, 212, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 231, 244, 245, 246, 247, 255}

For the output difference bit $\mathcal{OD}_1 = \Delta x^{(5)}_{2,0}$, they had obtained 156+48 = 204 PNBs, and they did not give the threshold to obtain these 204 PNBs. We checked this experimentally, and found that the threshold should be assigned to 0.938 to obtain 204 PNBs. The set of obtained 204 PNBs is listed as follows:

$PNB_{\mathcal{OD}_1}$ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 71, 72, 73, 74, 75, 76, 77, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 231, 232, 233, 234, 235, 236, 243, 244, 245, 246, 247, 248, 249, 255}

By a simple comparison, we find that there are 17 PNBs, i.e., {33, 34, 35, 36, 66, 67, 68, 78, 79, 80, 99, 100, 199, 200, 201, 226, 227}. Each of these 17 PNBs belongs to $PNB_{\mathcal{OD}}$, but does not belong to $PNB_{\mathcal{OD}_1}$. Clearly, it is contradictory with the observation $PNB_{\mathcal{OD}} \subset PNB_{\mathcal{OD}_i}$ of [2]. The key reason is that the

threshold 0.938 for $\mathcal{OD}_1$ is larger than the threshold 0.435 for $\mathcal{OD}$, and the key bit is a PNB for $\mathcal{OD}$ under the threshold 0.435 is not necessarily a PNB for $\mathcal{OD}_1$ under the threshold 0.938.

Similarly, for the output difference bit $\mathcal{OD}_2 = \Delta x_{6,7}^{(5)}$, they had obtained $156 + 68 = 224$ PNBs, and they did not give the threshold to obtain these 224 PNBs. We checked this experimentally, and found that the threshold should be assigned to 0.7655 to obtain 224 PNBs. The set of obtained 224 PNBs is listed as follows:

$PNB_{\mathcal{OD}_2}$ = {0, 1, 2, 3, 4, 5, 6, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 231, 232, 233, 234, 235, 236, 237, 238, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255}

By a simple comparison, we find that there are 9 PNBs, i.e., {7, 8, 9, 73, 74, 128, 129, 130, 194}. Each of these 9 PNBs belongs to $PNB_{\mathcal{OD}}$, but does not belong to $PNB_{\mathcal{OD}_2}$.

Similarly, for the output difference bit $\mathcal{OD}_3 = \Delta x_{6,19}^{(5)}$, they had obtained $156 + 48 = 204$ PNBs, and they did not give the threshold to obtain these 204 PNBs. We checked this experimentally, and found that the threshold should be assigned to 0.965 to obtain 204 PNBs. The set of obtained 204 PNBs is listed as follows:

$PNB_{\mathcal{OD}_3}$ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 198, 199, 200, 201, 202, 203, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 243, 244, 245, 246, 247, 248, 255}

By a simple comparison, we find that there are 17 PNBs, i.e., {19, 20, 21, 83, 84, 85, 86, 140, 141, 142, 192, 193, 194, 204, 205, 206, 207}. Each of these 17 PNBs belongs to $PNB_{\mathcal{OD}}$, but does not belong to $PNB_{\mathcal{OD}_3}$.

Similarly, for the output difference bit $\mathcal{OD}_4 = \Delta x_{10,12}^{(5)}$, they had obtained $156 + 32 = 188$ PNBs, and they did not give the threshold to obtain these 188 PNBs. We checked this experimentally, and found that the threshold should be assigned to 0.84 to obtain 188 PNBs. The set of obtained 188 PNBs is listed as follows:

$PNB_{\mathcal{OD}_4}$ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19, 20, 21, 22, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 123, 124, 125, 126, 127, 128, 129, 130, 131, 140, 141, 142, 143, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255}

By a simple comparison, we find that there are 8 PNBs, i.e., {14, 107, 108, 109, 110, 200, 201, 231}. Each of these 8 PNBs belongs to $PNB_{\mathcal{OD}}$, but does not belong to $PNB_{\mathcal{OD}_4}$.

Similarly, for the output difference bit $\mathcal{OD}_5 = \Delta x^{(5)}_{14,0}$, they had obtained $156 + 41 = 197$ PNBs, and they did not give the threshold to obtain these 197 PNBs. We checked this experimentally, and found that the threshold should be assigned to 0.8785 to obtain 197 PNBs. The set of obtained 197 PNBs is listed as follows:

$PNB_{\mathcal{OD}_5}$ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 95, 103, 104, 105, 106, 107, 108, 109, 110, 119, 120, 121, 122, 123, 124, 125, 128, 129, 135, 136, 137, 138, 139, 140, 141, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 240, 241, 242, 243, 244, 245, 252, 253, 254, 255}

By a simple comparison, we find that there are 15 PNBs, i.e., {89, 90, 91, 99, 100, 126, 127, 130, 142, 210, 211, 212, 231, 246, 247}. Each of these 15 PNBs belongs to $PNB_{\mathcal{OD}}$, but does not belong to $PNB_{\mathcal{OD}_5}$.

Since the whole attack is based on the observation that $PNB_{\mathcal{OD}} \subset PNB_{\mathcal{OD}_i}$ holds for $i = 1, \ldots, 5$, it naturally becomes invalid when the observation does not hold in this attack. Thus, the proposed attack on 7-round ChaCha256 of [2] is incorrect. Similarly, the proposed attacks on 7.25- and 7.5-round ChaCha256 of [2] are also incorrect due to the same errors. In addition, since the new attacks on ChaCha from INDOCRYPT 2024 [3] utilized the proposed approach of [2], they are also incorrect.

## 4 Corrected Attacks on ChaCha256

In this section, we will give corrected attacks on 7-, 7.25- and 7.5-round ChaCha256, respectively. Unlike the uncorrected attacks in [2, 3], our corrected attacks will use the same threshold to find PNBs for both the linear combination and the output difference bits.

In the attack on 7-round ChaCha256, we use the set of 156 PNBs $PNB_{\mathcal{OD}}$ as the same in [2], by assigning a threshold of 0.435. By assigning suitable values at the PNBs, we achieve a backward correlation $\varepsilon_a = 0.0026$. For the five output

difference bits $\mathcal{OD}_1, \cdots, \mathcal{OD}_5$, we find 236, 238, 238, 221 and 225 PNBs, respectively, by assigning the same threshold of 0.435. The backward correlations of these five PNB sets are 0.147, 0.229, 0.306, 0.123 and 0.065, respectively. The PNB sets used in the attack, i.e., $PNB'_{(2,0)}$, $PNB'_{(6,7)}$, $PNB'_{(6,19)}$, $PNB'_{(10,12)}$ and $PNB'_{(14,0)}$, are listed in the following table. Considering the forward correlation $\varepsilon_d = 2^{-34.56}$, and then we have

$$\varepsilon = 2^{-34.56} \times 0.0026 \times 0.147 \times 0.229 \times 0.306 \times 0.123 \times 0.065 \approx 2^{-56.71}$$

As in previous works, we consider the non-detection error probability $\Pr_{nd} = 1.3 \times 10^{-3}$, and then $\Phi^{-1}[\Pr_{nd}] = -3$. For $\alpha = 51$, we can get $N = 2^{120.45}$. Since the attack process has to be repeated $2^7$ times, the final time and data complexities of this attack are $2^{210.48} \times 2^7 = 2^{217.48}$ and $2^{120.45} \times 2^7 = 2^{127.45}$. As shown in Table 3, the memory complexity of the attack on 7-round ChaCha256 is about $2^{155.45}$.

Table 2: The PNB sets used in the attack on 7-round ChaCha256

| **Set** | **Key bits** | **Count** | **Correlation** |
|---|---|---|---|
| $PNB'_{(2,0)}$ | {10, 11, 12, 13, 15, 16, 17, 18, 22, 23, 24, 25, 26, 27, 28, 29, 30, 75, 76, 87, 88, 92, 93, 94, 96, 97, 98, 100, 102, 111, 112, 113, 114, 115, 116, 117, 118, 119, 122, 131, 132, 133, 134, 135, 136, 137, 143, 144, 145, 146, 147, 148, 149, 150, 151, 190, 195, 196, 197, 208, 209, 213, 214, 215, 216, 217, 228, 232, 233, 234, 235, 236, 237, 238, 239, 243, 248, 249, 250, 251, 252} | 80 | 0.147 |
| $PNB'_{(6,7)}$ | {15, 16, 17, 18, 22, 23, 24, 25, 26, 27, 28, 29, 30, 37, 38, 69, 70, 81, 82, 87, 88, 92, 93, 94, 96, 97, 98, 101, 102, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 135, 136, 137, 138, 139, 143, 144, 145, 146, 147, 148, 149, 150, 151, 190, 195, 202, 203, 208, 209, 213, 214, 215, 228, 232, 233, 234, 235, 236, 237, 238, 239, 240, 243, 248, 249, 250, 251, 252, 253, 254} | 82 | 0.229 |

| | | | |
|---|---|---|---|
| $PNB'_{(6,19)}$ | {10, 11, 12, 13, 15, 16, 17, 18, 26, 27, 28, 29, 30, 37, 38, 69, 70, 75, 76, 81, 82, 92, 93, 94, 96, 97, 98, 101, 102, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 131, 132, 133, 134, 135, 136, 137, 138, 139, 147, 148, 149, 150, 151, 190, 195, 202, 203, 213, 214, 215, 216, 217, 228, 229, 230, 232, 233, 234, 235, 236, 237, 238, 239, 240, 243, 248, 249, 250, 251, 252} | 82 | 0.306 |
| $PNB'_{(10,12)}$ | {10, 11, 12, 13, 22, 23, 24, 25, 26, 37, 38, 69, 70, 75, 76, 81, 82, 87, 88, 92, 93, 94, 96, 97, 98, 101, 102, 115, 116, 117, 118, 131, 132, 133, 134, 143, 144, 145, 146, 147, 190, 195, 196, 197, 208, 209, 213, 214, 215, 216, 217, 228, 229, 230, 236, 237, 238, 239, 248, 249, 250, 251, 252, 253, 254} | 65 | 0.123 |
| $PNB'_{(14,0)}$ | {10, 11, 12, 13, 15, 16, 17, 18, 22, 23, 26, 27, 28, 29, 30, 37, 38, 69, 70, 75, 76, 81, 82, 87, 88, 111, 112, 115, 119, 120, 121, 122, 131, 132, 135, 136, 137, 138, 139, 143, 144, 147, 148, 149, 150, 151, 190, 195, 196, 197, 202, 203, 208, 209, 216, 217, 228, 229, 230, 232, 233, 240, 241, 242, 243, 248, 252, 253, 254} | 69 | 0.065 |

**Table 3.** The complexity of preparing the list for 7-round ChaCha256

| $\mathcal{OD}$ | **List size** | **Complexity** |
|---|---|---|
| $(2,0)$ | $2^{20}$ | $2^{140.45}$ |
| $(6,7)$ | $2^{18}$ | $2^{138.45}$ |
| $(6,19)$ | $2^{18}$ | $2^{138.45}$ |
| $(10,12)$ | $2^{35}$ | $2^{155.45}$ |
| $(14,0)$ | $2^{31}$ | $2^{151.45}$ |

In the attack on 7.25-round ChaCha256, we use the set of 121 PNBs $PNB_{\mathcal{OD}}$ as the same in [2], by assigning a threshold of 0.44. By assigning suitable values at the PNBs, we achieve a backward correlation $\varepsilon_a = 0.002$. For the five output difference bits $\mathcal{OD}_1, \cdots, \mathcal{OD}_5$, we find 230, 225, 226, 206 and 219 PNBs, respectively, by assigning the same threshold of 0.44. The backward correlations of these five PNB sets are 0.0358, 0.0825, 0.1814, 0.0828 and 0.0848, respec-

tively. The PNB sets used in the attack, i.e., $PNB'_{(2,0)}$, $PNB'_{(6,7)}$, $PNB'_{(6,19)}$, $PNB'_{(10,12)}$ and $PNB'_{(14,0)}$, are listed in the following table. Considering the forward correlation $\varepsilon_d = 2^{-34.56}$, and then we have

$$\varepsilon = 2^{-34.56} \times 0.002 \times 0.0358 \times 0.0825 \times 0.1814 \times 0.0828 \times 0.0848 \approx 2^{-61.54}$$

For $\alpha = 6$, we can get $N = 2^{128.19}$. Since the attack process has to be repeated $2^7$ times, the final time and data complexities of this attack are $2^{253.18} \times 2^7 = 2^{260.18}$ and $2^{128.19} \times 2^7 = 2^{135.19}$. As shown in Table 5, the memory complexity of the attack on 7.25-round ChaCha256 is about $2^{178.19}$.

Table 4: The PNB sets used in the attack on 7.25-round ChaCha256

| **Set** | **Key bits** | **Count** | **Correlation** |
|---|---|---|---|
| $PNB'_{(2,0)}$ | {0, 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 39, 40, 48, 49, 50, 51, 52, 53, 54, 75, 76, 77, 80, 87, 88, 92, 93, 94, 96, 97, 98, 102, 103, 104, 105, 106, 107, 112, 113, 114, 115, 116, 117, 118, 119, 122, 130, 131, 132, 133, 134, 135, 136, 137, 143, 144, 145, 146, 147, 148, 149, 150, 151, 190, 196, 197, 208, 209, 212, 213, 214, 215, 216, 217, 228, 233, 234, 235, 236, 237, 238, 239, 243, 249, 250, 251, 252} | 109 | 0.0358 |
| $PNB'_{(6,7)}$ | {0, 1, 2, 3, 14, 15, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 39, 40, 41, 42, 43, 48, 49, 50, 51, 69, 70, 77, 78, 79, 80, 81, 82, 87, 88, 92, 93, 94, 96, 97, 98, 100, 101, 102, 103, 104, 105, 106, 107, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 130, 135, 136, 137, 138, 139, 143, 144, 145, 146, 147, 148, 149, 150, 151, 190, 202, 203, 208, 209, 212, 213, 214, 215, 228, 233, 234, 235, 236, 237, 238, 239, 240, 243, 249, 250, 251, 252, 253, 254} | 104 | 0.0825 |

| | | | |
|---|---|---|---|
| $PNB'_{(6,19)}$ | {0, 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 19, 26, 27, 35, 36, 37, 38, 39, 40, 41, 42, 43, 51, 52, 53, 54, 69, 70, 75, 76, 77, 78, 79, 80, 81, 82, 92, 93, 94, 96, 97, 98, 100, 101, 102, 103, 104, 105, 106, 107, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 147, 148, 149, 150, 151, 190, 202, 203, 212, 213, 214, 215, 216, 217, 228, 229, 230, 233, 234, 235, 236, 237, 238, 239, 240, 243, 249, 250, 251, 252} | 105 | 0.1814 |
| $PNB'_{(10,12)}$ | {0, 4, 5, 6, 12, 13, 22, 23, 32, 33, 34, 35, 36, 37, 38, 39, 40, 48, 49, 50, 51, 52, 53, 54, 69, 70, 75, 76, 77, 78, 79, 80, 81, 82, 87, 88, 92, 93, 94, 96, 97, 98, 100, 101, 102, 103, 104, 115, 116, 117, 118, 130, 131, 132, 133, 134, 143, 144, 145, 146, 147, 190, 196, 197, 208, 209, 212, 213, 214, 215, 216, 217, 228, 229, 230, 236, 237, 238, 239, 249, 250, 251, 252, 253, 254} | 85 | 0.0828 |
| $PNB'_{(14,0)}$ | {0, 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 13, 16, 17, 18, 19, 22, 23, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 48, 49, 50, 51, 52, 53, 54, 69, 70, 75, 76, 77, 78, 79, 80, 81, 82, 87, 88, 104, 105, 106, 107, 112, 120, 121, 122, 130, 131, 132, 135, 136, 137, 138, 139, 143, 144, 147, 148, 149, 150, 151, 190, 196, 197, 202, 203, 208, 209, 212, 216, 217, 228, 229, 230, 233, 240, 241, 242, 243, 252, 253, 254} | 98 | 0.0848 |

In the attack on 7.5-round ChaCha256, we use the set of 23 PNBs $PNB_{\mathcal{OD}}$ as the same in [2], by assigning a threshold of 0.26. By assigning suitable values at the PNBs, we achieve a backward correlation $\varepsilon_a = 0.012$. For the three output difference bits $\mathcal{OD}_1, \mathcal{OD}_2, \mathcal{OD}_3$, we find 79, 67 and 122 PNBs, respectively, by assigning the same threshold of 0.26. The backward correlations of these three PNB sets are $2^{-12.22}$, $2^{-7.45}$ and $2^{-20.26}$, respectively. The PNB sets used in the

**Table 5.** The complexity of preparing the list for 7.25-round ChaCha256

| $\mathcal{OD}$ | **List size** | **Complexity** |
|---|---|---|
| $(2, 0)$ | $2^{26}$ | $2^{154.19}$ |
| $(6, 7)$ | $2^{31}$ | $2^{159.19}$ |
| $(6, 19)$ | $2^{30}$ | $2^{158.19}$ |
| $(10, 12)$ | $2^{50}$ | $2^{178.19}$ |
| $(14, 0)$ | $2^{37}$ | $2^{165.19}$ |

attack, i.e., $PNB'_{(2,0)}$, $PNB'_{(7,7)}$ and $PNB'_{(8,0)}$, are listed in the following table. Considering the forward correlation $\varepsilon_d = 0.0032$, and then we have

$$\varepsilon = 0.0032 \times 0.012 \times 2^{-12.22} \times 2^{-7.45} \times 2^{-20.26} \approx 2^{-54.59}$$

For $\alpha = 0$, we can get $N = 2^{112.36}$. Since the attack process has to be repeated $2^5$ times, the final time and data complexities of this attack are $2^{333.55} \times 2^5 = 2^{338.55}$ and $2^{112.36} \times 2^5 = 2^{117.36}$. As shown in Table 7, the memory complexity of the attack on 7.5-round ChaCha256 is about $2^{301.36}$.

Table 6: The PNB sets used in the attack on 7.5-round ChaCha256

| **Set** | **Key bits** | **Count** | **Correlation** |
|---|---|---|---|
| $PNB'_{(2,0)}$ | {0, 1, 19, 35, 36, 37, 38, 39, 40, 41, 42, 43, 51, 56, 57, 58, 63, 64, 65, 66, 67, 69, 76, 77, 79, 80, 83, 84, 85, 92, 111, 112, 116, 140, 141, 167, 180, 181, 191, 204, 205, 206, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 231, 244, 245, 246} | 56 | $2^{-12.22}$ |
| $PNB'_{(7,7)}$ | {14, 64, 65, 66, 73, 74, 76, 77, 88, 89, 90, 91, 96, 97, 98, 99, 100, 101, 102, 108, 115, 116, 117, 118, 119, 124, 125, 126, 135, 136, 137, 138, 224, 225, 226, 234, 237, 238, 239, 243, 244, 245, 250, 251} | 44 | $2^{-7.45}$ |

| | | | |
|---|---|---|---|
| $PNB'_{(8,0)}$ | {0, 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 20, 21, 22, 23, 24, 28, 31, 39, 47, 48, 55, 60, 63, 64, 73, 74, 79, 80, 84, 85, 88, 92, 96, 97, 98, 99, 100, 101, 102, 108, 109, 110, 111, 112, 113, 115, 116, 124, 125, 126, 128, 129, 130, 131, 132, 133, 135, 140, 141, 142, 143, 144, 145, 146, 147, 148, 152, 153, 154, 160, 168, 169, 180, 184, 191, 195, 208, 211, 219, 223, 226, 227, 232, 233, 234, 235, 236, 240, 241, 243, 244, 250, 251, 252, 253, 254} | 99 | $2^{-20.26}$ |

**Table 7.** The complexity of preparing the list for 7.5-round ChaCha256

| $\mathcal{OD}$ | **List size** | **Complexity** |
|---|---|---|
| $(2, 0)$ | $2^{177}$ | $2^{289.36}$ |
| $(7, 7)$ | $2^{189}$ | $2^{301.36}$ |
| $(8, 0)$ | $2^{134}$ | $2^{246.36}$ |

## 5 Conclusions

In this paper, we revisit the attacks on ChaCha256 from IEEE TIT and INDOCRYPT 2024, and find that there are some errors in all these attacks, and then give corrected cryptanalytic attacks on ChaCha256. The corrected results show that the technique proposed in [2] may not be able to obtain improved differential-linear attacks on ChaCha.

## References


1. Bernstein, D.: ChaCha, a variant of Salsa20. In: Workshop Record of SASC, vol. 8, pp. 3-5 (2008). https://cr.yp.to/chacha/chacha-20080120.pdf
2. Dey, S.: Advancing the idea of probabilistic neutral bits: first key recovery attack on 7.5 round ChaCha. IEEE Trans. Inform. Theory, vol. 70, no. 8, pp. 6091-6106 (2024). https://doi.org/10.1109/TIT.2024.3389874
3. Sharma, N.K., Dey, S., Sarkar, S., Maitra, S. (2025). On Improved Cryptanalytic Results Against ChaCha for Reduced Rounds. In: Mukhopadhyay, S., Stănică, P. (eds) Progress in Cryptology-INDOCRYPT 2024, LNCS, vol. 15496, pp. 29-52. Springer, Cham. https://doi.org/10.1007/978-3-031-80311-6_2

4. Bellini, E., Gerault, D., Grados, J., Makarim, R.H., Peyrin, T.: Boosting differential-linear cryptanalysis of ChaCha7 with MILP. IACR Trans. Symmetric Cryptol. 2023(2), pp. 189-223 (2023). https://doi.org/10.46586/tosc.v2023.i2.189-223