Differential-Linear Cryptanalysis on Xoodyak

Orr Dunkelman[1] and Ariel Weizman[2]

[1] Computer Science Department, University of Haifa, Israel
[2] Department of Mathematics, Bar-Ilan University, Israel

Abstract. In this paper we present the first DL cryptanalysis on 4-round Xoodyak and the first related-key DL cryptanalysis on 5-round Xoodyak. We present the first DL distinguishers on 4- and 5-round Xoodoo, and then perform key-recovery attacks on 4- and 5-round Xoodyak using two recent techniques: the partitioning technique and the neutral bits idea. The data complexity of the 4-round DL attack is about $2^{23.34}$ nonces and the times complexity is about $2^{23.34}$ 4-round Xoodoo calls. The data complexity of the 5-round related-key DL attack is about $2^{22.04}$ nonces and the time complexity is about $2^{22.04}$ 5-round Xoodoo calls.

1 Introduction

1.1 Differential-Linear Cryptanalysis

Differential-Linear (DL in short) cryptanalysis [8] studies the relation between the parity of state bits of two ciphertexts generated from two plaintexts with a fixed difference. More precisely, given a difference Ω_I and state bits λ_O, DL cryptanalysis considers plaintexts pair $(P, P' = P \oplus \Omega_I)$, and checks whether the corresponding ciphertext pair (C, C') satisfies $C \cdot \lambda_O = C' \cdot \lambda_O$.

For a DL distinguisher, a cipher E is treated as a decomposition $E = E_1 \circ E_0$. A differential characteristic with a probability of p on E_0 is denoted by $\Omega_I \xrightarrow[E_0]{p} \Omega_M$ and a linear approximation with a bias of ϵ on E_1 is denoted by $\lambda_M \xrightarrow[E_1]{\epsilon} \lambda_O$. A DL characteristic on the entire cipher E relies on such differential characteristic and linear approximation, and its probability is:

$$Pr[C \cdot \lambda_O = C' \cdot \lambda_O \mid P \oplus P' = \Omega_I] = \frac{1}{2} + 2p\epsilon^2.$$

Therefore, such a DL characteristic is denoted by $\Omega_I \xrightarrow[E]{2p\epsilon^2} \lambda_O$.

Recently, two techniques produced to improve DL attacks: the partitioning technique of Leurent [9] and the neutral bits idea of Beierle et al. [2]. We now describe these techniques which are used in our attacks.

1.2 DL Cryptanalysis with Partitioning

The partitioning technique was first suggested to improve the cryptanalysis of ARX ciphers. In [3] Biham and Carmeli suggest the partitioning technique to improve linear cryptanalysis on FEAL-8X [11]. Leurent [9] extends this technique to DL cryptanalysis, and uses it to improve a DL attack on 7-round Chaskey [12]. We present here the technique in the DL settings.

The main idea of the partitioning technique is as follows: Let $\Omega_I \xrightarrow{\frac{1}{2}\pm 2p\epsilon^2} \lambda_O$ be a DL characteristic, based on $\Omega_I \xrightarrow[E_0]{p} \Omega_M, \lambda_M \xrightarrow[E_1]{\epsilon} \lambda_O$. The data complexity of an attack based on such a characteristic is $\mathcal{O}(p^{-2}\epsilon^{-4})$. Assume that one can partition the data into s disjoint subsets $\mathcal{F} = \{A_1, A_2, \ldots, A_s\}$, such that there is one right subset A_i in which the differential characteristic holds with significantly higher probability $p_i \gg p$, while for all other subsets the differential characteristic does not hold. One can now run the DL attack in each subset A_i independently, resulting in data complexity of $\mathcal{O}(s \cdot p_i^{-2}\epsilon^{-4})$: Generating about $s \cdot p_i^{-2}\epsilon^{-4}$ plaintext pairs, and performing the original attack on each subset. The highest bias points on the right subset and the key material defines it. Therefore, if $s \cdot p_i^{-2} < p^{-2}$ then the attack's complexity is reduced.[1]

1.3 DL Cryptanalysis with Neutral Bits

In [4] Biham and Chen suggest the *neutral bits* technique to improve collision and near-collision attacks on SHA-0 [1]. This idea is used also in secret key cryptanalysis (e.g., in [7]). Here we adapt the definitions of Biham and Chen to differential characteristics on block ciphers.

Definition 1 *Let $\Omega_I \rightarrow \Omega_O$ be a differential characteristic, the i'th bit of the block is called a neutral bit (w.r.t. $\Omega_I \rightarrow \Omega_O$) if for each input pair (P, P') that satisfies the characteristic, the pair $(P \oplus e_i, P' \oplus e_i)$ also satisfies the characteristic. In this case e_i is called a neutral vector.*

Using such neutral bits, an adversary could create many right pairs given one right pair. In addition, Beierle et al. [2] use t neutral bits to create neutral linear subspace with 2^t neutral vectors: Given t neutral bits $i_1, \ldots, i_t$ they use all the vectors of the linear subspace $\mathcal{U} = span\{e_{i_1}, \ldots, e_{i_t}\}$ (i.e., vectors of the form $v = \sum_{j=1}^{t} \alpha_j \cdot e_{i_j}, \alpha_j \in \{0, 1\}$) as neutral vectors.[2]

1.4 Our Contributions

In this paper we produce the first DL cryptanalysis on round reduced Xoodyak using these two previous techniques: partitioning and neutral bits. We give a brief description of Xoodyak in Section 2. Then we present the first DL attack on 4-round Xoodyak in Section 3. Finally, we present the first related-key DL attack on 5-round Xoodyak in Section 4.

[1] The partitioning can be applied to plaintexts, ciphertexts, or any other criteria. For example, in [9] the partitioning is performed also on the ciphertexts.

[2] It should be noted that not always all the vectors in such linear subspace are neutral (see [4] that discusses such examples). However, in all of the cases discussed here this is the scenario.

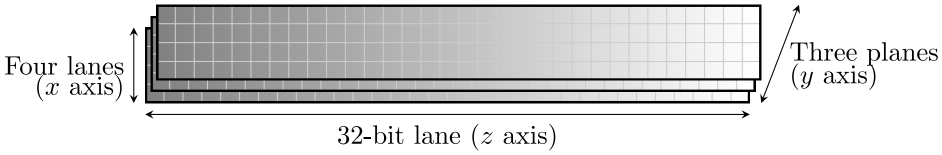

Fig. 1. A Xoodoo state.

2 A Brief Description of Xoodyak [6].

Xoodyak is a cryptographic primitive for hashing, authenticated encryption, and MAC computation, and is one of the finalists of the NIST LightWeight Cryptography (LWC) competition. Xoodyak relies on Xoodoo, a family of 384-bit to 384-bit permutations [5]. A 384-bit state is represented by three *planes*, each consists of four 32-bit *lanes*. The lanes within a plane are indexed by x, the planes are indexed by y, and the bits within a lane are indexed by z (see Figure 1). In addition, the i'th bit ($0 \leq i \leq 384$) of a state S is denoted by S_i. Given a state of three planes $S = (A_0, A_1, A_2)$, each round is defined by the following 5 steps:

$$\begin{aligned}
&\theta : \\
&\qquad P \leftarrow A_0 \oplus A_1 \oplus A_2 \\
&\qquad E \leftarrow P \lll (1,5) \oplus P \lll (1,14) \\
&\qquad A_y \leftarrow A_y \oplus E, y \in \{0,1,2\} \\
&\rho_{west} : \\
&\qquad A_1 \leftarrow A_1 \lll (1,0) \\
&\qquad A_2 \leftarrow A_2 \lll (0,11) \\
&\iota : \\
&\qquad A_0 \leftarrow A_0 \oplus C_i \\
&\chi : \\
&\qquad B_0 \leftarrow \overline{A_1} \wedge A_2 \\
&\qquad B_1 \leftarrow \overline{A_2} \wedge A_0 \\
&\qquad B_2 \leftarrow \overline{A_0} \wedge A_1 \\
&\qquad A_y \leftarrow A_y \oplus B_y, y \in \{0,1,2\} \\
&\rho_{east} : \\
&\qquad A_1 \leftarrow A_1 \lll (0,1) \\
&\qquad A_2 \leftarrow A_2 \lll (2,8)
\end{aligned}$$

Where $A_y \lll (i,j)$ denotes the left rotation which moves the bit in (x,z) to the new position $(x+i \pmod 4, z+j \pmod{32})$, C_i is a round constant, and $\overline{A_y}$ denotes the bitwise complement of A_y. All operations but χ are affine.

Xoodyak uses two modes: hash mode and keyed mode. Here, we discuss the keyed mode, and in particular the initialization phase: The first plane is initialized by an 128-bit key, and the additional two planes by a 256-bit nonce.

Then, Xoodoo is performed on the initialized state, and the first 192 bits are visible and XORed to the first block of the plaintext.

3 The 4-Round DL Attack

We now present the first DL distinguisher[3] on 4-round Xoodoo, and then a DL attack that based on it. Recall that the first plane A_0 is initialized by an 128-bit key, and the last two planes A_1, A_2 are initialized by a 256-bit nonce. Therefore, to mount a DL attack on Xoodoo, the DL characteristic is restricted: the input difference can be only in the last two planes, and the active bits of the output mask can be only in the first 192 bits, which are visible.

3.1 Description of Our Distinguisher.

To choose the input difference we examine the first two steps of the round function: θ and ρ_{west}. We note that given an input difference with two active bits in one column, then θ does not change the difference, and ρ_{west} shifts each bit by a different number of positions, resulting in two active S-boxes in the S-box layer χ (the constant addition does not change the difference). For comparison, if the input difference contains only one active bit, then after θ, in addition to this active bit, there are three additional active bits at two columns, and ρ_{west} shifts each bit by a different number of positions, resulting in 7 active S-boxes in the S-box layer χ. We thus consider an input difference of the form $(0, e_i, e_i), 0 \leq i < 128$.

Following the rotation-invariant property of Xoodoo's characteristics, and for sake of clarity, we consider the input difference $(0, e_0, e_0)$, but this characteristic can be easily rotated. This input difference leads to two active S-boxes before χ: S-box 11 with an input difference of 4 and S-box 32 with an input difference of 2. Denote the output differences (after χ) at S-box 11 by Ω_{11}, and the output differences (after χ) at S-box 32 by Ω_{32}. According to the DDT of χ we have: $\Omega_{11} \in \{4, 5, 6, 7\}, \Omega_{32} \in \{2, 3, 6, 7\}$ in a uniform distribution. We experimentally tested the bias of each DL characteristic with each of the 16 possible differences $(\Omega_{11}, \Omega_{32})$ after the first χ layer, and output mask of one or two active bits after 4.5 more rounds of Xoodoo. The best result was obtained for the output mask[4] $(0, e_{15}, 0)$. As reported in Table 1, the combination $(\Omega_{11}, \Omega_{32}) = (4, 2)$ results with a bias of $+2^{-6}$, where as $(\Omega_{11}, \Omega_{32}) = (4, 6)$ results with a bias of $+2^{-8}$. The

[3] Liu et al. [10] present a 4-round rotational DL distinguisher, with the highest possible bias of $\frac{1}{2}$, without any attack that uses it. Beyond the difference between a DL distinguisher and a rotational DL distinguisher, in this paper we focus on DL attacks (and not just distinguishers) using additional techniques. We give in Appendix A the rotational DL distinguisher used by Liu et al.

[4] In detail, for each $0 \leq i < 128$, when the input difference is $(0, e_i, e_i)$, the best results occurs for the output mask $(0, e_{32 \cdot \lfloor \frac{i}{32} \rfloor + (15+i \pmod{32})}, 0)$. It should be noted that since the mask is in the second plane and only the first 64 bits of this plane are visible, we can not use all the 128 characteristics, but only the 64 characteristics for which $0 \leq i < 64$. However, this fact does not impact our analysis.

The differences $(\Omega_{11}, \Omega_{32})$	$(4,2)$	$(4,6)$	$\neq (4,2),(4,6)$
Signed Bias	$+2^{-6}$	$+2^{-8}$	≈ 0

Table 1. The bias of $(0, e_{32}, \Delta) \to (0, e_{15}, 0)$ (where $\Delta \in \{e_{11}, e_{11,32}\}$) DL characteristic on 4-round Xoodoo, starting just after the first χ. The differences at columns 11 and 32 after the first χ is denoted by $(\Omega_{11}, \Omega_{32})$.

other differences have a bias of about zero. Summing all of these characteristics, we get the following DL characteristic:

$$(0, e_0, e_0) \xrightarrow[\text{4-round Xoodoo}]{\approx 2^{-9.68}} (0, e_{15}, 0).$$

The bias is calculated as follows: $\frac{1}{16} \cdot 2^{-6} + \frac{1}{16} \cdot 2^{-8} + \frac{14}{16} \cdot 0 \approx 2^{-9.68}$. In terms of state indexes, the input difference is $e_{128,256}$ and the output mask is e_{143}. We experimentally verified the bias, using 2^{28} pairs, observing a bias of about $2^{-9.7}$.

3.2 Improving the Distinguisher Using Neutral Bits

We now look for bits of the initial state, and in particular bits that are initialized by the nonce, that *do not* influence the output of the two active S-boxes in the first χ: S-box 11 and S-box 32. Denote the initial state by S, and the state just before the S-box layer χ by T (i.e., $T = \iota \circ \rho_{west} \circ \theta(S)$). In these terms, the two non-active bits of the 11'th S-box are: T_{11}, T_{139}, and the two non-active bits of the 32'nd S-box are: T_{32}, T_{288}. Each of them could be represented as the XOR of 7 bits of the initial state, as follows (see Figure 2):

$$\begin{aligned}
T_{11} &= \bigoplus_{i \in I_{11}} S_i, \;\; I_{11} = \{11, 102, 125, 230, 253, 358, 381\},\\
T_{139} &= \bigoplus_{i \in I_{139}} S_i, \; I_{139} = \{70, 93, 198, 221, 235, 326, 349\},\\
T_{32} &= \bigoplus_{i \in I_{32}} S_i, \;\; I_{32} = \{18, 27, 32, 146, 155, 274, 283\},\\
T_{288} &= \bigoplus_{i \in I_{288}} S_i, \; I_{288} = \{7, 16, 135, 144, 263, 272, 309\}.
\end{aligned} \tag{1}$$

It means that there are 28 bits of the initial state that influence the two active S-boxes (i.e., that influence the two non active bits of each active S-box), and 18 of them are initialized by the nonce. Therefore, we have $256 - 18 = 238$ neutral bits. By fixing all the 18 bits that influence these active S-boxes (i.e., all the non-neutral bits) in all of the nonces, we get the same values at the active S-boxes, which yields the same output difference. Hence, by generating about 2^4 sets of about $2^{13.34}$ nonce pairs (this number was calculated according to [13]), each is defined by another fixed value of the non-neutral bits, the good values (i.e., the values which satisfy $(\Omega_{11}, \Omega_{32}) = (4, 2)$) are expected to appear in about one set, which has the highest bias.

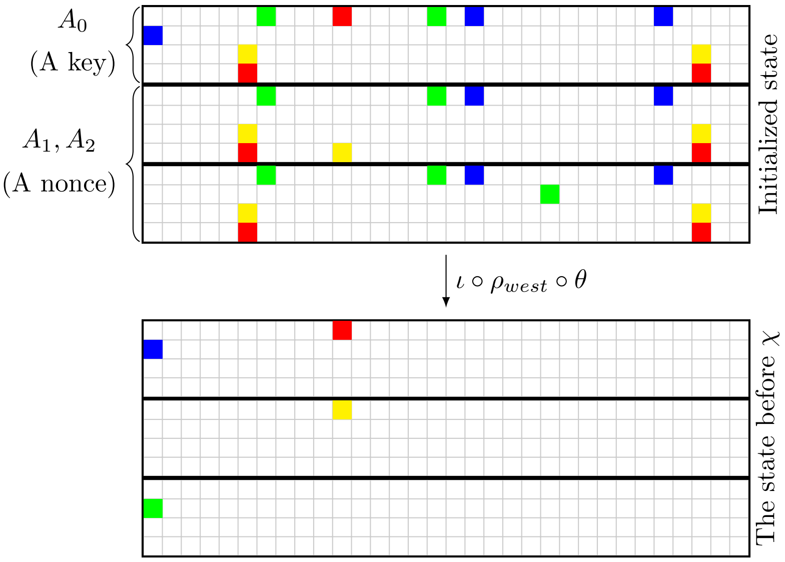

Fig. 2. The non-neutral bits that used in the DL attack on 4-round Xoodyak. Each colored bit of the state before χ is defined as the XOR of the appropriate colored bits of the initialized state.

3.3 Attacking 4-Round Xoodyak Using the Partitioning Technique

We now describe how the partitioning technique allows us to link between the good set (or, in other words, the good values for the non-neutral bits) and four key bits. As mentioned above, given a good pair (i.e., a pair that satisfies the first round) $S = K \parallel N, S' = K \parallel N'$ (where, K is the key, and N, N' are the nonces), we know that $(\Omega_{11}, \Omega_{32}) = (4, 2)$. According to the DDT of χ, the transition 4 → 4 occurs when the input values are 2 and 6 and the transition 2 → 2 occurs when the input values are 1 and 3. Thus, according to Table 1:

$$\begin{aligned} T_{11} \oplus T'_{11} &= 0, \\ T_{139} \oplus T'_{139} &= 1, \\ T_{32} \oplus T'_{32} &= 1, \\ T_{288} \oplus T'_{288} &= 0, \end{aligned}$$

where $T = \iota \circ \rho_{west} \circ \theta(S), T' = \iota \circ \rho_{west} \circ \theta(S')$. Therefore, we get the following four equations:

$$\begin{aligned} K_{11} \oplus K_{102} \oplus K_{125} &= N_{230} \oplus N_{253} \oplus N_{358} \oplus N_{381}, \\ K_{70} \oplus K_{93} &= N_{198} \oplus N_{221} \oplus N_{235} \oplus N_{326} \oplus N_{349} \oplus 1, \\ K_{18} \oplus K_{27} \oplus K_{32} &= N_{146} \oplus N_{155} \oplus N_{274} \oplus N_{283} \oplus 1, \\ K_{7} \oplus K_{16} &= N_{135} \oplus N_{144} \oplus N_{263} \oplus N_{272} \oplus N_{309}, \end{aligned} \tag{2}$$

Algorithm 1 DL Attack on 4-Round Xoodyak (Recovering 4 key bits).

```
Set an array keyOptions of $2^4$ key values to zeroes. The keyOptions bits are defined as the XOR of the key bits from Eq. (2).
for all $k \in \{0,1\}^4$ do
    Fix values for the non-neutral nonce bits, that satisfy Eq. (2).
    for all $1 \leq i \leq 2^{13.34}$ do
        Generate a nonce (according to the fixed bits) $N_i$, and set the pairs ($S = K \parallel N_i, S' = K \parallel N_i \oplus e_{128,256}$) as two initial states.
        Request the output of these initial states after the first performance of Xoodoo, denoted by $(O_i, O'_i)$.
        if $O_{i_{143}} = O'_{i_{143}}$ then
            Increment keyOptions[$k$].
        end if
    end for
end for
Output the key $k$ such that keyOptions[$k$] = $\max\{$keyOptions[$j$]$\}$.
```

where the key bits are indexed by $0 \leq i < 128$ and the nonce bits are indexed by $128 \leq i < 320$. It means that there is a partitioning of the space to 16 subsets, depending on four key values:

$$\begin{aligned}
k_0 &= K_{11} \oplus K_{102} \oplus K_{125}, \\
k_1 &= K_{70} \oplus K_{93}, \\
k_2 &= K_{18} \oplus K_{27} \oplus K_{32}, \\
k_3 &= K_7 \oplus K_{16}.
\end{aligned}$$

Each value for the bits k_0, k_1, k_2, k_3 determines another subset of the non-neutral nonce bits, in which the characteristic has a bias of 2^{-6} instead of $2^{-9.7}$, when the nonces are generated randomly. Algorithm 1 describes the attack. The data complexity required to revealed four key bit is about $2^4 \cdot 2^{13.34} \cdot 2 = 2^{18.34}$ nonces, and the time complexity is about $2^{18.34}$ 4-round Xoodoo calls. We experimentally verified the attack using 100 different keys. The observed success rate was 85%. Following the rotation-invariant property of Xoodoo's characteristics, it is possible to recover the entire key with data complexity of about $2^{23.34}$ nonces and time complexity of about $2^{23.34}$ 4-round Xoodoo calls.

4 The 5-Round Related-Key DL Attack

We now present the first DL distinguisher on 5-round Xoodoo [5], and then a related-key DL attack based on it. To construct our 5-round DL distinguisher we first construct a 4-round DL distinguisher and then add additional round at the beginning.

4.1 Description of Our Distinguisher

Similarly to the input difference $(0, e_i, e_i)$ of the 4-round DL characteristic that described in Section 3, the input differences of the form $(e_i, e_i, 0)$ and $(e_i, 0, e_i)$ are also good candidates, with an additional requirement: Due to the fact that there is an active bit in the first plane, initialized by a key, an attack using these characteristics requires related keys. Our experiments show that $(e_i, 0, e_i)$ offers better results than $(e_i, e_i, 0)$ and thus the reminder of our analysis concentrates on input difference of this form.

Following the rotation-invariant property of Xoodoo's characteristics, and for sake of clarity, we consider the input difference $(e_0, 0, e_0)$, but this characteristic can be easily rotated. This input difference leads to two active S-boxes before χ: S-box 0 with an input difference of 1 and S-box 11 with an input difference of 4. Denote the output differences (after χ) at S-box 0 by $\varOmega_0$, and the output differences (after χ) at S-box 11 by $\varOmega_{11}$. According to the DDT of χ we have: $\varOmega_0 \in \{1, 3, 5, 7\}, \varOmega_{11} \in \{4, 5, 6, 7\}$ in a uniform distribution. We experimentally tested the bias of each DL characteristic with each of the 16 possible differences $(\varOmega_0, \varOmega_{11})$ after the first χ layer and output mask of one or two active bits after 3.5 more rounds of Xoodoo. The best result was obtained for the output mask $(e_0, 0, 0)$. As reported in Table 2, the combinations $(\varOmega_0, \varOmega_{11}) \in \{(1, 4), (1, 6)\}$ result with a bias of -2^{-3}, the combinations $(\varOmega_0, \varOmega_{11}) \in \{(1, 5), (1, 7), (3, 4), (3, 6)\}$ result with a bias of -2^{-5}, and the combinations $(\varOmega_0, \varOmega_{11}) \in \{(3, 5), (3, 7)\}$ result with a bias of -2^{-7}. The other differences have a bias of about zero. Summing all of these characteristics, we get the following DL characteristic:

$$(e_0, 0, e_0) \xrightarrow[\text{4-round Xoodoo}]{\approx -2^{-5.36}} (e_0, 0, 0).$$

The bias is calculated as follows: $-\frac{2}{16} \cdot 2^{-3} - \frac{4}{16} \cdot 2^{-5} - \frac{2}{16} \cdot 2^{-7} + \frac{8}{16} \cdot 0 \approx -2^{-5.36}$. In terms of state indexes, the input difference is $e_{0,256}$ and the output mask is e_0. We experimentally verified the bias, using 2^{28} pairs.[5]

We now add one round at the beginning, by performing the inverse of the round function step by step. First, ρ_{east}^{-1} transforms $(e_0, 0, e_0)$ to $(e_0, 0, e_{88})$. Then χ^{-1} maintains this difference with probability of 2^{-4} (i.e., 2^{-2} for each S-box), which is not changed by ι^{-1}. Finally, the difference $(e_0, 0, e_{88})$ is transformed by $\theta^{-1} \circ \rho_{west}^{-1}$ to $\varOmega_I = (\varOmega A_0, \varOmega A_1, \varOmega A_2)$, where

$$\begin{aligned} \varOmega A_0 &= a8b23b19\ 98810919\ 52674513\ 95a876f3_x \\ \varOmega A_1 &= a8b23b18\ 98810919\ 52674513\ 95a876f3_x \\ \varOmega A_2 &= a8b23b18\ 98810919\ 52676513\ 95a876f3_x. \end{aligned}$$

Therefore, the entire DL distinguisher for 5-round Xoodoo is:

$$(\varOmega A_0, \varOmega A_1, \varOmega A_2) \xrightarrow[\text{5-round Xoodoo}]{-2^{-9.36}} (e_0, 0, 0).$$

[5] In detail, for each $0 \leq i \leq 128$, when the input difference is $(e_i, 0, e_i)$, the best results occurs for the output mask $(e_i, 0, 0)$.

The differences ($\varOmega_0, \varOmega_{11}$)	$(1,4)$ $(1,6)$	$(1,5)$ $(1,7)$ $(3,4)$ $(3,6)$	$(3,5)$ $(3,7)$	Differences of the form $(5,\varOmega_{11}),(7,\varOmega_{11})$ where $\varOmega_{11} \in \{4,5,6,7\}$
Signed Bias	-2^{-3}	-2^{-5}	-2^{-7}	≈ 0

Table 2. The bias of $e_{0,11} \rightarrow e_0$ DL characteristic on 5-round Xoodoo, starting just after the first χ. The differences at columns 0 and 11 after the first χ is denoted by $(\varOmega_0, \varOmega_{11})$.

4.2 Improving the Distinguisher Using Neutral Bits

We now present an attack on 5-round Xoodyak, which reveals four key bits of the initial state. We look for bits of the initial state, and in particular those initialized by the nonce, that not influence the output of the two active S-boxes in the first χ layer: S-box 0 with an input difference of 1 and S-box 88 with an input difference of 4. Denote the output differences (after χ) by $\varOmega_0, \varOmega_{88}$, respectively. As mentioned above, we need $(\varOmega_0, \varOmega_{88}) = (1,4)$, which happens with a probability of 2^{-4} using random data. Denote the initial state by S, and the state just before the S-box layer χ by T (i.e., $T = \iota \circ \rho_{west} \circ \theta(S)$). In these terms, the two non-active bits of the 0'th S-box are: T_{128}, T_{256} and the two non-active bits of the 88'th S-box are: T_{88}, T_{216}. Each of them could be represented as the XOR of 7 bits of the initial state, as follows:

$$
\begin{aligned}
T_{128} &= \bigoplus_{i \in I_{128}} S_i,\ I_{128} = \{82, 91, 210, 219, 224, 338, 347\},\\
T_{256} &= \bigoplus_{i \in I_{256}} S_i,\ I_{256} = \{103, 112, 231, 240, 277, 359, 368\},\\
T_{88} &= \bigoplus_{i \in I_{88}} S_i,\ \ I_{88} = \{42, 51, 88, 170, 179, 298, 307\},\\
T_{216} &= \bigoplus_{i \in I_{216}} S_i,\ I_{216} = \{10, 19, 138, 147, 184, 266, 275\}.
\end{aligned}
\tag{3}
$$

It means that there are 28 bits of the initial state that influence the two active S-boxes (i.e., that influence the two non active bits of each active S-box), and 19 of them are initialized by a nonce. Therefore, we have $256 - 19 = 237$ neutral bits. By fixing all the 19 bits that influence these active S-boxes (i.e., all the non-neutral bits) in all of the nonces, we get the same values at the active S-boxes, which yields the same output difference. Hence, by generating about 2^4 sets of about $2^{12.04}$ initial state pairs (this number was calculated according to [13]), each is defined by another fixed value of the non-neutral bits, the good values (i.e., the values which satisfy $(\varOmega_0, \varOmega_{88}) = (1,4)$) are expected to appear in about one set, which has the highest bias. To produce an attack using this characteristic, we need also the partitioning technique.

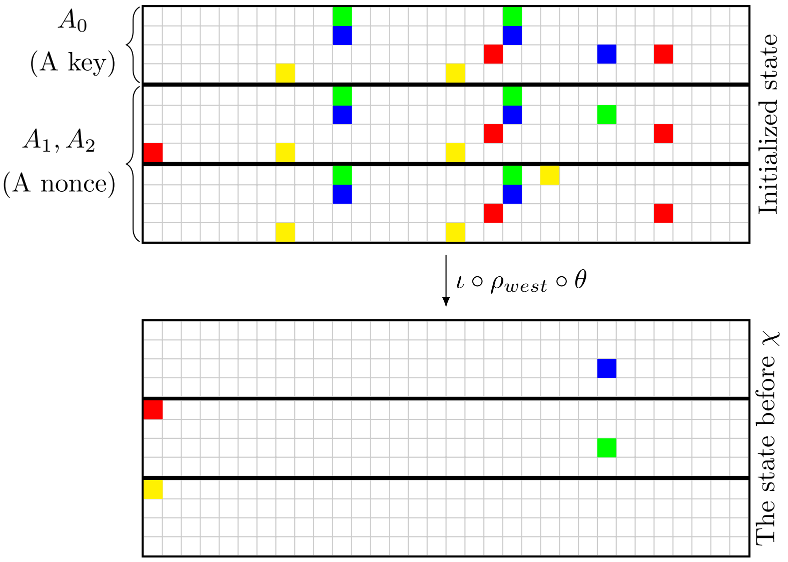

Fig. 3. The non-neutral bits that used in the related-key DL attack on 5-round Xoodyak. Each colored bit of the state before χ is defined as the XOR of the appropriate colored bits of the initialized state.

4.3 Attacking 5-Round Xoodyak Using the Partition Technique

We now describe how the partitioning technique allows us to link between the good set (or, in other words, the good values for the non-neutral bits) and four key bits. As mentioned above, given a good pair (i.e., a pair that satisfies the first round) $S = K \parallel N, S' = (K \parallel N) \oplus \Omega_I$, we know that $(\Omega_0, \Omega_{88}) = (1, 4)$. According to the DDT of χ, the transition $1 \to 1$ occurs when the input values are 4 and 5 and the transition $4 \to 4$ occurs when the input values are 2 and 6. Thus, according to Table 3:

$$\begin{aligned}
T_{128} \oplus T'_{128} &= 0\\
T_{256} \oplus T'_{256} &= 1\\
T_{88} \oplus T'_{88} &= 0\\
T_{216} \oplus T'_{216} &= 1
\end{aligned}$$

where $T = \iota \circ \rho_{west} \circ \theta(S), T' = \iota \circ \rho_{west} \circ \theta(S')$. Therefore, we get the following four equations:

$$\begin{aligned}
K_{82} \oplus K_{91} &= N_{210} \oplus N_{219} \oplus N_{224} \oplus N_{338} \oplus N_{347},\\
K_{103} \oplus K_{112} &= N_{231} \oplus N_{240} \oplus N_{277} \oplus N_{359} \oplus N_{368} \oplus 1,\\
K_{42} \oplus K_{51} \oplus K_{88} &= N_{170} \oplus N_{179} \oplus N_{298} \oplus N_{307},\\
K_{10} \oplus K_{19} &= N_{138} \oplus N_{147} \oplus N_{184} \oplus N_{266} \oplus N_{275} \oplus 1.
\end{aligned} \tag{4}$$

Algorithm 2 DL Attack on 5-Round Xoodyak (Recovering 4 key bits).

```
Set an array keyOptions of $2^4$ key values to zeroes. The keyOptions bits are defined as the XOR of the key bits from Eq. (4).
for all $k \in \{0,1\}^4$ do
    Fix values for the non-neutral nonce bits, that satisfy Eq. (4).
    for all $1 \leq i \leq 2^{12.04}$ do
        Generate a nonce (according to the fixed bits) $N_i$, and set the pairs $(S = K \parallel N_i, S' = (K \parallel N_i) \oplus \Omega_I)$ as two initial states.
        Request the output of these initial states after the first performance of Xoodoo, denoted by $(O_i, O'_i)$.
        if $O_{i_0} = O'_{i_0}$ then
            Increment keyOptions[k].
        end if
    end for
end for
Output the key $k$ such that $keyOptions[k] = \min\{keyOptions[j]\}$.
```

where the key bits are indexed by $0 \leq i < 128$ and the nonce bits are indexed by $128 \leq i < 320$. It means that there is a partitioning of the space to 16 subsets, depending on four key values:

$$\begin{aligned} k_0 &= K_{82} \oplus K_{91}, \\ k_1 &= K_{103} \oplus K_{112}, \\ k_2 &= K_{42} \oplus K_{51} \oplus K_{88}, \\ k_3 &= K_{10} \oplus K_{19}. \end{aligned}$$

Each value for k_0, k_1, k_2, k_3 determines another subset of the non-neutral nonce bits, in which the characteristic has the bias of $2^{-5.36}$, instead of $2^{-9.36}$ when the nonces are generated randomly. Algorithm 2 describes the attack. The data complexity required to reveal four key bits is about $2^4 \cdot 2^{12.04} \cdot 2 = 2^{17.04}$ nonces, and the time complexity is about $2^{17.04}$ 5-round Xoodoo performances. We experimentally verified the attack using 100 different keys. The observed success rate was 89%. Following the rotation-invariant property of Xoodoo's characteristics, it is possible to recover the entire key with data complexity of about $2^{22.04}$ nonces and time complexity of about $2^{22.04}$ 5-round Xoodoo encryptions.

References

1. Federal Information Processing Standard (FIPS) 180, 1993.
2. Christof Beierle, Gregor Leander, and Yosuke Todo. Improved differential-linear attacks with applications to ARX ciphers. In *Advances in Cryptology - Proceedings of CRYPTO 2020*, volume 12172 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.
3. Eli Biham and Yaniv Carmeli. An improvement of linear cryptanalysis with addition operations with applications to FEAL-8X. In *Selected Areas in Cryptography, SAC 2014*, volume 8781 of *Lecture Notes in Computer Science*, pages 59–76. Springer, 2014.
4. Eli Biham and Rafi Chen. Near-collisions of SHA-0. In *Advances in Cryptology - Proceedings of CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2004.
5. Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. The design of Xoodoo and Xoofff. *IACR Trans. Symmetric Cryptol.*, 2018(4):1–38, 2018.
6. Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Xoodyak, a lightweight cryptographic scheme. *IACR Trans. Symmetric Cryptol.*, 2020(1):60–87, 2020.
7. Orr Dunkelman, Nathan Keller, and Adi Shamir. A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony. *J. Cryptology*, 27(4):824–849, 2014.
8. Susan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In *Advances in Cryptology - proceedings of CRYPTO 1994*, volume 839 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 1994.
9. Gaëtan Leurent. Improved differential-linear cryptanalysis of 7-round chaskey with partitioning. In *Advances in Cryptology - Proceedings of EUROCRYPT 2016*, volume 9665 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2016.
10. Yunwen Liu, Siwei Sun, and Chao Li. Rotational Cryptanalysis from a Differential-Linear Perspective - Practical Distinguishers for Round-Reduced FRIET, Xoodoo, and Alzette. In *Advances in Cryptology - proceedings of EUROCRYPT 2021*, volume 12696 of *Lecture Notes in Computer Science*, pages 741–770. Springer, 2021.
11. Shoji Miyaguchi. The FEAL cipher family. In *Advances in Cryptology - Proceedings of CRYPTO 1990*, volume 537 of *Lecture Notes in Computer Science*, pages 627–638. Springer, 1990.
12. Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In *Selected Areas in Cryptography, SAC 2014*, volume 8781 of *Lecture Notes in Computer Science*, pages 306–323. Springer, 2014.
13. Ali Aydin Selçuk. On Probability of Success in Linear and Differential Cryptanalysis. *J. Cryptology*, 21(1):131–147, 2008.

A A Rotational DL Distinguisher On 4-Round Xoodoo [10]

In [10] Liu et al. present the first rotational DL distinguisher on 4-round Xoodoo, by constructing a 3-round rotational DL distinguisher and adding one round at the beginning. They show that given a pair with all-zero difference and left-rotate mount of one bit (i.e., $(P, P' = P \lll 1)$), then after 3-round Xoodoo there are many high-biased bits, including the highest bias of half on the following masks: 10000_x at lane $(1, 0)$, 20000_x at lane $(1, 1)$, and 1000000_x at lane $(3, 2)$. To add one round at the beginning they note that since the round constant is XORed right after the two linear steps, it is possible to choose an input RX-difference such that the injection of the round constant cancels the difference, resulting with all-zero difference and left-rotate mount of one bit. For the first round constant of 4-round Xoodoo $C = 00000480_x$, the required input difference is $\mathit{\Omega}_I = (\mathit{\Omega} A_0, \mathit{\Omega} A_1, \mathit{\Omega} A_2)$ where

$$\begin{aligned}\mathit{\Omega} A_0 &= 484ccc80\ 3ab9821a\ 37b6cde9\ 45a3f0cb_x,\\ \mathit{\Omega} A_1 &= 484cc800\ 3ab9821a\ 37b6cde9\ 45a3f0cb_x,\\ \mathit{\Omega} A_2 &= 484cc800\ 3ab9821a\ 37b6cde9\ 45a3f0cb_x.\end{aligned}$$

Therefore, given a plaintext pair $(P, P' = (P \lll 1) \oplus \mathit{\Omega}_I)$, their ciphertext pair (after 4-round Xoodoo) (C, C') satisfies:

$$\lambda \cdot (C \lll 1) = \lambda \cdot C'.$$