# Another Look at Differential-Linear Attacks

Orr Dunkelman[1,⋆] and Ariel Weizman[2,⋆⋆]

[1] Computer Science Department, University of Haifa, Israel
orrd@cs.haifa.ac.il
[2] Department of Mathematics, Bar-Ilan University, Israel
Nathan.Keller@biu.ac.il,relweiz@gmail.com

**Abstract.** Differential-Linear (DL) cryptanalysis is a well known cryptanalytic technique that combines differential and linear cryptanalysis. Over the years, multiple techniques were proposed to increase its strength and applicability. Two relatively recent ones are: The partitioning technique by Leurent and the use of neutral bits adapted by Beierle et al. to DL cryptanalysis.

In this paper we compare these techniques and discuss the possibility of using them together to achieve the best possible DL attacks. We study the combination of these two techniques and show that in many cases they are indeed compatible. We demonstrate the strength of the combination in two ways. First, we present the first DL attack on 4-round Xoodyak and an extension to 5-round in the related key model. We show that the attacks are possible only by using these two techniques simultaneously. In addition, using the combination of the two techniques we improve a DL attack on 9-round DES. We show that the partitioning technique mainly reduces the time complexity, and the use of neutral bits mainly reduces the data complexity, while the combination of them reduces both the time and data complexities.

**Keywords:** Differential-Linear Cryptanalysis, Partitioning, Neutral Bits, Xoodyak, DES.

## 1 Introduction

### 1.1 Differential and Linear Cryptanalysis

The two main statistical cryptanalytic techniques are differential cryptanalysis [10] and linear cryptanalysis [27]. Differential cryptanalysis was introduced

⋆ The first author was supported in part by the Center for Cyber, Law, and Policy in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office and by the Israeli Science Foundation through grants No. 880/18 and 3380/19.

⋆⋆ The second author was supported by the European Research Council under the ERC starting grant agreement n. 757731 (LightCrypt), by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office, and by the President Scholarship for Ph.D. students at the Bar-Ilan University.

by Biham and Shamir [10]. It analyzes the development of differences of plaintext pairs through the encryption process. Let $E$ be an $n$-bit block cipher consisting of $r$ rounds, and denote the input of the $i$'th round by $X_i$. A differential with probability $p$ of $t$ rounds of $E$ is a statistical property of the form $Pr[X_{i+t} \oplus X'_{i+t} = \Omega_O \mid X_i \oplus X'_i = \Omega_I] = p$, denoted by $\Omega_I \xrightarrow{p} \Omega_O$. Differential attacks exploit differential characteristics (with high probability) to recover key material.

Linear cryptanalysis was published by Matsui [27]. It analyzes the development of parities of state bits of a single plaintext through the encryption process. A linear approximation with bias $\epsilon$ is a statistical property of the form $Pr[C \cdot \lambda_O = P \cdot \lambda_I] = \frac{1}{2} + \epsilon$, for two masks $\lambda_I, \lambda_O$, and $\cdot$ denoting the scalar product. The quality of such linear approximation is measured by the absolute value $|\epsilon|$. Linear attacks exploit linear approximations (with high bias) to recover key materials.

Given the strength of both differential and linear cryptanalysis, modern block ciphers are designed to withstand these attacks. A new block cipher should ensure that there are neither high-probability differential characteristics nor high-bias linear approximations for "many" rounds of the cipher [30], and some design methodologies have been developed to achieve that (e.g., the wide trail strategy [16]). The result is ciphers with a sufficient number of rounds such that there are no differential characteristics of probability $p \gg 2^{-n}$ and no linear approximations of bias $\mid \epsilon \mid \gg 2^{-\frac{n}{2}}$.

### 1.2 Differential-Linear Cryptanalysis

While it is possible to provide resistance against "long" differential characteristics and linear approximations, "short" characteristics (with high probability or bias) are inevitable. This fact led to the development of several cryptanalytic techniques which exploit two "short" characteristics instead of one "long" characteristic. Such techniques look on the cipher $E$ as a decomposition[1] $E = E_1 \circ E_0$, and combine two "short" characteristics, one for $E_0$ and the other for $E_1$, as one "long" characteristic for $E$.

The first combined technique is the *Differential-Linear* (DL in short) cryptanalysis of Langford and Hellman [21]. DL cryptanalysis 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 $\Omega_I$ and state bits $\lambda_O$, DL cryptanalysis considers plaintexts pair $(P, P' = P \oplus \Omega_I)$, and checks whether

[1] We note that some works divide the cipher into three sub-ciphers $E = E_1 \circ E_m \circ E_0$ (e.g., [3, 11–13, 25, 26]). This is mostly done to better understand the transition between the two main sub-ciphers $E_0, E_1$ and most importantly the dependencies between the two sub-ciphers. The emphasis of this paper is the external rounds (rather the internal rounds and the transition). Our results are independent of these works and thus we use the simpler description of DL attacks.

We note that both partition and neutral bits may still result in subtle dependencies which may impact the transition, and hence we experimentally verified our results whenever possible.

the corresponding ciphertext pair $(C, C')$ satisfies $C \cdot \lambda_O = C' \cdot \lambda_O$. Such a DL characteristic $\Omega_I \to \lambda_O$ relies on a differential characteristic $\Omega_I \xrightarrow[E_0]{p} \Omega_M$ and a linear approximation $\lambda_M \xrightarrow[E_1]{\epsilon} \lambda_O$, and the probability of the DL characteristic is:

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

Later, additional combined attacks were published, such as the boomerang attack [32], the amplified boomerang attack [19], and the rectangle attack [9].

Consider a DL attack based on a differential characteristic with a probability of $p$ and a linear approximation with a bias of $\epsilon$, the bias of the full distinguisher is $2p\epsilon^2$, and the data complexity of the corresponding DL attack is $\mathcal{O}\left(p^{-2}\epsilon^{-4}\right)$. Therefore, even a small improvement of the inner characteristics may have a large impact on the data complexity (and as a result also on the time complexity) of the attack.

*Previous Works.* Two previous works show how dependencies between plaintext pairs can improve DL attacks:[2] Leurent [22] extends the partitioning technique of [6] to DL characteristics on ARX ciphers. Using some properties of the modular addition operator, Leurent shows how to partition the data into disjoint subsets (the partition is done both in the differential part and the linear part), such that each subset satisfies the DL characteristic with a higher bias. As a result, in each subset, the bias is significantly higher, resulting in a gain stemming from the squaring of this bias.

Beierle et al. [4] suggest a different approach, which adapts the idea of *Neutral Bits* [7]. For the differential part, Beierle et al. suggest to look for a subspace $\mathcal{U} \subseteq \mathbb{F}_2^n$, such that given a plaintext pair $(P, P')$ satisfying the differential part, then $\forall u \in \mathcal{U} : (P \oplus u, P' \oplus u)$ also satisfies the differential part (with high probability). Therefore, one right pair w.r.t. the differential characteristic produces a (possibly large) set of pairs all of which have the same parity in the beginning of the linear part.

### 1.3 Our Contributions

The main goal of this paper is to study the combination of these two techniques, partitioning and neutral bits, to minimize the attack's data and time complexities. We describe the techniques, compare them, and discuss the possibility of combining them in Section 3. Then, in Section 4, we present the first DL distinguisher on 5-round Xoodoo [14], and show that the use of these two techniques together allows a key-recovery RK attack on 5-round Xoodyak [15], which is impossible using each technique separately.[3] Finally, we show how the combination

[2] A similar idea is used in the chosen-plaintext linear attack of Knudsen and Mathiassen on DES [20].

[3] More precisely, we show that each subset in the partitioning (which is defined according to the key material) determines a good value for the non-neutral bits. Without the combination, the distinguisher cannot be used for a key recovery attack.

of the two techniques can improve two previous DL attacks: We achieve the best DL attack on 9-round DES [1], which improves the previous DL attack of [8] by a factor of about $2^8$ (in Section 5), thus, showing that partitioning works also for S-box based constructions.

## 2 Notations

- $e_i$ $(0 \leq i \leq n-1)$ denotes the $n$-bit word with zeros in all bits but the $i$'th bit, and $e_{i_1,\ldots,i_j} = e_{i_1} \oplus \cdots \oplus e_{i_j}$.
- The probabilities of differential characteristics are denoted by $p$, and the biases of linear approximations are denoted by $\epsilon$.
- 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 $\epsilon$ on $E_1$ is denoted by $\lambda_M \xrightarrow[E_1]{\epsilon} \lambda_O$.
- For a boolean function $f : \{0,1\}^m \to \{0,1\}^n$, the Difference Distribution Table (DDT in short) is the $2^m \times 2^n$ table, which is defined by:
$$\mathrm{DDT}^f[\Omega_I, \Omega_O] = \left|\{(X,X') \mid X \oplus X' = \Omega_I \wedge f(X) \oplus f(X') = \Omega_O\}\right|.$$

## 3 Partitioning, Neutral Bits, and Combination of them

### 3.1 DL Cryptanalysis with Partitioning

The partitioning technique was first proposed to improve the cryptanalysis of ARX ciphers. In [6] Biham and Carmeli suggest the partitioning technique to improve linear cryptanalysis on FEAL-8X [28]. Leurent [22] extends this technique to DL cryptanalysis, and uses it to improve a DL attack on 7-round Chaskey [29]. 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 $\Omega_I \xrightarrow[E_0]{p} \Omega_M, \lambda_M \xrightarrow[E_1]{\epsilon} \lambda_O$. As mentioned above, 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 of plaintexts $A_1, A_2, \ldots, A_s$, 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. Formally, denote the probability of the differential characteristic in a specific subset $A_i$ by $p_i$, we assume that: $\exists 1 \leq i \leq s : p_i \gg p \wedge \forall j \neq i : p_j \approx 0$. One can now run the DL attack in each subset $A_i$ independently, resulting in a 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. Therefore, if $s \cdot p_i^{-2} < p^{-2}$ then the attack's complexity is reduced.[4] Figure 1a illustrates the partitioning technique.

[4] The partitioning can be applied to plaintexts, ciphertexts, or any other criteria. For example, in [22] the partitioning is performed also according to the values of the ciphertexts.

### 3.2 Neutral Bits

In [7] Biham and Chen presented the *neutral bits* technique to improve collision and near-collision attacks on SHA-0. This idea is used also in secret key cryptanalysis (e.g., in [18]). Here we adapt the definitions of Biham and Chen to differential characteristics on block ciphers.

**Definition 1** *Let $\Omega_I \to \Omega_O$ be a differential characteristic, the i'th bit of the block, $e_i$, is called a neutral bit (w.r.t. $\Omega_I \to \Omega_O$) 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.*

Using such neutral bits, an adversary can create many right pairs given one right pair. In addition, Beierle et al. [4] 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.[5] Figure 1b illustrate this idea.

*Reducing Attack's Complexity Using Neutral Bits.* Let $\Omega_I \xrightarrow{\frac{1}{2}+2p\epsilon^2} \lambda_O$ be a DL characteristic that relies on a differential $\Omega_I \xrightarrow[E_0]{p} \Omega_M$ and a linear approximation $\lambda_M \xrightarrow[E_1]{\epsilon} \lambda_O$. The data complexity of an attack based on such a characteristic is $\mathcal{O}(p^{-2}\epsilon^{-4})$. As was done in [4], assume that we have a neutral subspace $\mathcal{U}$ (w.r.t. $\Omega_I \to \Omega_M$) with $|\mathcal{U}| \geq c \cdot \epsilon^{-4}$ for a small $c$, then it is possible to reduce the data complexity by a factor of $p^{-1}$: Generate $p^{-1}$ plaintext pairs $(P_i, P'_i = P_i \oplus \Omega_I)$, and ask for sets of $\epsilon^{-4}$ plaintext pairs of the form $\{(P_i \oplus u, P'_i \oplus u) \mid u \in \mathcal{U}\}$. For the set where $(P_i, P'_i)$ is a right pair w.r.t. the differential, then all the pairs $(P_i \oplus u, P'_i \oplus u)$ are right pairs w.r.t. the differential. Hence, once we have such a pair, then we have $\epsilon^{-4}$ plaintext pairs that satisfy the differential part, which is sufficient to detect a bias of $\epsilon^2$ in the ciphertext pairs.

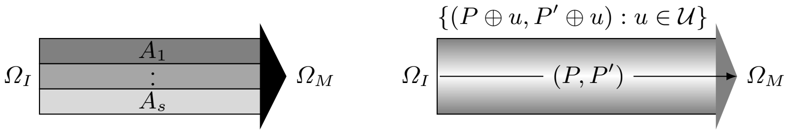
When the neutral subspace is not big enough,[6] it is possible to use a neutral subspace w.r.t. the beginning of the differential (and not the entire differential): Assume that $E_0 = E_{01} \circ E_{00}$, and the differential $\Omega_I \xrightarrow[E_0]{p} \Omega_M$ is composed from two shorter differentials $\Omega_I \xrightarrow[E_{00}]{p_0} \Omega_1, \Omega_1 \xrightarrow[E_{01}]{p_1} \Omega_M$ (for $p_0 \cdot p_1 = p$), and we have a neutral subspace $\mathcal{U}$ w.r.t. $\Omega_I \xrightarrow[E_{00}]{p_0} \Omega_1$ with $|\mathcal{U}| \geq p_1^{-2}\epsilon^{-4}$, then it is possible to reduce the data complexity by factor of $p_0^{-1}$.

In addition, Beierle et al. [4] show that it is possible to use probabilistic neutral bits (PNBs, in short) [2]: Given an input pair $(P, P')$ that satisfies the differential, the pair $(P \oplus e_i, P' \oplus e_i)$ also satisfies the characteristic with high probability[7] (see also [17]).

[5] It should be noted that not always all the vectors in the linear subspace are neutral (see [7] that discusses such examples). However, in all of the cases discussed here this is the scenario.

[6] Similar issue also affected chosen-plaintext linear cryptanalysis [20].

[7] The probability should be significantly higher than the differential's probability.



(a) Partitioning: One good subset. (b) Neutral Bits: Many right pairs from one.

Fig. 1: The Partitioning Technique and the Neutral Bits Idea.

Beierle et al. [4] use these two observations to offer an improved DL attack on Chaskey [29], which achieves better results than achieved by [22] using the partitioning technique, and to perform an improved DL attack on Chacha [5].

### 3.3 A Comparison Between Partitioning and Neutral Bits

We note that these two previous techniques actually improve the data complexity in two different ways:

1. In the partitioning technique, the adversary identifies a subset of the data in which the probability of the differential (or the bias) is higher than for random data. This subset is identified according to an external condition on the data (and not on the pairs among themselves), and therefore cannot be chosen in advance.
2. In the neutral bits technique, the adversary generates subsets of data in advance (with an internal condition on the pairs among themselves), such that in each subset all the pairs satisfy the differential part together (or not).

In other words, the goal of the partitioning technique is to increase the bias by using a partial subset of the data in which the bias is higher. In contrast, the neutral bits technique takes one right pair and generates many right pairs.

In addition to the data complexity reduction, the effect of these techniques on the time complexity should be examined. Beierle et al. [4] point out an advantage of the neutral bits technique: While usually the partitioning technique requires guessing key material which results in increasing the time complexity, the neutral bits technique is independent of the key. However, it depends on the attack details, and it is possible to achieve lower time complexity using the partitioning technique than using neutral bits (see, e.g., Sections 5.2.2 and 5.2.3).

### 3.4 Combining Partitioning and Neutral Bits

When the two techniques refer to the same part of the DL characteristic, then the use of neutral bits may obviates the use of partition. For example, in the case of the two DL attacks [4,22] on Chaskey [29], Leurent [22] uses partitioning in both the differential characteristic and the linear approximation, and Beierle et al. [4] improve the attack by using neutral bits in the differential characteristic instead of partitioning. Beierle et al. do not apply neutral bits on top of the partitioning technique, but replace the partitioning technique on the differential part with neutral bits. However, we point out two situations in which it is possible to use the two techniques together on the differential part, which leads to an attack with lower complexities than can be achieved by using each technique separately.

1. In some cases each subset in the partitioning determines a good value for the non-neutral bits (i.e., an input with this determined value and any value for the neutral bits satisfies the differential part). In Section 4 we present the first DL distinguisher on 5-round Xoodoo [14], and then we apply this idea and show how a combination of the two techniques together allows us to attack 5-round Xoodyak [15].
2. In some cases it is possible to decompose the differential characteristic into two parts, and to perform the partitioning technique on the first part and the neutral bits technique on the second part, to take advantage of these two techniques.[8] We use this idea to improve the DL attack of [8] on round-reduced DES [1] in Section 5.

## 4 New DL Attacks on Round-Reduced Xoodyak

In this section we present the first DL distinguishers on 4- and 5-round Xoodoo [14], and use them to perform key-recovery attacks on 4- and 5-round Xoodyak [15]. We show that the key-recovery attacks are possible only by using the partitioning technique and the neutral bits idea together.

*A Brief Description of Xoodyak.* 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. 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 2). In addition, the $i$'th bit ($0 \leq i < 384$) 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:

$$\begin{array}{ll}
\theta: & P \leftarrow A_0 \oplus A_1 \oplus A_2 \\
 & E \leftarrow P \lll (1,5) \oplus P \lll (1,14) \\
 & A_y \leftarrow A_y \oplus E, y \in \{0,1,2\} \\
\rho_{west}: & A_1 \leftarrow A_1 \lll (1,0) \\
 & A_2 \leftarrow A_2 \lll (0,11) \\
\iota: & A_0 \leftarrow A_0 \oplus C_i \\
\chi: & B_y \leftarrow \overline{A_{y+1 \pmod 3}} \wedge A_{y+2 \pmod 3}, y \in \{0,1,2\} \\
 & A_y \leftarrow A_y \oplus B_y, y \in \{0,1,2\} \\
\rho_{east}: & A_1 \leftarrow A_1 \lll (0,1) \\
 & A_2 \leftarrow A_2 \lll (2,8),
\end{array}$$

[8] This idea was used in [20].

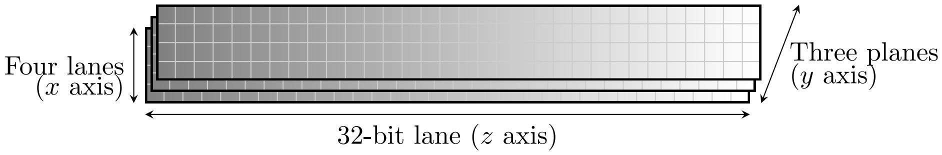


Fig. 2: A Xoodoo state.

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 $\chi$ 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.

### 4.1 4-Round DL Attack on Xoodyak

We now present the first DL distinguisher[9] 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.

**4.1.1 Description of Our Distinguisher.** To choose the input difference we examine the first two steps of the round function: $\theta$ and $\rho_{west}$. We note that given an input difference with two active bits in one column, then $\theta$ does not change the difference, and $\rho_{west}$ shifts each bit by a different number of positions, resulting in two active S-boxes in the S-box layer $\chi$ (the constant addition does not change the difference). For comparison, if the input difference contains only one active bit then after $\theta$, in addition to this active bit, there are three additional active bits at two columns, and $\rho_{west}$ shifts each bit by a different number of positions, resulting in 7 active S-boxes in the first S-box layer. 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 rotated (each word is rotated by the same amount of bits). This input

[9] Liu et al. [24] present a 4-round rotational DL distinguisher, with the highest possible bias of $\frac{1}{2}$, without any attack that uses it. We give in Appendix A the rotational DL distinguisher used by Liu et al. and recall that rotational DL distinguisher is not a DL distinguisher.

difference leads to two active S-boxes before $\chi$: S-box 11 with an input difference of 4 and S-box 32 with an input difference of 2. Denote the output differences (after $\chi$) at S-box 11 by $\Omega_{11}$, and the output differences (after $\chi$) at S-box 32 by $\Omega_{32}$. According to the DDT of $\chi$ 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 $\chi$ layer, and output mask of one or two active bits after 3.5 additional rounds of Xoodoo. The best result was obtained for the output mask[10] $(0, e_{15}, 0)$. The combination $(\Omega_{11}, \Omega_{32}) = (4, 2)$ results in a bias of $+2^{-6}$, where as $(\Omega_{11}, \Omega_{32}) = (4, 6)$ results in a bias of $+2^{-8}$. The 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 $2^{-9.7}$.

### 4.1.2 Attacking 4-Rounds Xoodyak.

We now present an attack, which reveals four key bits of the initialized state. It is should be noted that in this case it is impossible to reveal key bits using a classical DL attack: Assume that we generate about $2^{21.3}$ nonce pairs (this number was calculated according to [31]) with the required input difference, and we compute the number of pairs that are the same on the output mask of the output. Indeed, about $2^{21.3} \cdot \left(\frac{1}{2} + 2^{-9.7}\right)$ pairs are expected to be equal on the output mask, but it does not tell us anything about the key as the question which pairs satisfy the DL characteristic is independent of the key (like in [24]). We show that using neutral bits and the partitioning technique, it is possible to reveal four key bits.

*Finding The Neutral Bits.* We now look for bits of the initial state, and in particular those initialized by the nonce, that *do not* influence the output of the two active S-boxes in the first $\chi$: S-box 11 and S-box 32. Denote the initial state by $S$, and the state just before the S-box layer $\chi$ 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

[10] 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.

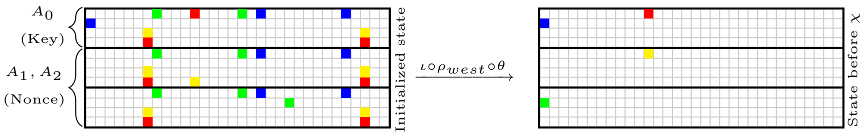


Fig. 3: The non-neutral bits used in the DL attack on 4-round Xoodyak. Each colored bit of the state before $\chi$ is defined as the XOR of the appropriate colored bits of the initialized state.

represented as the XOR of 7 bits of the initial state, as follows (see Figure 3):

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

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 [31] for success rate of 95%), 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. To produce an attack using this characteristic, we need also the partitioning technique.

*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 right 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 $\chi$, the transition $4 \to 4$ occurs when the input values are 2 and 6 and the transition $2 \to 2$ occurs when the input values are 1 and 3. Thus, according to Eq. (1):

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

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

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

where the key bits are indexed by $0 \leq i < 128$ and the nonce bits are indexed by $128 \leq i \leq 383$. It means that there is a partitioning of the space to 16 subsets, depending on four key values: $K_{11} \oplus K_{102} \oplus K_{125}, K_{70} \oplus K_{93}, K_{18} \oplus K_{27} \oplus K_{32}, K_7 \oplus K_{16}$. Each value for these key values 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 without the use of these techniques. The data complexity required to find four key bit is about $2^4 \cdot 2^{13.34} \cdot 2 = 2^{18.34}$ chosen nonces, and the time complexity is about $2^{18.34}$ 4-round Xoodoo calls. We experimentally verified the attack using 100 different keys.[11] 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 $2^{23.34}$ chosen nonces and time complexity of about $2^{23.34}$ 4-round Xoodoo calls.

### 4.2 5-Round Related-Key DL Attack on Xoodyak

We now present the first DL distinguisher on 5-round Xoodoo, 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 one round at the beginning.

**4.2.1 Description of Our Distinguisher.** Similarly to the input difference $(0, e_i, e_i)$ of the 4-round DL characteristic that described in Section 4.1, 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 $\chi$: S-box 0 with an input difference of 1 and S-box 11 with an input difference of 4. Denote the output differences (after $\chi$) at S-box 0 by $\Omega_0$, and the output differences (after $\chi$) at S-box 11 by $\Omega_{11}$. According to the DDT of $\chi$ we have: $\Omega_0 \in \{1, 3, 5, 7\}, \Omega_{11} \in \{4, 5, 6, 7\}$ in a uniform distribution. We experimentally

[11] All the experiments can be found in https://github.com/ArielWeizman/AW/blob/master/Xoodoo.

tested the bias of each DL characteristic with each of the 16 possible differences $(\Omega_0, \Omega_{11})$ after the first $\chi$ 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 $(e_0, 0, 0)$. The combinations $(\Omega_0, \Omega_{11}) \in \{(1,4),(1,6)\}$ result in a bias of $-2^{-3}$, the combinations $(\Omega_0, \Omega_{11}) \in \{(1,5),(1,7),(3,4),(3,6)\}$ result in a bias of $-2^{-5}$, and the combinations $(\Omega_0, \Omega_{11}) \in \{(3,5),(3,7)\}$ result in 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.[12]

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

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

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

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

**4.2.2 Attacking 5-Round Xoodyak.** The 5-round attack is quite similar to the 4-round attack and therefore we give here only a brief description of the attack. The two active S-boxes in the first $\chi$ layer are: S-box 0 with an input difference of 1 and S-box 88 with an input difference of 4. Denote the initial state by $S$ and $T = \iota \circ \rho_{west} \circ \theta(S)$. In these term 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 follows:

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

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

Therefore, we have $256 - 19 = 237$ neutral bits. By fixing all the 19 bits that influence these active S-boxes 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 [31] for a success rate of 95%), each is defined by another fixed value of the non-neutral bits, the good values (i.e., the values which satisfy $(\Omega_0, \Omega_{88}) = (1, 4)$) are expected to appear in about one set, which has the highest bias.

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 $\chi$, the transition $1 \rightarrow 1$ occurs when the input values are 4 and 5 and the transition $4 \rightarrow 4$ occurs when the input values are 2 and 6. Thus, according to Eq. (3):

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

Therefore, we get the following four equations:

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

It means that there is a partitioning of the space to 16 subsets, depending on four key values: $K_{82} \oplus K_{91}, K_{103} \oplus K_{112}, K_{42} \oplus K_{51} \oplus K_{88}, K_{10} \oplus K_{19}$. Each value for these key values 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 1 describes the attack. The data complexity required to reveal four key bits is about $2^4 \cdot 2^{12.04} \cdot 2 = 2^{17.04}$ chosen 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 $2^{22.04}$ chosen nonces and time complexity of about $2^{22.04}$ 5-round Xoodoo encryptions. Table 1 compares previous and ours attacks.

## 5 Improved DL Attacks on Round-Reduced DES

In [8] Biham et al. present two attacks on round-reduced DES. The 8- and 9-round attacks are based on a 7-round DL distinguisher composed of a 4-round differential characteristic and a 3-round linear approximation. We now show how to use the partitioning technique to decrease the attacks' complexity. Then we show an improvement of the 9-round attack using neutral bits. Finally, we show how to combine the partitioning technique and the neutral bits to get the best known attack against 9-round DES.

**Algorithm 1** 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 $keyOptions[k] = \min\{keyOptions[j]\}$.
```

| Attack's Type | Rounds | Complexity |
|---|---|---|
| Zero-sum distinguisher [23] | 12 | $2^{33}$ |
| Differential-linear attack (Sect. 4.1) | 4 | $2^{23.34}$ |
| Differential-linear RK attack (Sect. 4.2) | 5 | $2^{22.04}$ |

Table 1: Comparison of attacks on Xoodyak

*Brief Description of DES [1].* DES is a 64-bit block size, 56-bit key size block cipher, composed of 16 Feistel rounds. Each round is defined by

$$F_{K_r}(x, y) = (y, x \oplus P(S(E(y) \oplus K_r))),$$

where $E : \{0,1\}^{32} \to \{0,1\}^{48}$ is a linear expansion function, $K_r$ is the round key, $S$ is an S-box layer consisting of 8 different S-boxes $S_i : \{0,1\}^6 \to \{0,1\}^4$ $(1 \leq i \leq 8)$ that performed in parallel on 8 different 6-bit parts, and $P : \{0,1\}^{32} \to \{0,1\}^{32}$ is a bit permutation.

### 5.1 Description of Biham et al.'s Attacks [8]

*The 8-Round Attack [8].* To attack 8-round DES Biham et al. use the following distinguisher on 7-round DES and additional round after it covered by guessing some key bits. The distinguisher uses the decomposition of 7-round DES to $E_1 \circ E_0$, where $E_0$ consists of rounds 1–4 and $E_1$ consists of rounds 5–7. $E_0$ is covered by the truncated differential characteristic

$$\Omega_I = 00808200\ 60000000_x \xrightarrow[E_0]{p=\frac{14}{64}} \Omega_M = 00W0XY0Z\ ????U???_x,$$

where $U \in \{0,1,2,\ldots,7\}, W, X \in \{0,8\}, Y, Z \in \{0,2\}$, and ? is an unknown value (see Figure 4a). The linear approximation for $E_1$ is:

$$\lambda_M = 21040080\ 00008000_x \xrightarrow[E_1]{\epsilon=0.195} \lambda_O = \lambda_M.$$

Hence, the bias of the full DL characteristic is about $2p\epsilon^2 = 2^{-5.9}$.

The 8-round attack of [8] is as follows: Ask for the encryption of $2^{12.82}$ plaintext pairs with input difference of $\Omega_I = 00808200\ 60000000_x$, and initialize an array of $2^6$ counters to zeros (corresponding to the six key bits entering to $S_1$ in the last round). Then, for each possible key, compute each ciphertext pair backwards by one round and check if the parity of the resulting pair is equal in the mask $\lambda_O = 21040080\ 00008000_x$. If yes, increment the related entry in the array. The highest entry in the array should correspond to the six key bits. The data and time complexity of this attack both are about[13] $2^{13.82}$.

*The 9-Round Attack [8].* To attack 9-round DES, an additional round preceding the distinguisher is used, covered by guessing additional key material. In order to minimize the number of actives S-boxes in this round, they replace the 4-round truncated differential. The main replacement is in the first round, where the new first round differential is:

$$40000000_x \xrightarrow[F]{p'=\frac{12}{64}} 00000202_x.$$

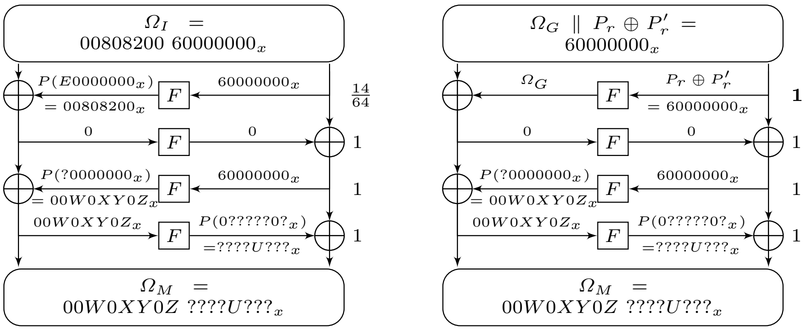
Using this characteristic, there are only two active S-boxes in the first round: $S_6$ and $S_8$. The bias of the new 7-round DL characteristic is thus $2p'\epsilon^2 \approx 2^{-6.13}$.

In [8], $2^7$ structures each consisting of $2^9$ plaintexts are generated as follows:

1. Select a plaintext $P_0$.
2. Select the plaintexts $P_1, \ldots P_{255}$ which differ from $P_0$ by all the possible subsets of the eight bits related to the output of $S_6$ and $S_8$ according to the $P$ permutation of the round function (i.e., the bits masked by $18222828\ 00000000_x$).
3. Select the plaintexts $P_{256}, \ldots P_{511}$ as $P_i = P_{i-256} \oplus 40000000\ 00000202_x$.

Now, for each possible guess of 12 key bits related to $S_6$ and $S_8$ in the first round, find in each structure the $2^8$ appropriate pairs such that the input difference for the second round is $00000202\ 40000000_x$, and perform the 8-round attack on them. The data complexity of this attack is about $2^{16}$ chosen plaintexts. The time complexity is about $2^{16} \cdot 2^{18} \cdot \frac{3}{72} \approx 2^{29.42}$ 9-round DES encryptions (each parity computation takes about 3 S-boxes out of 72 in 9-round DES).

[13] We note that the time complexity is about $2^{12.82} \cdot 2 \cdot 2^6$ one S-box evaluations, which are equivalent to about $2^{13.82}$ 8-round encryptions.



(a) The Differential Characteristic Used in [8].

(b) Our Improved Differential Characteristic.

Fig. 4: Comparison between the differential characteristic used in [8] and ours.

## 5.2 Our Improved Attacks On Round-Reduced DES

**5.2.1 Improved 8-Round Attack, Using Partitioning.** We now revisit the 8-round attack of [8], based on the 4-round truncated differential $\Omega_I = 00808200\ 60000000_x \xrightarrow[E_0]{p=\frac{14}{64}} \Omega_M = ????U???\ 00W0XY0Z_x$. Instead of using standard plaintext pairs with the difference $\Omega_I$, i.e., randomly selected $P$ and $P' = P \oplus \Omega_I$, we propose to use structures in a similar way to that of [20]: Fix two right half values $R, R' = R \oplus 60000000$ ($60000000_x$ is the input difference of $F$ in the first round). Now, generate a structure of $2^4$ left halves $L_0, \ldots, L_{15}$, containing all the possible values in the four bits corresponding to the output of $S_1$ (i.e., bits 9, 17, 23, 31). The structure contains all the $2^5$ plaintexts induced by the two right halves and the $2^4$ left halves. We get that all the plaintext pairs $(P, P')$ (from any structure) have the same input values for $F$ in the first round ($R, R'$, respectively), and thus have the same output difference of $F$ in the first round. Denote this output difference by $\Omega_G$. Obviously, $\Omega_G$ is key dependent.

We now partition the plaintext pairs according to the value of $\Omega_G$ (which has 10 possible values): Each time we concentrate only on plaintext pairs with difference $\Omega_G$ in the left half, i.e., pairs with zero difference at the beginning of the second round with probability of 1. For the correct subset, the probability of the differential characteristic is 1 instead of $\frac{14}{64}$, and the bias of the entire DL characteristic is $2 \cdot 1 \cdot (2 \cdot (-\frac{20}{64})^2)^2 \approx 2^{-3.71}$ instead of $2^{-5.9}$. Figure 4b illustrates the new differential characteristic.

Our attack thus tries all the possible values of $\Omega_G$, and for each such value tries to recover the 6 key bits entering $S_1$ in the last round (which is done as in [8]). Given the correct output difference $\Omega_G$, we can also recover some key material for the first round.

The data complexity of this attack is about $2^{10.14}$ chosen plaintexts. This number was calculated according to [31] for a success rate of 75% (i.e., the right key, $\Omega_G$ combination has the highest bias). Note that since we run over all the 10 possible output differences in the first round and all $2^6$ possible keys, the number of the possible keys in the formula of [31] is replaced by $2^6 \cdot 10 = 640$. The time complexity is mainly affected by checking the parities of the appropriate ciphertexts. Since there are about $2^{10.14}$ ciphertexts and we check it for each six key bits and for each output difference in the first round, the time complexity is about $2^{10.14} \cdot 640 \cdot \frac{1}{64} \approx 2^{13.46}$ 8-round DES encryptions (The parity of each plaintext should be checked about 640 times, each test takes about one S-box computation out of the 64 S-boxes of a full 8-round encryption). This improved attack was experimentally verified using 100 different keys with a success rate of 76%. Table 2 compares the previous attack and ours for a success rate of 75%.[14]

**5.2.2 Improved 9-Round Attack, Using Partitioning.** The idea of running over the output differences in the first round can also be extended to the 9-round attack. Instead of guessing the 12 key bits related to $S_6$ and $S_8$ in the first round, it is possible to partition the plaintext pairs according to the output difference of $S_6$ and $S_8$: Fix two right half values $P_r, P'_r = P_r \oplus 00000202_x$, and generate the structures as described in [8]. The input differences of $S_6$ and $S_8$ (after the expansion permutation of DES) are both 4. The fixing ensures that in all of the pairs, the differences in $S_6$ and $S_8$ are composed from the same values, which ensures that all the output differences are the same. Now, we guess the output differences in $S_6$ and $S_8$, and select the appropriate plaintext pairs from each structure according to that. For input difference of 4 in $S_6$ and $S_8$, there are 9 and 10 possible output differences, respectively. Hence, the plaintext pairs for the attack are selected from 90 possibilities, instead of 4096 as in [8]. The rest of the attack remains the same. Given the right output differences in the first round, we can also recover key material for the first round.

According to [31], since the right key is revealed from only $2^6 \cdot 90 \approx 2^{12.5}$ options (instead of $2^{18}$ in the original attack), about $2^{15.44}$ chosen plaintexts are needed to detect the right key with a success rate of 75%. The time complexity is about $2^{15.44} \cdot 2^6 \cdot 90 \cdot \frac{1}{72} \approx 2^{21.76}$ 9-round DES encryptions. This improved attack was experimentally verified using 100 different keys with a success rate of 78%.

A subtle point should be noted: In contrast to the analysis in Section 3.1 which claims for reducing the data complexity using the partitioning technique, in our case the main effect of this idea is reducing the time complexity. The reason is that Biham et al. [8] already used structures to ensure that enough pairs with the required input difference exist in the data, and the key guessing is needed to identify the structure. This idea affects the data complexity in the same way as the partitioning technique. However, there is a subtle difference between their structures and the partitioning technique: their structures heavily depend on the specific structure of DES, while the partitioning technique is more general. To emphasize this difference between the two ideas, consider a DES variant for which the $P$-permutation is an 8-bit key dependent permutation. For this variant, the structures of Biham et al. [8] are of size $2^{32}$ to ensure that indeed

[14] All the experiments can be found in https://github.com/ArielWeizman/AW/blob/master/DES.

all the required pairs exist in the data, since the positions of the relevant bits after the $P$-permutation are unknown. However, most of these plaintexts are not going to be used in the attack. In comparison, the partitioning technique would define a structure of $2^{15.44}$ plaintexts for each of the $2^8$ possible $P$-permutations, i.e., a total of $2^{23.44}$ plaintexts.

**5.2.3 Improved 9-Round Attack, Using Neutral Bits.** We now show how to use neutral bits in order to minimize the attack's complexity. The goal is to find a subspace $\mathcal{U} \subseteq \mathbb{F}_2^{64}$ such that if a pair $(P, P')$ satisfies the differential characteristic, than $\forall u \in \mathcal{U} : (P \oplus u, P' \oplus u)$ also satisfies the differential characteristic. Recall that the differential characteristic covers rounds 2–5 (the first round is covered by guessing key material). The first round of the differential characteristic is

$$40000000_x \xrightarrow[F]{\frac{12}{64}} P(30000000_x) = 00000202_x$$

(the rest of the characteristic has probability 1). In the left half, all the bits that are masked by 07fffffe$_x$ do not affect the input values of $S_1$ in the second round, and thus they are neutral bits. In the right half, the six bits that affect the input values of $S_1$ in the second round come from six different S-boxes $(S_2, S_4, S_5, S_6, S_7, S_8)$, thus the bits that are masked by $60600000_x$ are also neutral bits. Note that since all these neutral bits do not affect $S_1$ in the second round, every linear combination of them is also neutral (unlike in the case of [7]). Thus, all these neutral bits are a basis for the desired neutral subspace $\mathcal{U} \subseteq \mathbb{F}_2^{64}$ with $2^{30}$ neutral vectors. Now, the bits that are masked by $18222828_x$ in the left half can be used to define the structures used in the attack. Each structure defines 4 different pairs entering $S_1$ in the second round, i.e., 4 subsets of $2^6$ pairs each (since there are six bits that are masked by $00222828_x$), such that either all of them satisfy the differential characteristic, or no one satisfies it. Using the other 24 neutral bits, it is possible to increase the size of these subsets up to $2^{30}$ right pairs.[15]

We now compute the data complexity. Since each structure (which is defined according to a fixed value) generates four subsets, and for each subset the probability of being a right subset is $\frac{12}{64}$, the probability that all of the four subsets are wrong is $\left(1 - \frac{12}{64}\right)^4$. Thus, the probability of having at least one right subset, given $s$ structures, is $1 - \left(1 - \frac{12}{64}\right)^{4s}$. Therefore, using two structures we get a probability of about 0.81 to have a right subset. Hence, for a success rate of 75% of the entire attack, a success rate in each subset is needed to be about $\frac{0.75}{0.81} \approx 0.93$. Now, given such a subset of pairs such that all of them satisfy the differential characteristic, the bias of the DL approximation is about $2^{-3.71}$, as before. According to [31], about $2^{10.57}$ pairs are needed to detect the 18 key bits with success rate of 93%. Therefore, increasing each structure by

[15] For attacks that needed more plaintext pairs, we refer the reader to the chosen plaintext linear cryptanalysis techniques suggested Knudsen and Mathiassen [20].

| Rounds | | [8][17] | Partitioning | Neutral Bits | Combination |
|---|---|---|---|---|---|
| 8 | Data (CP) | $2^{13.82}$ | $2^{10.14}$ | – | – |
| | Time (Enc.) | $2^{13.82}$ | $2^{13.46}$ | – | – |
| | Recovered Key Bits | 6 | 6 | – | – |
| 9 | Data (CP) | $2^{16}$ | $2^{15.44}$ | $2^{14.57}$ | $2^{14.1}$ |
| | Time (Enc.) | $2^{29.42}$ | $2^{21.76}$ | $2^{28}$ | $2^{22}$ |
| | Recovered Key Bits | 18 | 6 | 18 | 6 |

Table 2: Comparison of DL attacks on round-reduced DES.

$2^{4.57}$ plaintexts, using the neutral bits, is sufficient. Thus the data complexity is about $2^{13.57} \cdot 2 = 2^{14.57}$ chosen plaintexts. The time complexity is about $2^{14.57} \cdot 2^{18} \cdot \frac{3}{72} \approx 2^{28}$ 9-round DES encryptions. This improved attack was experimentally verified using 100 different keys with a success rate of 79%.

**5.2.4 Improved 9-Round Attack, Using a Combination of Partitioning and Neutral Bits.** We note that the main effect of the partitioning technique is on the time complexity, and that of the neutral bits is on the data complexity. Therefore, combining these two techniques together can reduce both. To do that, we fix two right half values $P_r, P_r' = P_r \oplus 00000202_x$, generate the structures as described in [8], and partition the data according to the output difference of $S_6$ and $S_8$ as described in Section 5.2.2. As described in Section 5.2.3, each structure defines 4 subsets of $2^6$ pairs that satisfy together the differential characteristic. To increase the size of these subsets, note that since the right half is fixed in all of the inputs, the neutral bits are only those from the left half that are masked by 07fffffe$_x$. However, the resulting neutral subspace is big enough for the attack. Algorithm 2 describes the improved attack.

Since the right key is detected from only $2^6 \cdot 90$ options (instead of $2^{18}$ in Section 5.2.3), about $2^{10.1}$ pairs are needed in each subset to detect the right key with a success rate of 93% (which leads to a success rate of 75% of the entire attack). Thus, the data complexity is about $2^{13.1} \cdot 2 = 2^{14.1}$ chosen plaintexts. The time complexity is about $2^{14.1} \cdot 2^6 \cdot 90 \cdot \frac{3}{72} \approx 2^{22}$ 9-round DES encryptions. This improved attack was experimentally verified using 100 different keys with a success rate of 77%. Table 2 compares the previous attacks and ours for a success rate of 75%.

[16] Since each plaintext pair passes the differential characteristic has zero difference in the bits masked by $\lambda_M$ (i.e., $(P \oplus P') \cdot \lambda_M = \Omega_M \cdot \lambda_M = 0$), the sign of the bias is necessarily positive.

[17] The values given here are calculated according to [31] for success rate of 75%, which differs a bit from [8].

**Algorithm 2** Improved Attack on 9-Round DES (Recovering 6 key bits and the output difference $\Omega_G$).

```
Fix a random right half value $R$.
Set[16] $max\ counter = 0$.
for all iteration $\in \{0,1\}$ do
    Fix a random value for all left half non-neutral bits $L_{\text{fixed}}$.
    Generate $2^{4.1}$ structures $S_i$ as follows:
    for all Structure $S_i$ do
        Select A random left half value for all neutral bits $L^i_{\text{rand}}$.
        Set $L^i_0 = L_{\text{fixed}} \oplus L^i_{\text{rand}}, P^i_0 = L^i_0 \parallel R$.
        Select the plaintexts $P^i_1, \ldots, P^i_{255}$ which differ from $P^i_0$ by all the 255 possible subsets of the eight bits masked by 18222828 00000000$_x$.
        Set $P^i_j = P^i_{j-256} \oplus$ 40000000 00000202$_x, \forall 256 \le j \le 511$.
    end for
    Request the ciphertexts of these plaintexts.
    for all $K_1 \in \{0,1\}^6$ (The subkey entering $S_1$ in the last round) do
        for all $\Omega_G$ (The output difference of $F$ in the first round, of two plaintexts $P_i, P_j$, such that $0 \le i \le 255, 256 \le j \le 511$). do
            Select the pairs with difference $\Omega_G$ in the left half.
            Partially decrypt all these pairs through S-box $S_1$ of the last round.
            Check how many ciphertext pairs are equal in the parity of the five bits masked by 21040080 00008000$_x$ and denote this number by $c$.
            if $c > max\ counter$ then
                Set $max\ counter = c, K_{\max} = K_1, \Omega_{\max} = \Omega_G$.
            end if
        end for
    end for
end for
Output $K_{\max}, \Omega_{\max}$.
```

## 6 Conclusions

In this paper we discussed the possibility of combining two techniques to improve DL attacks: The partitioning technique, which is used to find a subset of the data in which the bias of the entire distinguisher is higher than using random data; and the use of neutral bits, which is used to create many right pairs given one right pair.

When using neutral bits, two issues should be taken in account: First, the probability of each neutral bit: The basic definition of a neutral bit [7] is: if $(P, P')$ is a right pair then $(P \oplus e_i, P' \oplus e_i)$ is also a right pair with probability of 1. But, in practice, it is possible to use almost neutral bits, which satisfy the condition with probability of $p < 1$ [4]. Second, the transition from $t$ neutral bits to $2^t$ neutral vectors: Although all the vectors in the linear span of some neutral vectors are expected to be neutral, this is not always the case [7]. We note that for the three ciphers discussed in this paper are based on S-boxes, and in such ciphers all the neutral bits satisfy with probability of 1, and the transition to linear subspace works as expected.

We also point out a case in which a link between the partitioning and the neutral bits ideas allow us to perform the first DL attack on 5-round Xoodyak. We also showed that when these two techniques are performed on different parts of the DL characteristic, then it is possible to combine them to achieve the best results. We applied combinations of them to improve two DL attacks, on 9-round DES, and we achieve the best DL attack, significantly improving previous DL results.

## References

1. Data Encryption Standard, Federal Information Processing Standards publications no. 46, 1977.
2. Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba. In *proceedings of FSE 2008, LNCS 5086, pp. 470–488, Springer.*
3. Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. DLCT: A New Tool for Differential-Linear Cryptanalysis. In *proceedings of EUROCRYPT 2019, LNCS 11476, pp. 313–342, Springer.*
4. Christof Beierle, Gregor Leander, and Yosuke Todo. Improved Differential-Linear Attacks with Applications to ARX Ciphers. In *proceedings of CRYPTO 2020, LNCS 12172, pp. 329–358, Springer.*
5. Daniel J. Bernstein. The Salsa20 Family of Stream Ciphers. In *New Stream Cipher Designs - The eSTREAM Finalists, 2008, LNCS 4986, pp. 84–97, Springer.*
6. Eli Biham and Yaniv Carmeli. An Improvement of Linear Cryptanalysis with Addition Operations with Applications to FEAL-8X. In *proceedings of SAC 2014, LNCS 8781, pp. 59–76, Springer.*
7. Eli Biham and Rafi Chen. Near-Collisions of SHA-0. In *proceedings of CRYPTO 2004, LNCS 3152, pp. 390–305, Springer.*
8. Eli Biham, Orr Dunkelman, and Nathan Keller. Enhancing Differential-Linear Cryptanalysis. In *proceedings of ASIACRYPT 2002, LNCS 2501, pp. 254–266, Springer.*
9. Eli Biham, Orr Dunkelman, and Nathan Keller. The Rectangle Attack - Rectangling the Serpent. In *proceedings of EUROCRYPT 2001, LNCS 2045, pp. 340–357, Springer.*
10. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.
11. Céline Blondeau, Gregor Leander, and Kaisa Nyberg. Differential-Linear Cryptanalysis Revisited. *J. Cryptology*, 30(3):859–888, 2017.
12. Céline Blondeau and Kaisa Nyberg. New Links between Differential and Linear Cryptanalysis. In *proceedings of EUROCRYPT 2013, LNCS 7881, pp. 388–404, Springer.*
13. Florent Chabaud and Serge Vaudenay. Links Between Differential and Linear Cryptanalysis. In *proceedings of EUROCRYPT 1994, LNCS 950, pp. 356–365, Springer.*
14. 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.

15. 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.

16. Joan Daemen and Vincent Rijmen. The Wide Trail Design Strategy. In *proceedings of Cryptography and Coding, IMA International Conference, 2001, LNCS 2260, pp. 222–238, Springer.*

17. Sabyasachi Dey, Hirendra Kumar Garai, Santanu Sarkar, and Nitin Kumar Sharma. Revamped Differential-Linear Cryptanalysis on Reduced Round ChaCha. In *proceedings of EUROCRYPT 2022, LNCS 13277, pp. 86–114, Springer.*

18. 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.

19. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In *proceedings of FSE 2000, LNCS 1978, pp. 75–93, Springer.*

20. Lars R. Knudsen and John Erik Mathiassen. A Chosen-Plaintext Linear Attack on DES. In *proceedings of FSE 2000, LNCS 1978, pp. 262–272, Springer.*

21. Susan K. Langford and Martin E. Hellman. Differential-Linear Cryptanalysis. In *proceedings of CRYPTO 1994, LNCS 839, pp. 17–25, Springer.*

22. Gaëtan Leurent. Improved Differential-Linear Cryptanalysis of 7-Round Chaskey with Partitioning. In *proceedings of EUROCRYPT 2016, LNCS 9665, pp. 344–371, Springer.*

23. Fukang Liu, Takanori Isobe, Willi Meier, and Zhonghao Yang. Algebraic Attacks on Round-Reduced Keccak/Xoodoo. *IACR Cryptol. ePrint Arch.*, page 346, 2020.

24. Yunwen Liu, Siwei Sun, and Chao Li. Rotational Cryptanalysis from a Differential-Linear Perspective - Practical Distinguishers for Round-Reduced FRIET, Xoodoo, and Alzette. In *proceedings of EUROCRYPT 2021, LNCS 12696, pp. 741–770, Springer.*

25. Zhiqiang Liu, Dawu Gu, Jing Zhang, and Wei Li. Differential-Multiple Linear Cryptanalysis. In *proceedings of Inscrypt 2009, LNCS 6151, pp. 35–49, Springer.*

26. Jiqiang Lu. A methodology for differential-linear cryptanalysis and its applications. *Des. Codes Cryptography*, 77(1):11–48, 2015.

27. Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In *proceedings of EUROCRYPT 1993, LNCS 765, pp. 386–397, Springer.*

28. Shoji Miyaguchi. The FEAL Cipher Family. In *proceedings of CRYPTO 1990, LNCS 537, pp. 627–638, Springer.*

29. Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers. In *proceedings of SAC 2014, LNCS 8781, pp. 306–323, Springer.*

30. Kaisa Nyberg and Lars R. Knudsen. Provable Security Against Differential Cryptanalysis. In *proceedings of CRYPTO 1992, LNCS 740, pp. 566–574, Springer.*

31. Ali Aydin Selçuk. On Probability of Success in Linear and Differential Cryptanalysis. *J. Cryptology*, 21(1):131–147, 2008.

32. David A. Wagner. The Boomerang Attack. In *proceedings of FSE 1999, LNCS 1636, pp. 156–170, Springer.*

# A A Rotational DL Distinguisher On 4-Round Xoodoo [24]

In [24] 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 amount 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 in an all-zero difference and left-rotate amount of one bit. For the first round constant of 4-round Xoodoo $C = 00000480_x$, the required input difference is $\Omega_I = (\Omega A_0, \Omega A_1, \Omega A_2)$ where

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

Therefore, given a plaintext pair $(P, P')$ and their ciphertext pair (after 4-round Xoodoo) $(C, C')$ we have:

$$Pr\left[\lambda \cdot (C \lll 1) = \lambda \cdot C' \mid P' = (P \lll 1) \oplus \Omega_I\right] = 1.$$