New Approaches for Estimating the Bias of Differential-Linear Distinguishers (Full Version)

Ting Peng[1,2], Wentao Zhang[1,2]✉, Jingsui Weng[1,2], and Tianyou Ding[1,2]

[1] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, CAS
[2] School of Cyber Security, University of Chinese Academy of Sciences
{pengting, zhangwentao, wengjingsui, dingtianyou}@iie.ac.cn

Abstract. Differential-linear cryptanalysis was introduced by Langford and Hellman in 1994 and has been extensively studied since then. In 2019, Bar-On et al. presented the Differential-Linear Connectivity Table (DLCT), which connects the differential part and the linear part, thus an attacked cipher is divided to 3 subciphers: the differential part, the DLCT part, and the linear part.
In this paper, we firstly present an accurate mathematical formula which establishes a relation between differential-linear and truncated differential cryptanalysis. Using the formula, the bias estimate of a differential-linear distinguisher can be converted to the probability calculations of a series of truncated differentials. Then, we propose a novel and natural concept, the TDT, which can be used to accelerate the calculation of the probabilities of truncated differentials. Based on the formula and the TDT, we propose two novel approaches for estimating the bias of a differential-linear distinguisher. We demonstrate the accuracy and efficiency of our new approaches by applying them to 5 symmetric-key primitives: Ascon, Serpent, KNOT, AES, and CLEFIA. For Ascon and Serpent, we update the best known differential-linear distinguishers. For KNOT AES, and CLEFIA, for the first time we give the theoretical differential-linear biases for different rounds.

Keywords: Differential-linear cryptanalysis · Truncated cryptanalysis · SPN ciphers · TDT · Ascon · Serpent · KNOT · AES · CLEFIA.

1 Introduction

1.1 Background and Previous Work

Differential cryptanalysis and linear cryptanalysis are the two best-known cryptanalysis techniques for symmetric-key primitives. The basic idea of differential cryptanalysis, introduced by Biham and Shamir [1,2], is to study the development of differences of the cipher between two plaintexts through the encryption process, such that high-probability differentials are obtained with as many rounds as possible. Linear cryptanalysis proposed by Matsui [3] studies the development of parities of plaintext bits through the encryption process, such that we can obtain high-bias linear approximations for as many rounds as possible.

Differential and linear cryptanalysis has been used to analyze numerous symmetric-key primitives, and resistance to the two cryptanalysis techniques has become a central criterion in the design of symmetric-key primitives. While precluding long differentials and linear approximations are sufficient for making a cipher immune to the two attacks, it turned out that in many cases, short differential characteristics and/or linear approximations can be combined and exploited to break the cipher. In 1994, Langford and Hellman [4] introduced a new cryptanalyis technique called *differential-linear cryptanalysis* (in short: *DL cryptanalysis*), a cipher E is decomposed as a cascade $E = E_1 \circ E_0$, then a high-probability differential for E_0 and a high-bias linear approximation for E_1 can be combined into an efficient distinguisher for the entire cipher E.

In 2002, Biham et al. [5] extended and improved DL cryptanalysis to obtain wider scope of applications. In 2017, Blondeau et al. [7] presented a formal treatment of the DL cryptanalysis, based on a general link between differential and linear cryptanalysis introduced by Chabaud and Vaudenay [9] and developed by Blondeau and Nyberg [8]. The formal treatment provides an exact expression for the differential-linear bias under the sole assumption that the two parts of the cipher are independent.

In EUROCRYPT 2019, Bar-On et al. [10] introduced a new tool: the *differential-linear connectivity table* (DLCT), which takes into account the dependency between the differential part and the linear part. They decomposed a cipher E into $E = E_1 \circ E_m \circ E_0$, where E_0 is covered by a differential, E_m is covered by a DLCT, E_1 is covered by a linear approximation. They also demonstrated that the DLCT can be constructed efficiently using the Fast Fourier Transform. Then, they showed that the DLCT can be utilized to improve the differential-linear cryptanalysis; taking Serpent, Ascon, and ICEPOLE for examples, they found better distinguishers than previous known ones for Serpent and Ascon, and improved the DL attacks for ICEPOLE.

Recently, in CRYPTO 2021, Liu et al. [24] studied differential-linear cryptanalysis from an algebraic perspective by introducing a technique called *Differential Algebraic Transitional Form* (DATF). Based on the DATF algorithm, they developed a new theory of estimating of the differential-linear bias and techniques for key recovery in differential-linear cryptanalysis. As a result, they improved the DL attacks on reduced-round Grain v1, Ascon and Serpent.

In ASIACRYPT 2023, Hu et al. [34] revisit HD/HDL cryptanalysis from an algebraic perspective and provide two novel tools for detecting possible HD/HDL distinguishers, including: (a) Higher-order Algebraic Transitional Form (HATF) for probabilistic HD/HDL attacks; (b) Differential Supporting Function (DSF) for deterministic HD attacks. They found that HATF works well for DL (1^{st}-order HDL) attacks and some well-known DL biases of Ascon and XOODYAK.

Table 1: The differential-linear bias

Cipher	Rounds	Experimental value	Theoretical estimate				
			DLCT [10]	DATF [24]	HATF [34]	Approach in Sect.4.2	Approach in Sect.4.3
Ascon	4/12	2^{-2}[16]	2^{-5}	$2^{-2.365}$	$2^{-2.09}$	$\mathbf{2^{-2}}$	
	5/12	2^{-10}[16]					$\mathbf{2^{-10.1}}$
	6/12‡						$\mathbf{2^{-22.43}}$
Serpent	**3/32**†	$\mathbf{2^{-1.415}}$				$\mathbf{2^{-1.415}}$	
	4/32	$2^{-13.75}$[14]	$2^{-13.68}$	$2^{-13.736}$		$\mathbf{2^{-13.696}}$	
	4/32†	$\mathbf{2^{-5.30}}$				$\mathbf{2^{-5.415}}$	
	5/32	$2^{-17.75}$[14]		$2^{-17.736}$		$\mathbf{2^{-17.696}}$	
	5/32†	$\mathbf{2^{-11.44}}$				$\mathbf{2^{-11.415}}$	
	6/32†					$\mathbf{2^{-19.61}}$	
	7/32†					$\mathbf{2^{-29.45}}$	
	8/32†					$\mathbf{2^{-39.45}}$	
	9/32		$2^{-57.68}$	$2^{-57.736}$		$\mathbf{2^{-57.696}}$	
	9/32†					$\mathbf{2^{-52}}$	
	9/32†					$\mathbf{2^{-55.33}}$	
KNOT256	9/52	$\mathbf{2^{-1.20}}$				$\mathbf{2^{-1.20}}$	
	10/52	$\mathbf{2^{-3.27}}$				$\mathbf{2^{-3.66}}$	
	11/52	$\mathbf{2^{-4.31}}$				$\mathbf{2^{-6.38}}$	
	12/52	$\mathbf{2^{-9.91}}$				$\mathbf{2^{-9.27}}$	
	13/52	$\mathbf{2^{-14.04}}$				$\mathbf{2^{-12.27}}$	
	14/52					$\mathbf{2^{-16.23}}$	
	15/52					$\mathbf{2^{-23.31}}$	
	16/52					$\mathbf{2^{-30.52}}$	
AES	2/10	$\mathbf{2^{-1}}$				$\mathbf{2^{-1}}$	
	3/10	$\mathbf{2^{-8.66}}$				$\mathbf{2^{-8.66}}$	
	4/10					$\mathbf{2^{-27.85}}$	
	5/10					$\mathbf{2^{-51.85}}$	
CLEFIA	4/18	$\mathbf{2^{-1}}$				$\mathbf{2^{-1}}$	
	5/18	$\mathbf{2^{-2.54}}$				$\mathbf{2^{-2.54}}$	
	6/18	$\mathbf{2^{-7.54}}$				$\mathbf{2^{-7.54}}$	
	7/18					$\mathbf{2^{-12.37}}$	
	8/18					$\mathbf{2^{-33.59}}$	
	9/18					$\mathbf{2^{-55.84}}$	

Entry with "‡" is a longer DL distinguisher of Ascon found in this paper.
Entries with "†" are better DL distinguishers of Serpent found in this paper.
In bold type are the results from our paper.

Remark 1. All experimental value in this paper are obtained by encrypting 2^{30} plaintext pairs with specific input differences, and checking the output values with respect to whether the parity of the output subset is the same or not.

1.2 Our contributions

A formula, establishing a link between the bias of a differential-linear distinguisher and the probabilities of a series of truncated differentials.*(see Formula (4) in Sect.3)* The authors wrote in [7]: 'The differential-linear attack can be, in the theoretical sense, considered either as a multidimensional linear or a truncated differential attack' and equation (5) in [7] also reflects this conversion, however, their idea stops proceeding and remained only in theoretical sense. We emphasize that we find this link independently, and more important, we will show in the following how to utilize this link to construct effective/better differential-linear distinguishers for practical symmetric-key primitives. Specifically, we select 5 symmetric-key primitives —- the LWC winner Ascon, the AES finalist Serpent, the LWC candidate Knot, the AES block cipher and the CLEFIA block cipher to demonstrate the power of our new approaches.

The TDT. We propose a novel and natural concept, *the Truncated Difference Distribution Table* (TDT for short), which fully characterizes truncated differences of an S-box or a truncated differential over multiple rounds, in the case of only one input difference, see Sect.3.2 for details. It can be seen later in Sect.4 and Sect.5 that the TDT can be utilized to accelerate the calculation of truncated differential probabilities greatly.

Two new approaches for estimating the differential-linear bias. Based on the relation between differential-linear and truncated differential cryptanalysis (i.e., Formula (4)) and the TDT, We propose two new approaches for evaluating the differential-linear bias. The first approach, introduced in Sect.4.2, can be used to compute the differential-linear bias when E_m can be covered by multiple rounds. The second approach, introduced in Sect.4.3, can be used to compute the differential-linear bias when E_m consists of only one round.

To demonstrate the accuracy and efficiency of our new approaches, we apply them to 5 symmetric-key ciphers, see details in Sect.5. In the following, we briefly summarizes our results on the 5 primitives respectively, and Table.1 presents the differential-linear biases for reduced-round versions of the 5 ciphers.

Applications to Ascon. In [16], the authors stated that, the overall bias of a 4-round DL distinguisher is expected to be 2^{-20} by the theory, while the experiments showed that the bias is significantly higher: 2^{-2}. They also presented a 5-round distinguisher with an experimental bias of 2^{-10}.

In this paper, we revisit the 4-round and 5-round distinguishers in [16]. Using our first approach in Sect.4.2, we obtain that the bias of the 4-round distinguisher is 2^{-2}, which is in accordance with the experimental result in [16] and better than

the results in [10] and in [24] (2^{-5} in [10] and $2^{-2.365}$ in [24] and $2^{-2.09}$ in [34]). Moreover, using our second approach in Sect.4.3, we estimate that the bias of the 5-round distinguisher is $2^{-10.1}$, which is extremely close to the experimental result in [16]. Although in [24], the authors presented a 5-round DL distinguisher with a bias of $2^{-5.415}$, however, it needs to be imposed 9 conditions, while we do not impose any condition on the DL distinguishers presented in this paper. In [34], the authors gave the theoretical bias for a 5-round DL distinguisher: the bias is estimated as 2^{-10} while the experimental value is 2^{-9}.

Furthermore, we construct a 6-round DL distinguisher, by trying many of possible combinations of truncated differentials and DLCTs using the approach in Sect.4.3. As a result, we obtain a 6-round DL distinguisher with a bias of $2^{-22.43}$, which is the highest number of rounds for Ascon with respect to differential-linear distinguishers.

Applications to Serpent. In [13], the authors presented a 9-round DL distinguisher with a bias of 2^{-60} starting with round 2, and then in [14], the authors concluded that the actual bias of the 9-round distinguisher is $2^{-57.75}$ instead of 2^{-60} by checking the first 4 rounds of the distinguisher in [13] experimentally, where the bias of the first 4-round DL distinguisher is $2^{-13.75}$. In [10], the authors revisited the 4-round variant in [13] by the DLCT and concluded that the bias of the 4-round distinguisher examined in [13] is $2^{-13.68}$. In [24], the authors revisited the 4-round DL distinguisher in [13] by the DATF and concluded that the bias is $2^{-13.736}$, they also applied their methods to check the first 5 rounds of the 9-round DL distinguisher in [13], and obtained a bias of $2^{-17.736}$.

In this paper, we firstly apply our approach introduced in Sect.4.2 to estimate the bias of the first 4-round DL distinguisher in [13], we concluded that the theoretical bias is $2^{-13.696}$, which is very close to the experimental result in [14]. Next, we estimate the bias of the first 5 rounds of the 9-round DL distinguisher in [13], and obtained a bias of $2^{-17.696}$. Furthermore, we search for the DL distinguishers up to 9 rounds. Ignoring the key recovery attack, we can get DL distinguishers with a bias of 2^{-52}. If we take the key recovery attack into consideration, two better 9-round DL distinguishers are achieved by trying a lot of combinations of different differentials and linear approximations, the biases of our new distinguishers are both $2^{-55.33}$, compared to the previous bias $2^{-57.75}$, which is a factor of $2^{2.42}$ improvement.

Applications to KNOT. We focus on analysis for the initialization phase of KNOT-AEAD(128,256,64) (in short: KNOT256). In [28], the authors presented key-recovery attacks for reduced-round KNOT256 based on conditional differential-linear distinguishers. For 15-round KNOT256, it needs $2^{48.8}$ time complexity and $2^{47.5}$ blocks to recover the full 128-bit key.

In our paper, we begin by carrying out experiments to validate the accuracy of the bias estimate in Sect.5.3. Then, we investigate differential-linear distinguishers using the approach in Sect.4.2. Considering the properties of the initialization phase, the input difference is limited on the nonce, and the output mask is limited on the rate. As a result, we obtain differential-linear distinguishers for

KNOT256 up to 16 rounds, where the bias of a 16-round DL distinguisher is $2^{-30.52}$. The 16-round DL distinguisher is the best distinguisher of KNOT256, with respect to the number of rounds; moreover, it is possible to present better DL attacks on KNOT256 with respect to the highest number of attacked rounds [35], compared to that in [28].

Applications to AES. In this paper, we detect many highly biased DL distinguishers for round-reduced AES. We first exhaust all possible 3-round DL distinguishers where the number of active S-boxes in the first round and the third round is both 1, which helps us obtain a highly biased 3-round DL distinguisher.

For the 4-round and 5-round AES's DL distinguisher, we search a 3-round DL distinguisher where the number of active S-box in the first round is 1, and the number of active S-box in the third round is 4. A 4-round DL distinguisher is obtained by appending a 1-round linear approximation after the 3-round DL distinguisher. After that, if we perform forward extension on the 4-round DL distinguisher by adding a one round differential trail, a 5-round DL distinguisher is generated. The biases are $2^{-8.66}/2^{-27.85}/2^{-51.85}$ for 3/4/5-round DL distinguisher, respectively.

Applications to CLEFIA. For CLEFIA, when the number of rounds is less than or equal to 4, the optimal DL distinguisher has the bias of 2^{-1}. For the 5-round CLEFIA's DL distinguisher, we search for numerous combinations of the input difference and the output mask and obtain many highly biased DL distinguisher. And then, we perform forward extension and backward extension based on these 5-round DL distinguisher. As a result, we first give DL distinguishers up to 9 rounds, where the 9-round DL distinguisher has the bias of $2^{-55.84}$.

Remark 2. Significantly, we cannot give any lower or upper bound for DL biases in theory. And these results do not threaten the security of the target ciphers.

2 Preliminaries

2.1 Notations

In this paper, we focus on SPN symmetric-key primitives, and aim to estimate the bias of differential-linear distinguishers. We assume that our target ciphers are Markov ciphers [37], that means the cipher rounds are both differentially and linear round independent. The following notations are used in this paper:

Notation	Description
$\mathcal{S}$, $\mathcal{L}$	the non-linear layer, the linear layer respectively in SPN cipher
$E = E_1 \circ E_m \circ E_0$	the encryption of a block cipher

R_0, R_m, R_1	the number of rounds for E_0, E_m, E_1 respectively
R	the round function of E
$n = s \times \ell$	n is the size of state the state is separated into ℓ S-boxes of s bits each
$X[i]$	the i-th S-box of a state X , e.g., a state X with ℓ S-boxes $X = X[\ell-1] \parallel \cdots \parallel X[0]$
$X^{\{i\}}$	the i-th bit of a bit string X, e.g., a n-bit string $X = X^{\{n-1\}} \parallel \cdots \parallel X^{\{0\}}$
$P[n][k]$	the look-up table of the linear layer, e.g., $Y^{\{i\}} = X^{\{P[i][0]\}} \oplus \cdots \oplus X^{\{P[i][k-1]\}}$
ΔX	the difference of X and X'
$\mathcal{T}$	a set of differences
$\lvert\mathcal{T}\rvert$	the size of the set $\mathcal{T}$
ΛX_i, ΛY_i	the state before and after $\mathcal{S}$ layer in the i-th layer in the forward propagation
∇X_i, ∇Y_i	the state before and after $\mathcal{S}$ layer in the i-th layer in the backward propagation
$(\mathcal{TD}_0, \cdots, \mathcal{TD}_{t-1})$	the truncated differential pattern
$hw(X)$	the Hamming weight of X
$X \parallel Y$	bit string concatenation of X and Y

2.2 The Differential-Linear Attack

The classical differential-linear attack [4] was proposed by Langford and Hellman in 1994, which consists of two stages. The first stage is covered by differential cryptanalysis and ensures the propagation of a useful differential property in the middle of the cipher. The second stage is covered by linear cryptanalysis and ensures the propagation of a useful linear property from the middle of the cipher to the end.

Let E be a cipher which is decomposed into a cascade $E = E_1 \circ E_0$. Assuming that we have a differential $\Delta_I \xrightarrow{p} \Delta_O$, i.e., an input difference Δ_I leads to an output difference Δ_O with probability p in E_0, and a linear approximation $\lambda_I \xrightarrow{q} \lambda_O$ with probability $1/2 + q$ (or with bias q), i.e., an input mask λ_I leads to an output mask λ_O with bias q in E_1. Under the assumptions that E_0 and E_1 are independent, and that the cipher behaves randomly if the differential is not satisfied, the probability $\Pr[C \cdot \lambda_O = C' \cdot \lambda_O | P \oplus P' = \Delta_I]$ that a plaintext pair (P, P') with input difference Δ_I satisfies $C \cdot \lambda_O = C' \cdot \lambda_O$, where (C, C') is the ciphertext pair corresponding to the plaintext pair, is

$$\Pr[C \cdot \lambda_O = C' \cdot \lambda_O | P \oplus P' = \Delta_I] = p(1/2 + 2q^2) + (1-p) \cdot 1/2 = 1/2 + 2pq^2 \quad (1)$$

Hence, if p, q are sufficiently large, then the adversary can distinguish E from a random permutation using $O(p^{-2}q^{-4})$ chosen plaintexts.

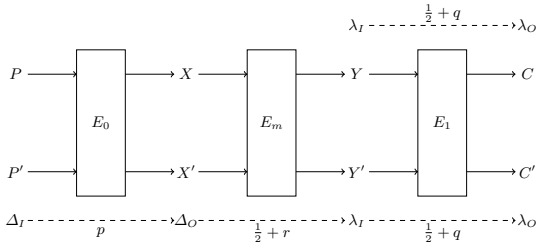


Fig. 1: The differential-linear cryptanalysis

In 2017, Blondeau et al. [7] studied the differential-linear cryptanalysis more accurately. They pointed out that the basic idea of differential-linear cryptanalysis is to split the cipher under consideration into two parts. The split should be such that, for the first part of the cipher there exists a strong truncated differential and for the second part there exists a strongly biased linear approximation, which enlightens our approach in Sect.4.3.

In order to (partially) take the effects of dependency into account, in EUROCRYPT 2019, Bar-On et al. [10] introduced a new tool: the *differential-linear connectivity table* (DLCT). The decomposition $E = E_1 \circ E_0$ used in the standard DL attack was replaced by the decomposition $E = E_1 \circ E_m \circ E_0$.

As illustrated in Fig.1, let P and P' denote a pair of plaintexts, C and C' denote their ciphertexts, X and X' denote the intermediate values between E_0 and E_m, Y and Y' denote the intermediate values between E_m and E_1, respectively, where E_0 is covered by a differential $\Delta_I \xrightarrow{p} \Delta_O$ with differential probability $\Pr[X \oplus X' = \Delta_O | P \oplus P' = \Delta_I] = p$, E_1 is covered by a linear approximation $\lambda_I \xrightarrow{q} \lambda_O$ with $\Pr[C \cdot \lambda_O = Y \cdot \lambda_I] = \frac{1}{2} + q$, E_m is covered by a DLCT $\Delta_O \xrightarrow{r} \lambda_I$ with $\Pr[\lambda_I \cdot E_m(X) = \lambda_I \cdot E_m(X') | X \oplus X' = \Delta_O] = \frac{1}{2} + r$. Under the assumption of independence between the subciphers, adapting the naive analysis of the DL attack complexity presented above (i.e., Eq. (1)), they [10] obtained

$$\mathcal{E}_{\Delta_I,\lambda_O} = 4p \cdot \overline{DLCT}_{E_m}(\Delta, \lambda) \cdot q^2 = 4prq^2 \tag{2}$$

and the data complexity is $O(p^{-2}r^{-2}q^{-4})$. In order to adapt the exact analysis of [7], a bit more computation is needed. The exact bias of the DL distinguisher is

$$\mathcal{E}_{\Delta_I,\lambda_O} = \sum_{\Delta,\lambda} \Pr[\Delta_I \xrightarrow{E_0'} \Delta] \cdot \overline{DLCT}_{E_m}(\Delta, \lambda)(c^{E_1'}_{\lambda,\lambda_O})^2 \tag{3}$$

With the DLCT, the authors further improved the differential-linear attacks on ICPOLE, Serpent, Ascon and 8-round DES.

In CRYPTO 2021, Liu et al. [24] studied the differential-linear cryptanalysis from an algebraic perspective by introducing a technique called DATF. They presented a new theory of bias estimate and techniques for key-recovery attacks. The techniques were applied to Ascon, Serpent and Grain v1.

Based on the HD/HDL cryptanalysis, Hu et al. [34] provided novel methods to study HD and HDL cryptanalysis, where one is the Higher-order Algebraic

Transitional Form (HATF), which is used to detect probabilistic HDL approximations. The other is the Differential Supporting Function (DSF), which is useful to find deterministic HD distinguishers. And then, these two methods were applied to Ascon and XOODYAK.

2.3 Definitions

Definition 1. *Let $f: \{0,1\}^n \to \{0,1\}^n$ be a bijective vectorial boolean function. The DDT is a $2^n \times 2^n$ table whose rows correspond to input differences of f and whose columns correspond to output differences of f. Formally, for $\Delta_I \in \{0,1\}^n$ and $\Delta_O \in \{0,1\}^n$, the DDT entry (Δ_I, Δ_O) is*

$$\mathrm{DDT}_f(\Delta_I, \Delta_O) \triangleq |\{X | f(X) \oplus f(X \oplus \Delta_I) = \Delta_O\}|$$

The probability of the differential $\Delta_I \to \Delta_O$ is defined by

$$\overline{\mathrm{DDT}}_f(\Delta_I, \Delta_O) = \frac{\mathrm{DDT}_f(\Delta_I, \Delta_O)}{2^n}$$

Typically, truncation refers to a shortening of an input, but for the purposes of differential cryptanalysis truncation refers to a relaxation in the specifications of a differential.

Definition 2. *[29] Using the symbol $*$ denotes an unknown value and, for a n-bit string $U = U^{\{n-1\}} \parallel \cdots \parallel U^{\{0\}}$, define*

$$V^{\{n-1\}} \parallel \cdots \parallel V^{\{0\}} \in \mathrm{TRUNC}(U^{\{n-1\}} \parallel \cdots \parallel U^{\{0\}})$$

if, and only if, $V^{\{i\}} = U^{\{i\}}$ or $V^{\{i\}} = *$ for all $0 \leq i < n$. This notion extends naturally to differences. If there is an t-round differential characteristic

$$\Delta_0 \xrightarrow{R} \Delta_1 \xrightarrow{R} \cdots \xrightarrow{R} \Delta_t$$

then

$$\mathcal{T}_0 \xrightarrow{R} \mathcal{T}_1 \xrightarrow{R} \cdots \xrightarrow{R} \mathcal{T}_t$$

is a truncated differential characteristic if $\mathcal{T}_i \in \mathrm{TRUNC}(\Delta_i)$ for $0 \leq i \leq t$.

The probability of a truncated differential $(\mathcal{T}_0 \xrightarrow{t \text{ rounds}} \mathcal{T}_t)$ is defined by

$$\Pr[\mathcal{T}_0 \xrightarrow{t \text{ rounds}} \mathcal{T}_t] = \frac{1}{|\mathcal{T}_0|} \sum_{\Delta_0 \in \mathcal{T}_0} \Pr[(f(X) \oplus f(X \oplus \Delta_0)) \in \mathcal{T}_t]$$

In [10], the authors present the definition of the Differential-Linear Connectivity Table (DLCT) which takes into account the dependency between differential part and linear part.

Definition 3. *[10] Let $f: \{0,1\}^n \to \{0,1\}^n$ be a bijective vectorial boolean function. The DLCT of f is a $2^n \times 2^n$ table whose rows correspond to input differences to f and whose columns correspond to bit masks of output of f. Formally, $\Delta \in \{0,1\}^n$ and $\lambda \in \{0,1\}^n$, the DLCT entry (Δ, λ) is*

$$\mathrm{DLCT}_f(\Delta, \lambda) \triangleq |\{X | \lambda \cdot f(X) = \lambda \cdot f(X \oplus \Delta)\}| - 2^{n-1}$$

For DLCT, sometimes it will be more convenient to use the normalized DLCT entry

$$\overline{\mathrm{DLCT}}_f(\Delta,\lambda)=\frac{\mathrm{DLCT}_f(\Delta,\lambda)}{2^n}$$

The differential-linear probability (DLP) of a DL distinguisher is defined by

$$\Pr[\Delta\to\lambda]=\frac{|\{X|\lambda\cdot(f(X)\oplus f(X\oplus\Delta))=0\}|}{2^n}$$

In this paper, we use the bias ε to measure the unbalancedness. The differential-linear bias is defined as $\varepsilon=\Pr[\Delta\to\lambda]-\frac{1}{2}$.

3 The Truncated Difference Distribution Table and its Relation with the DLP

In this section, we present an important observation on the DLP and then introduce a novel concept: *the Truncated Difference Distribution Table* (TDT for short), which can be utilized to calculate the probability of a truncated differential characteristic much more efficiently. Furthermore, we show that how to calculate the differential-linear bias round by round using the TDT.

3.1 An Important Observation on DLP

In [7], there is a observation that any differential-linear relation can be regarded as a truncated differential. However, their idea stops deriving a general and remained in theoretical sense. In this paper, we thoroughly study that converts a differential-linear relation by a series of truncated differentials, so that we can calculate differential-linear bias using automated algorithm.

In the following, we will show that how to convert differential-linear probability (DLP) to the sum of multiple truncated differential probabilities. Suppose that there are t rounds, the calculation of DLP is as follows,

$$\begin{aligned}\Pr[\Delta\to\lambda]&=\frac{|\{X|\lambda\cdot(f(X)\oplus f(X\oplus\Delta))=0\}|}{2^n}\\&=\frac{\sum_{\Delta_i\in\mathbb{F}_2^n,\lambda\cdot\Delta_i=0}|\{X|f(X)\oplus f(X\oplus\Delta)=\Delta_i\}|}{2^n}\\&=\sum_{\Delta_i\in\mathbb{F}_2^n,\lambda\cdot\Delta_i=0}\overline{\mathrm{DDT}}_f(\Delta,\Delta_i)\end{aligned}\tag{4}$$

Note that the cases satisfying $\overline{\mathrm{DDT}}_f(\Delta,\Delta_i)=0$ are not excluded in the above formula, just for ease of simple expression. From Formula (4), we can see that the estimate of DLP can be converted into probability calculations of a series of differential probabilities $\overline{\mathrm{DDT}}_f(\Delta,\Delta_i)$.

In Formula (4), it needs to judge whether $\lambda\cdot\Delta_i=0$ for each Δ_i $(0\le i<2^n)$ satisfying $\overline{\mathrm{DDT}}_f(\Delta,\Delta_i)\neq 0$. Let us write $\lambda\cdot\Delta_i$ using their bits $\{\lambda^j\}$ and $\{\Delta_i^j\}$, i.e.,

$$\lambda\cdot\Delta_i=\lambda^{\{n-1\}}\Delta_i^{\{n-1\}}\oplus\lambda^{\{n-2\}}\Delta_i^{\{n-2\}}\oplus\cdots\lambda^{\{1\}}\Delta_i^{\{1\}}\oplus\lambda^{\{0\}}\Delta_i^{\{0\}}\tag{5}$$

then we have another crucial observation. In Eq.(5), if $\lambda^{\{j\}} = 0$, then $\lambda^{\{j\}}\Delta_i^{\{j\}} = 0$ always holds, which means that the value of this bit of Δ_i does not affect the value of $\lambda \cdot \Delta_i$. Hence, if $\lambda^{\{j\}} = 0$, the value of $\Delta_i^{\{j\}}$ can be either 0 or 1; and if $\lambda^{\{j\}} = 1$, the value of $\Delta_i^{\{j\}}$ must be determined to be 0 or 1.

Define the product of the symbol $*$ and the bit 0 to be 0, then we can extend the inner product of two binary vectors to the inner product of one binary vector and one truncated difference, which will be used in the following.

By Definition 2, the set of the 2^n possible output differences $\{\Delta_i\}(\Delta_i \in \mathbb{F}_2^n)$ can be divided into $2^{hw(\lambda)}$ subsets. The subsets $\mathcal{T}_{t,0}, \ldots, \mathcal{T}_{t,2^{hw(\lambda)}-1}$ are distinct from each other. For each subset $\mathcal{T}_{t,j}$, every element $\Delta_k \in \mathcal{T}_{t,j}$ has the same value of $\lambda \& \Delta_k$. We can deduce that each subset corresponds to a truncated differential $\mathcal{T}_0 = \Delta \xrightarrow{t\ rounds} \mathcal{T}_{t,j}$ $(0 \le j < 2^{hw(\lambda)})$. Thus, we have

$$\Pr[\Delta \to \lambda] = \sum_{\Delta_i \in \mathbb{F}_2^n, \lambda \cdot \Delta_i = 0} \overline{\mathrm{DDT}}_f(\Delta, \Delta_i) = \sum_{\substack{0 \le j < 2^{hw(\lambda)} \\ \lambda \cdot \mathcal{T}_{t,j} = 0}} \Pr[\Delta \xrightarrow{t\ rounds} \mathcal{T}_{t,j}] \quad (6)$$

Note that, among all the $\mathcal{T}_{t,j}(0 \le j < 2^{hw(\lambda)})$, there are a half (i.e. $2^{hw(\lambda)-1}$) satisfying that $\lambda \cdot \mathcal{T}_{t,j} = 0$. Hence, we convert the estimate of DLP to the probability calculations of $2^{hw(\lambda)-1}$ truncated differentials. We will come back to this problem in Sect.3.4 after we introduce a new concept, i.e., the TDT table.

3.2 The Truncated Difference Distribution Table

To efficiently compute the probability of a truncated differential, we introduce a new concept: the Truncated Difference Distribution Table (TDT for short).

Since each bit in a truncated difference $\mathcal{T}$ has three possibilities: $*$, 0, and 1, we need to firstly introduce a concept of Truncated Differential Mask (abbreviated as TD mask). For a given truncated difference $\mathcal{T}$, define its TD mask $\mathcal{M}$ as follows

$$\mathcal{M}^{\{i\}} = \begin{cases} 0, & if\ \mathcal{T}^{\{i\}} = * \\ 1, & if\ \mathcal{T}^{\{i\}} = 0\ or\ 1 \end{cases}$$

Similarly, define the & operation of the bit 0 and the symbol $*$ to be 0, then we can extend the & operation of two binary vectors to the & operation of one binary vector and one truncated difference. Notice that $\mathcal{M}\&\mathcal{T}$ and $\mathcal{T}$ have the same values in the bit positions where $\mathcal{T}$ has the values of 0 or 1. Therefore, the combination of $\mathcal{M}$ and $\mathcal{M}\&\mathcal{T}$ fully determines the truncated difference $\mathcal{T}$.

Let $\mathcal{Z} = \mathcal{M}\&\mathcal{T}$, that is to say

$$\mathcal{T}^{\{i\}} = \begin{cases} *, & if\ \mathcal{M}^{\{i\}} = 0 \\ 0, & if\ \mathcal{M}^{\{i\}} = 1\ and\ \mathcal{Z}^{\{i\}} = 0 \\ 1, & if\ \mathcal{M}^{\{i\}} = 1\ and\ \mathcal{Z}^{\{i\}} = 1 \end{cases}$$

Now we are ready to present the definition of TDT.

Definition 4. *Let $f: \{0,1\}^n \to \{0,1\}^n$ be a bijective vectorial boolean function. the TDT of f is a three-dimensional table whose first parameter $\Delta_I \in \{0,1\}^n$ is an input difference of f, and whose second parameter $\mathcal{M} \in \{0,1\}^n$ is the TD mask of a truncated output difference $\mathcal{T} \in \{*,0,1\}^n$ of f and whose third parameter is $\mathcal{Z} \in \{0,1\}^n$. Define the TDT entry $(\Delta_I, \mathcal{M}, \mathcal{Z})$ as*

$$\mathrm{TDT}_f(\Delta_I, \mathcal{M}, \mathcal{Z}) = |\{X | \mathcal{M}\&(f(X) \oplus f(X \oplus \Delta_I)) = \mathcal{Z}\}|$$

where the TDT entry is equal to zero if $\mathcal{Z}$ has one-bits outside the coverage of $\mathcal{M}$.

Similarly, define the probability of the TDT entry $(\Delta_I, \mathcal{M}, \mathcal{Z})$ as

$$\overline{\mathrm{TDT}}_f(\Delta_I, \mathcal{M}, \mathcal{Z}) = \frac{\mathrm{TDT}_f(\Delta_I, \mathcal{M}, \mathcal{Z})}{2^n}.$$

Proposition 1. *The TDT is an extension of the DDT. There is a connection between DDT and TDT:*

$$\mathrm{TDT}_f(\Delta_I, \mathcal{M}, \mathcal{Z}) = \sum_{\Delta : \mathcal{M}\&\Delta = \mathcal{Z}} \mathrm{DDT}_f(\Delta_I, \Delta)$$

Obviously, $\mathrm{TDT}_f(\Delta, \mathcal{M}, \mathcal{Z}) \leq 2^n$. The $\overline{\mathrm{TDT}}_f(\Delta_I, \mathcal{M}, \mathcal{Z})$ corresponds to the probability of the truncated differential $\Delta_I \to \mathcal{T}$, where $\mathcal{T}$ is uniquely determined by $\mathcal{M}$ and $\mathcal{Z}$.

Proposition 2. *Let $f: \{0,1\}^n \to \{0,1\}^n$ be a bijective vectorial boolean function, Δ and λ denote an input difference and an output mask of f respectively.*

$$\overline{\mathrm{DLCT}}_f(\Delta, \lambda) = \Pr[\Delta \to \lambda] - \frac{1}{2} = \sum_{\substack{0 \leq j < 2^{hw(\lambda)} \\ \lambda \cdot \mathcal{Z}_j = 0}} \overline{\mathrm{TDT}}_f(\Delta, \lambda, \mathcal{Z}_j) - \frac{1}{2}$$

The TDT table can help us eliminate numerous unnecessary computations. For example, for a 4-bit S-box S, given the input difference $\Delta_I = 0011$, we want to calculate the probability of the truncated differential $0011 \xrightarrow{S} 1{*}{*}{*}$. Using the DDT, 8 queries must be performed, the probability is calculated as $\sum_{i=8}^{15} \overline{\mathrm{DDT}}_S(0011, i)$; as a comparison, notice that the TD mask of $1{*}{*}{*}$ is $\mathcal{M} = 1000$, and $\mathcal{M}\&(1{*}{*}{*}) = 1000$, hence, using the TDT, only one query is needed, the probability is obtained by just checking $\overline{\mathrm{TDT}}_S(0011, 1000, 1000)$.

In Table 3, we present a part of the TDT for KNOT's S-box S, with $\Delta_I = 0001$. Consider $\mathcal{M} = 0001$, which means that the least significant bit of the truncated output difference is 0 or 1, while the other 3 bits are all $*$. Thus, in row $\mathcal{M} = 0001$ of Table 3, there are at most two non-zero entries, which lie in column 0000 and column 0001. From Table 3, we can see that $\mathrm{TDT}_S(0001, 0001, 0000) = 8$, which means the probability of $0001 \xrightarrow{S} {*}{*}{*}0$ is $\frac{1}{2}$; and $\mathrm{TDT}_S(0001, 0001, 0001) = 8$, which means the probability of $0001 \xrightarrow{S} {*}{*}{*}1$ is also $\frac{1}{2}$. Next, consider row $\mathcal{M} = 0100$, there is only one non-zero entry $\mathrm{TDT}_S(0001, 0100, 0100) = 16$, which means that the probability of $0001 \xrightarrow{S} {*}1{*}{*}$ is 1.

In Sect.4, we will present more details on how to compute the probability of a truncated differential over multiple rounds using the TDT.

Table 3: The TDT of KNOT's S-box with $\Delta_I = 0001$

$\mathcal{M}$ \ $\mathcal{Z}$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010	8	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
0011	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0100	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	8	8	0	0	0	0	0	0	0	0	0	0
0110	0	0	0	0	8	0	8	0	0	0	0	0	0	0	0	0
0111	0	0	0	0	4	4	4	4	0	0	0	0	0	0	0	0
1000	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0
1001	4	4	0	0	0	0	0	0	4	4	0	0	0	0	0	0
1010	4	0	4	0	0	0	0	0	4	0	4	0	0	0	0	0
1011	2	2	2	2	0	0	0	0	2	2	2	2	0	0	0	0
1100	0	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0
1101	0	0	0	0	4	4	0	0	0	0	0	0	4	4	0	0
1110	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4	0
1111	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2

3.3 Properties of the TDT

In this subsection, we present 4 properties of the TDT.

Property 1. For any $\mathcal{M} \in \{0,1\}^n$,

$$\mathrm{TDT}_f(0, \mathcal{M}, \mathcal{Z}) = \begin{cases} 2^n, & if\ \mathcal{Z} = 0 \\ 0, & if\ \mathcal{Z} \neq 0 \end{cases}$$

If $\Delta_I = 0$, then we have $\mathrm{TDT}_f(0, \mathcal{M}, 0) = 2^n$ for any $\mathcal{M}$. Indeed, if two inputs to f are equal, then the corresponding two outputs are also equal. This means that when computing the truncated differential probability for an S-box, if the input difference $\Delta_I = 0$, then the truncated difference $\mathcal{T} = 0$ with probability 1.

Property 2.

$$\mathrm{TDT}_f(\Delta_I, 0, \mathcal{Z}) = \begin{cases} 2^n, & if\ \mathcal{Z} = 0 \\ 0, & if\ \mathcal{Z} \neq 0 \end{cases}$$

If $\mathcal{M} = 0$, then we have $\mathrm{TDT}_f(\Delta_I, 0, 0) = 2^n$ for any Δ_I. Indeed, if $\mathcal{M} = 0$, then all bits of the truncated output difference are $*$. This means that when computing the truncated differential probability for an S-box, if the TD mask $\mathcal{M} = 0$, then all compatible output differences are allowed, and the truncated differential probability is 1.

Property 3.

$$\mathrm{TDT}_f(\Delta_I, 2^n - 1, \mathcal{Z}) = \mathrm{DDT}_f(\Delta_I, \mathcal{Z})$$

This can be evidently inferred from Proposition 1. If $\mathcal{M} = 2^n - 1$, then the corresponding entries in TDT are identical to those in the DDT.

Property 4. Given Δ_I and $\mathcal{M}$, there are at most $2^{hw(\mathcal{M})}$ non-zero entries in the TDT. $\mathcal{M}$ indicates which bits of the truncated output difference are marked as

$*$ and which bits are marked as 0 or 1, indeed, there are $hw(\mathcal{M})$ bits needed to be determined to be 0 or 1. Thus, given Δ_I and $\mathcal{M}$, there are at most $2^{hw(\mathcal{M})}$ possible truncated output differences. For instance, in Table 3, if $hw(\mathcal{M}) = 2$, there are at most 4 non-zero entries. This could involve the case $\mathcal{M} = 0011$, which has 4 non-zero entries, or $\mathcal{M} = 1100$, which has only two non-zero entries.

3.4 Estimation of the DLP based on TDT

Suppose that an t-round cipher $E : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is represented as the following composition,

$$E = E_{t-1} \circ E_{t-2} \circ \cdots \circ E_0$$

Since we assume that E is a Markov cipher, that means the probability of a truncated differential characteristic is often computed by multiplying the probabilities round by round. Then we know

$$\Pr[\mathcal{T}_0 \xrightarrow{R} \mathcal{T}_1 \xrightarrow{R} \cdots \xrightarrow{R} \mathcal{T}_t] = \prod_{i=0}^{t-1} \Pr[\mathcal{T}_i \xrightarrow{R} \mathcal{T}_{i+1}] \tag{7}$$

$$\Pr[\mathcal{T}_0 \xrightarrow{t\ rounds} \mathcal{T}_t] = \sum_{\mathcal{T}_1,\cdots,\mathcal{T}_{t-1}} \prod_{i=0}^{t-1} \Pr[\mathcal{T}_i \xrightarrow{R} \mathcal{T}_{i+1}] \tag{8}$$

The truncated probability of the i-th round is

$$\begin{aligned}
\Pr[\mathcal{T}_i \xrightarrow{R} \mathcal{T}_{i+1}] &= \frac{1}{|\mathcal{T}_i|} \cdot \sum_{\Delta_i \in \mathcal{T}_i} \sum_{\Delta_{i+1} \in \mathcal{T}_{i+1}} \Pr[\Delta_i \xrightarrow{R} \Delta_{i+1}] \\
&= \frac{1}{|\mathcal{T}_i|} \cdot \sum_{\Delta_i \in \mathcal{T}_i} \Pr[\Delta_i \xrightarrow{R} \mathcal{T}_{i+1}] \\
&= \frac{1}{|\mathcal{T}_i|} \cdot \sum_{\Delta_i \in \mathcal{T}_i} \Pr[\Delta_i \xrightarrow{\mathcal{S}} \hat{\mathcal{T}}_i \xrightarrow{\mathcal{L}} \mathcal{T}_{i+1}]
\end{aligned}$$

where $\mathcal{T}_{i+1} = \mathcal{L}(\hat{\mathcal{T}}_i)$ with $1 \le i < t$. Since the linear layer do not affect the calculation of probability, we have

$$\begin{aligned}
\Pr[\mathcal{T}_i \xrightarrow{R} \mathcal{T}_{i+1}] &= \frac{1}{|\mathcal{T}_i|} \cdot \sum_{\Delta_i \in \mathcal{T}_i} \Pr[\Delta_i \xrightarrow{\mathcal{S}} \hat{\mathcal{T}}_i] \\
&= \frac{1}{|\mathcal{T}_i|} \cdot \sum_{\Delta_i \in \mathcal{T}_i} \prod_{j=0}^{\ell-1} \Pr[\Delta_i[j] \xrightarrow{S} \hat{\mathcal{T}}_i[j]] \\
&= \prod_{j=0}^{\ell-1} \Pr[\mathcal{T}_i[j] \xrightarrow{S} \hat{\mathcal{T}}_i[j]]
\end{aligned} \tag{9}$$

This means

$$\Pr[\mathcal{T}_0 \xrightarrow{R} \mathcal{T}_1 \xrightarrow{R} \cdots \xrightarrow{R} \mathcal{T}_t] = \prod_{i=0}^{t-1} \prod_{j=0}^{\ell-1} \Pr[\mathcal{T}_i[j] \xrightarrow{S} \hat{\mathcal{T}}_i[j]] \tag{10}$$

$$\Pr[\mathcal{T}_0 \xrightarrow{t\ rounds} \mathcal{T}_t] = \sum_{\mathcal{T}_1,\cdots,\mathcal{T}_{t-1}} \prod_{i=0}^{t-1}\prod_{j=0}^{\ell-1} \Pr[\mathcal{T}_i[j] \xrightarrow{S} \hat{\mathcal{T}}_i[j]] \tag{11}$$

Next, we will demonstrate the methodology behind utilizing TDT for computing the probability of a truncated differential characteristic. Since the input difference of the TDT is a determined input difference, in this paper, we limit that every $\mathcal{T}_i[j]$ is either a determined difference (i.e. $|\mathcal{T}_i[j]| = 1$) or a truncated difference but all differential bits in $\hat{\mathcal{T}}_i[j]$ being $*$ (i.e. $|\mathcal{T}_i[j]| > 1$ and $|\hat{\mathcal{T}}_i[j]| = 2^s$). The two cases are listed as follows,

1. When $|\mathcal{T}_i[j]| = 1$, $\Pr[\mathcal{T}_i[j] \xrightarrow{S} \hat{\mathcal{T}}_i[j]] = \overline{\mathrm{TDT}}(\mathcal{T}_i[j], \mathcal{M}_i[j], \mathcal{Z}_i[j])$, where $\hat{\mathcal{T}}_i[j]$ is uniquely determined by $\mathcal{M}_i[j]$ and $\mathcal{Z}_i[j]$,
2. When $|\mathcal{T}_i[j]| > 1$ and $|\hat{\mathcal{T}}_i[j]| = 2^s$, $\Pr[\mathcal{T}_i[j] \xrightarrow{S} \hat{\mathcal{T}}_i[j]] = 1$.

Therefore, the probabilities of the truncated differential characteristic and the the truncated differential can be calculated by

$$\Pr[\mathcal{T}_0 \xrightarrow{R} \mathcal{T}_1 \xrightarrow{R} \cdots \xrightarrow{R} \mathcal{T}_t] = \prod_{i=0}^{t-1}\prod_{|\mathcal{T}_i[j]|=1} \overline{\mathrm{TDT}}(\mathcal{T}_i[j], \mathcal{M}_i[j], \mathcal{Z}_i[j]) \tag{12}$$

$$\Pr[\mathcal{T}_0 \xrightarrow{t\ rounds} \mathcal{T}_t] = \sum_{\mathcal{T}_1,\cdots,\mathcal{T}_{t-1}} \prod_{i=0}^{t-1}\prod_{|\mathcal{T}_i[j]|=1} \overline{\mathrm{TDT}}(\mathcal{T}_i[j], \mathcal{M}_i[j], \mathcal{Z}_i[j]) \tag{13}$$

Consequently, we can calculate the truncated differential probability round by round using the TDT. However, we face the difficulty that how to determine all possible values of $\mathcal{T}_i$. To overcome the problem, we precompute the TD mask $\mathcal{M}_i$ according to some known information, specifically, the input difference and the output mask, as described in this paper. Subsequently, we exhaust every potential $\mathcal{Z}_i$ to generate all possible $\mathcal{T}_i$.

Furthermore, using the TDT, we can calculate the differential-linear probability. According to Formula (6), we have

$$\begin{aligned}\Pr[\Delta \to \lambda] &= \sum_{\substack{0\le k<2^{hw(\lambda)}\\ \overline{\lambda}\cdot\mathcal{T}_{t,k}=0}} \Pr[\Delta \xrightarrow{t\ rounds} \mathcal{T}_{t,k}] \\ &= \sum_{\substack{0\le k<2^{hw(\lambda)}\\ \overline{\lambda}\cdot\mathcal{T}_{t,k}=0}} \sum_{\mathcal{T}_1,\cdots,\mathcal{T}_{t-1}} \prod_{i=0}^{t-1}\prod_{|\mathcal{T}_i[j]|=1} \overline{\mathrm{TDT}}(\mathcal{T}_i[j], \mathcal{M}_i[j], \mathcal{Z}_i[j])\end{aligned} \tag{14}$$

where $\mathcal{T}_0 = \Delta$.

Finally, the differential-linear bias is equal to $\Pr[\Delta \to \lambda] - \frac{1}{2}$.

4 Computing the Differential-Linear Bias

In this section, we propose two new approaches to evaluate the bias of a differential-linear distinguisher. The first approach concerns situations where E_m contains many rounds. The second approach is focused on scenarios where E_m contains only one round.

4.1 Computing the Truncated Differential Pattern given an Input Difference and an Output Mask

In this part, we introduce the concept of the truncated differential pattern (TD pattern for short) $(\mathcal{TD}_0, \cdots, \mathcal{TD}_{t-1})$, which assists in ascertaining whether a S-box in the output difference of the $\mathcal{S}$ layer between rounds is ignorable or not. For each $\mathcal{TD}_i$, we have

$$\mathcal{TD}_i[j] = \begin{cases} 0, & if\ \hat{\mathcal{T}}_i[j] = * \ or\ \hat{\mathcal{T}}_i[j] = 0 \\ \mathcal{M}_i[j], & otherwise. \end{cases} \tag{15}$$

where $\mathcal{T}_{i+1} = \mathcal{L}(\hat{\mathcal{T}}_i)$, $*$ is the s-bit vector with all elements being $*$. This means we can eliminate some unnecessary S-box that satisfies $\mathcal{TD}_i[j] = 0$.

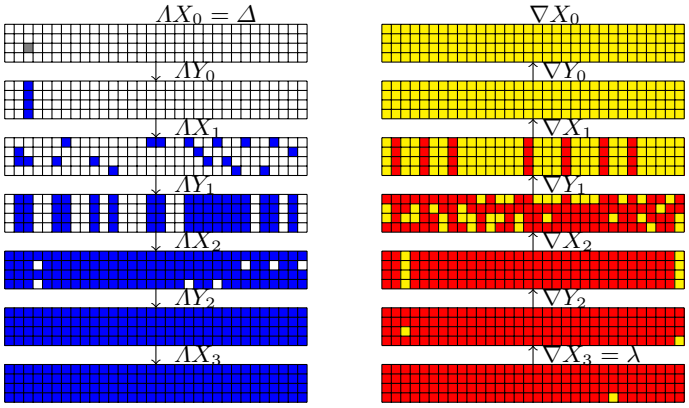
Given the number of rounds t, an input difference Δ, and an output mask λ, we demonstrate how to generate the TD pattern $(\mathcal{TD}_0, \cdots, \mathcal{TD}_{t-1})$. For the input difference, a forward propagation is performed. For the output mask, a backward propagation is performed. A t-round TD pattern is then obtained by combining these two propagations.

We use a 3-round variant of Serpent that starts at round 3 as an example. Given the input difference Δ and the output mask λ, the forward propagation and the backward propagation are shown in Fig.2.

Forward propagation of the input difference. First, we focus on the forward propagation of the given input difference Δ. For the $\mathcal{S}$ layer, if the input difference of the S-box is 0, the output difference of the S-box must be 0. If the input difference of an S-box is not 0 or has any undetermined bit, we set all bit differences of the output difference to be undetermined. For the $\mathcal{L}$ layer, if a bit difference is active or undetermined, all the output bit differences associated with it are set to be undetermined. The forward propagation of Δ of the 3-round Serpent is depicted in Fig.2(a).

Backward propagation of the output mask. Given the output mask λ, we study the backward propagation along the decryption direction. For the $\mathcal{L}^{-1}$ layer, if an output bit difference needs to be determined, all the input bit differences associated with it also need to be determined. For the $\mathcal{S}^{-1}$ layer, there are two cases for each S-box regarding the output difference: One case is that all output bit differences are arbitrary, which means that all input bit differences are arbitrary. The other case is that if there is any output bit difference that must be determined or is a fixed bit, all input bit differences must be determined. The backward propagation of λ of the 3-round Serpent is depicted in Fig.2(b).

Computing the mask pattern. In the estimate process of the differential-linear bias, the complexity increases exponentially with the number of rounds, whether it is the input difference's forward propagation or the output mask's backward propagation. To improve this process, we examine the possibility of utilizing both the forward and backward propagation concurrently. It is worth noting that every bit difference has three scenarios:



(a) The forward propagation of Δ: a blank cell indicates a bit of which the bit difference is always inactive; a gray cell indicates an active bit difference; a blue cell indicates a bit of which the bit difference is undetermined

(b) The backward propagation of λ: a yellow cell indicates a bit of which the bit difference need to be determined; a red cell indicates a bit of which the bit difference is arbitrary

Fig. 2: The propagations of 3-round Serpent

1. If a bit difference is inactive in the forward propagation (the blank cell in Fig.2(a)), e.g., $\Lambda Y_i^{\{j\}} = 0$, it will always be inactive, and we can ignore it.
2. If a bit difference is arbitrary in the backward propagation (the red cell in Fig.2(b)), e.g., $\nabla Y_i^{\{j\}} = 0$, there is no need to consider its value, so we can also ignore it.
3. If a bit difference need to be determined in both forward and backward propagation simultaneously (the blue cell in Fig.2(a) and the yellow cell Fig.2(b)), e.g., $\Lambda Y_i^{\{j\}} = 1$ and $\nabla Y_i^{\{j\}} = 1$, it means we must determine its difference in that bit position.

As a result, we can ignore the first 2 cases and only focus on case 3.

$\mathcal{TD}_i$ depends on ΛY_i and ∇Y_i. For each $\mathcal{TD}_i$, there is $\mathcal{TD}_i^{\{j\}} = 0$ when the bit difference satisfies case 1 or case 2, which indicates its irrelevance in the later program, alternatively there is $\mathcal{TD}_i^{\{j\}} = 1$ when the bit satisfies case 3, leading us to traverse its bit differences. The TD pattern $(\mathcal{TD}_0, \mathcal{TD}_1, \mathcal{TD}_2)$ of the 3-round Serpent is shown in Fig.3.

The pseudocode of calculating the TD pattern is showed in Algorithm 1.

Algorithm 1 Our Search Approach for Computing TD Pattern $(\mathcal{TD}_0, \mathcal{TD}_1, \cdots, \mathcal{TD}_{t-1})$

Input: input difference Δ, output mask λ, the number of rounds t
Output: TD pattern $(\mathcal{TD}_0, \mathcal{TD}_1, \cdots, \mathcal{TD}_{t-1})$

function FPropagation
  Let $\Lambda X_0 = \Delta$
  for $0 \le i < t$ **do**
    for $0 \le j < \ell$ **do**
      if $\Lambda X_i[j] \neq 0$ **then**
        $\Lambda Y_i[j] = 2^s - 1$
      end if
    end for
    ΛX_{i+1}=$\mathcal{L}$-LAYER(ΔY_i)
  end for
  return $(\Lambda Y_0, \cdots, \Lambda Y_{t-1})$
end function

function BPropagation
  Let $\nabla X_t = \lambda$
  for $t \ge i > 0$ **do**
    ∇Y_{i-1}=$\mathcal{L}^{-1}$-LAYER(∇X_i)
    for $0 \le j < \ell$ **do**
      if $\nabla Y_{i-1}[j] \neq 0$ **then**
        $\nabla X_{i-1}[j] = 2^s - 1$
      end if
    end for
  end for
  return $(\nabla Y_0, \cdots, \nabla Y_{t-1})$
end function

function TDPattern
  for $0 \le i < t$ **do**
    $\mathcal{TD}_i = \Lambda Y_i \& \nabla Y_i$
  end for
  return $(\mathcal{TD}_0, \mathcal{TD}_1, \cdots, \mathcal{TD}_{t-1})$
end function

function $\mathcal{L}$-layer
  Let $\Lambda X_{i+1} = 0$
  for $0 \le j < n$ **do**
    for $0 \le l < k$ **do**
      if $\Lambda Y_i^{\{P[j][l]\}} = 1$ **then**
        $\Lambda X_{i+1}^{\{j\}} = 1$
      end if
    end for
  end for
  return ΛX_{i+1}
end function

function $\mathcal{L}^{-1}$-layer
  Let $\nabla Y_{i-1} = 0$
  for $j = 0$ to $n-1$ **do**
    if $\nabla X_i^{\{j\}} = 1$ **then**
      for $0 \le l < k$ **do**
        $\nabla Y_{i-1}^{\{P[j][l]\}} = 1$
      end for
    end if
  end for
  return ∇Y_{i-1}
end function

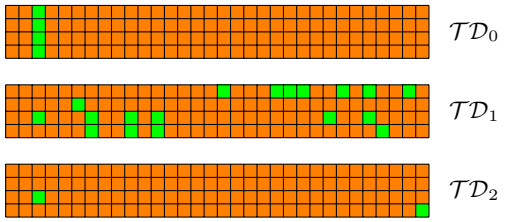


Fig. 3: The TD pattern $(\mathcal{TD}_0, \mathcal{TD}_1, \mathcal{TD}_2)$ of 3-round Serpent: an orange cell indicates a bit of which the bit difference always be inactive or arbitrary, which is of no concern; a green cell indicates a bit difference need to be determined

4.2 Estimate of the DL Bias when E_m Consists of Multiple Rounds

According to Formula 15, the TD pattern $\mathcal{TD}_i[j]$ servers as a TD mask $\mathcal{M}_i[j]$ when $\mathcal{TD}_i[j] \neq 0$. Suppose that Δ and λ represent the input difference and the output mask of E_m, respectively. In the following, we take the breadth-first search method to compute the DLP $(\Pr[\Delta \to \lambda])$, then we will obtain the differential-linear bias of E_m. Then, forward backward extensions are preformed to generate a longer DL distinguisher.

To compute the DLP $(\Pr[\Delta \to \lambda])$, we need to maintain a few tables:

1. A_i: a table to store all possible truncated output differences after the $\mathcal{S}$ layer in the i-th round. $A_i = \{\mathcal{T} = \Delta_i \& \mathcal{TD}_i | \Delta_i \in \mathbb{F}_2^n\}$, where Δ_i denotes all possible differences and A_i contains no duplicate elements. Let $A_{i,j}$ denote the j-th candidate in A_i, which is a truncated difference, and $A_{i,j}[k]$ denote the k-th S-box of $A_{i,j}$.
2. B_i: a table to store all possible truncated output differences in the i-th round, where $B_{i,j}$ denotes the j-th candidate, and $B_{i,j}[k]$ denotes the k-th S-box of $B_{i,j}$, and $B_{i,j} = \mathcal{L}(A_{i,j})$.
3. TDP_i: a table to store the truncated differential probability over i rounds, where $\mathrm{TDP}_{i,j}$ denotes that input difference Δ leads to the truncated output difference $A_{i,j}$ over i rounds, i.e., $\mathrm{TDP}_{i,j} = \Pr[\Delta \xrightarrow{i\ rounds} A_{i,j}]$.

In this paper, we offer a round-by-round approach to compute the truncated differential probabilities. For the $\mathcal{S}$ layer in the 0-th round and for each $0 \leq j < |A_0|$, we compute

$$\Pr[\Delta \xrightarrow{1\ round} A_{0,j}] = \prod_{k:\mathcal{TD}_0[k]\neq 0} \overline{\mathrm{TDT}}(\Delta[k], \mathcal{TD}_0[k], A_{0,j}[k])$$

and insert it into a table $\mathrm{TDP}_{i,j}$. Then the linear transformation is applied to each $A_{0,j}$ and the results are stored in the table B_0. We continue the same way for the $\mathcal{S}$ layer in the i-th round $(1 \leq i < R_m)$, and for each $0 \leq j < |A_i|$,

compute

$$\Pr[\Delta \xrightarrow{i\ rounds} A_{i,j}] = \sum_{t=0}^{|B_{i-1}|-1} \mathrm{TDP}_{i-1,t} \cdot \Pr[B_{i-1,t} \xrightarrow{\mathcal{S}} A_{i,j}]$$
$$= \sum_{t=0}^{|B_{i-1}|-1} \mathrm{TDP}_{i-1,t} \cdot \left(\prod_{k:\mathcal{TD}_i[k]\neq 0} \overline{\mathrm{TDT}}(B_{i-1,t}[k], \mathcal{TD}_i[k], A_{i,j}[k]) \right)$$

and insert it into $\mathrm{TDP}_{i,j}$. Then we apply the linear transformation to each $A_{i,j}$ and the results are stored in the table B_i. Finally, to compute DLP, we check each candidate in B_{R_m-1}, and we have

$$\Pr[\Delta \to \lambda] = \sum_{j=0}^{|B_{R_m-1}|-1} \mathrm{TDP}_{R_m-1,j} \cdot \pi(\lambda \cdot B_{R_m-1,j})$$

where $\pi(x) = 1$ if $x = 0$ and $\pi(x) = 0$ otherwise. Finally, the bias of E_m is $\Pr[\Delta \to \lambda] - \frac{1}{2}$.

Regarding the complexity, in round i with $0 < i < R_m$ (i.e., all of the rounds except the first), we need to traverse B_{i-1} and A_i, there is $|B_{i-1}| \cdot |A_i|$ multiplication and addition operations, which is equal to $|A_{i-1}| \cdot |A_i|$. For the $\mathcal{S}$ layer in the 0th round, because there is only an input difference, we need to do multiplication and addition $|A_0|$ times. In the last step, there are $|B_{R_m-1}|$ (i.e., $|A_{R_m-1}|$) multiplication and addition operations to compute DLP. We omit the expense of the linear transformation. Therefore, the computational complexity of this approach in multiplication and addition operations is about

$$|A_0| + |A_{R_m-1}| + \sum_{i=1}^{R_m-1} |A_{i-1}| \cdot |A_i| = 2^{hw(\mathcal{TD}_0 \| \mathcal{TD}_{R_m-1})} + \sum_{i=1}^{R_m-1} 2^{hw(\mathcal{TD}_{i-1} \| \mathcal{TD}_i)}$$

In each round, we must keep four tables: two for the current round (to store the truncated output differences and correlated probabilities) and two from the previous round (to store the truncated input differences and correlated probabilities). Therefore, the memory complexity is about

$$\max_{1 \le i < R_m} (|A_{i-1}| + |A_i| + |\mathrm{TDP}_{i-1}| + |\mathrm{TDP}_i|) = \max_{1 \le i < R_m} (2 \times (2^{hw(\mathcal{TD}_{i-1} \| \mathcal{TD}_i)}))$$

In practical application, to construct longer differential-linear distinguisher, we extend a differential with high-probability in the differential part E_0 and a linear approximation with high-bias in the linear part E_1.

4.3 Estimate of the DL Bias When E_m Consists of One Round

In this subsection, we try to use DLCT to connect a strong truncated differential and a strongly biased linear approximation. In addition, we introduce a new method to compute the truncated differential probability of E_0 with TDT.

If for E_0, we have the number of rounds R_0, the input difference Δ, and the truncated output difference $\mathcal{T}$, then we can generate a output mask λ according to $\mathcal{T}$,

$$\lambda^{\{i\}} = \begin{cases} 1, & if\ \mathcal{T}^{\{i\}} = 0\ or\ 1 \\ 0, & if\ \mathcal{T}^{\{i\}} = * \end{cases}$$

For a truncated differential that consist of insufficient truncated differential characteristics, its precise probabilities can be computed. Otherwise, instead of conducting an exhaustive search, we evaluate the truncated differential probability by finding as many high-probability truncated differential characteristics as possible using the TDT.

For each S-box TDT with fixed Δ_I and $\mathcal{M}$, we should reorder $\mathcal{Z}$ in descending order by the $\overline{\mathrm{TDT}}(\Delta_I, \mathcal{M}, \mathcal{Z})$, and then reorder $\mathcal{Z}$ in ascending order by the Hamming weight if the entries have the same probability, so that we can search the truncated differential characteristics with high-probability first.

Our program is based on depth-first search with branch-and-bound technique. Since traverse all possible truncated difference characteristics whose probabilities are greater than a threshold, we require a "threshold", which is represented as $\overline{TS}$. Define the probability of the truncated differential as P_{TD}.

The framework of our algorithm is now established by the following procedures including essentially recursive calls:

Procedure Round-0

Begin the program.

Let $P_{TD} = 0$.

For each candidate for $\mathcal{Z}_0$ with fixed $\mathcal{TD}_0$, do the following:

- Let $p_0 = \overline{\mathrm{TDT}}(\Delta X_0, \mathcal{TD}_0, \mathcal{Z}_0)$.
- If $p_0 \geq \overline{TS}$, then call *Procedure Round-1.*

Exit the program.

Procedure Round-i ($1 \leq i < R_0 - 1$)

For each candidate for $\mathcal{Z}_i$ with fixed $\mathcal{TD}_i$, do the following:

- Let $\Delta X_1 = \mathcal{L}(\mathcal{Z}_0)$ and $p_i = \overline{\mathrm{TDT}}(\Delta X_i, \mathcal{TD}_i, \mathcal{Z}_i)$.
- If $\prod_{k=0}^{i} p_k \geq \overline{TS}$, then call *Procedure Round-*$(i+1)$.

Reture to the upper procedure.

Procedure Round-$(R_0 - 1)$

For each candidate for $\mathcal{Z}_{R_0-1}$ with fixed $\mathcal{TD}_{R_0-1}$, do the following:

- Let $\Delta X_{R_0-1} = \mathcal{L}(\mathcal{Z}_{R_0-2})$.
- Let $p_{R_0-1} = \overline{\mathrm{TDT}}(\Delta X_{R_0-1}, \mathcal{TD}_{R_0-1}, \mathcal{Z}_{R_0-1})$.
- If $p = \prod_{k=0}^{R_0-1} p_k \geq \overline{TS}$, then a linear transformation is performed, i.e., $\Delta X_{R_0} = \mathcal{L}(\mathcal{Z}_{R_0-1})$.
- Let $\mathcal{Z}_{R_0} = \lambda \& \mathcal{T}$. If $\Delta X_{R_0} = \mathcal{Z}_{R_0}$, then $P_{TD} = P_{TD} + p$.

Reture to the upper procedure.

In the following, we show an explicit implementation of *Procedure Round-1* that realizes a practical search using another recursive calls. Other procedures can be also carried out in a similar way:

Procedure Round-1: (detailed)

Call *Procedure Round-1-0.*

Return to the upper procedure.

Procedure Round-1-j $(0 \le j < \ell)$

If $\Delta X_1[j] = 0$ or $\mathcal{TD}_1[j] = 0$, then $\mathcal{Z}_1[j] = 0$ and $p_1[j] = 1$, call *Procedure Round-1-*$(j+1)$

If $\Delta X_1[j] \neq 0$ and $\mathcal{TD}_1[j] \neq 0$, for each candidate for $\mathcal{Z}_1[j]$, do the following:

- Let $p_1[j] = \overline{\mathrm{TDT}}(\Delta X_1[j], \mathcal{TD}_1[j], \mathcal{Z}_1[j])$ and $p_1 = \prod_0^j p_1[j]$
- If $p_0 \times p_1 \ge \overline{TS}$ and $j \neq \ell - 1$, then call *Procedure Round-1-*$(j+1)$
- Call *Procedure Round-2*

Reture to the upper procedure.

5 Experimental Results

5.1 Applications to Ascon

The previous results. The Ascon family [20] of cryptographic primitives were selected for standardization in the Lightweight Cryptography Competition by NIST. In [16], the authors started with the analysis of a 4-round initialization and created a differential-linear characteristic for it. They placed the S-boxes in a way that the linear active S-boxes in round 3 do not overlap with the 11 differential active S-boxes they estimated. The bias of the generated DL distinguisher was $2pq^2 = 2^{-20}$. And then, they checked the vast amount of possible combinations of differential and linear characteristic experimentally. They found that in the best case, they placed difference in bit 63 of x_3 and x_4, and got a bias of 2^{-2} in bit 9 of x_0 on the output of the substitution layer of round 4. For a 5-round initialization, they obtained a bias of 2^{-10} on $x_0[16]$ (last substitution layer) for differences in $x_3[63]$ and $x_4[63]$. In [10], the authors obtained a higher bias of 2^{-5} by revisiting the analysis of the 4-round distinguisher in [16] using the DLCT, which is still much lower than the experimentally obtained bias of 2^{-2}. In [24], the authors studied the DL cryptanalysis from an algebraic perspective by introducing a technique called DATF. They obtained a bias of $2^{-2.365}$ of the 4-round distinguisher in [16]. In [34], the authors estimated the bias of the well-studied 4-round Ascon's DL distinguisher as $2^{-2.09}$, which is better than previous tools such as the DLCT [16] (2^{-5}) or the ATF [24] ($2^{-2.365}$); also, they gave the theoretical bias for the 5-round Ascon's DL distinguisher for the first time: the bias is estimated as 2^{-10}.

Table 4: The 4-round DL distinguisher

the input difference Δ	the output mask λ
0000000000000000	0000000000000200
0000000000000000	0000000000000000
0000000000000000	0000000000000000
8000000000000000	0000000000000000
8000000000000000	0000000000000000

Table 5: The value of Δ_0'

the input difference Δ_0'
0000000000000000
0000000000000000
0000000000000000
0000000000000000
8100000000400000

Revisiting the 4-round distinguisher. We now apply our approach in Sect.4.2 to estimate the bias of the 4-round distinguisher in [16]. Let us decompose E into $E = E_1 \circ E_m$, where E_m consists of rounds 1-4, and $E_1 = Id$. The input difference Δ and the output mask λ are the same with those in [16], which are showed in Table 4 (the arrangement is from right to left, from bottom to top). Using Algorithm 1 and the approach in Sect.4.2, we obtain the bias of the 4-round distinguisher in [16] is 2^{-2}, which is the same as the experimental result in [16].

Revisiting the 5-round distinguisher. Using the approach in Sect.4.3, we revisit the 5-round DL distinguisher in [16], which has a experimental bias of 2^{-10}. Let us decompose E into $E = E_1 \circ E_m \circ E_0$, where E_0 consists of round 1-4, E_m consists of round 5, and $E_1 = Id$. Note that since in the DL distinguisher of [16], the output mask λ consists of the MSB in the output of S-box no.16, we are only interested in the entries of the DLCT of that S-box. For E_0, we use a truncated differential of the form $\Delta_0 \xrightarrow[4\ rounds]{p} \Delta_4$, where Δ_0 is the input difference of the DL distinguisher of [16]. In our value of Δ_4, four of the input bits to S-box no.16 are known to be zero; specifically, the input is of the form $00*00$. The probability of the truncated differential p is approximately equal to $2^{-8.1}$. The relevant normalized entries of the DLCT of S satisfy:

$$\overline{\mathrm{DLCT}}_S(0,16) = 2^{-1}, \quad \overline{\mathrm{DLCT}}_S(4,16) = 0$$

Hence, assuming that each input difference of S occurs in Δ_4 with the same probability 2^{-1} and using Eq. (2), we obtain the estimate

$$\mathcal{E} = 4 \cdot 2^{-8.1} \cdot 2^{-1}(2^{-1} + 0) \cdot (2^{-1})^2 = 2^{-10.1}$$

for the overall bias of the 5-round DL distinguisher of [16]. This value is extremely close to the experimentally obtained bias of 2^{-10}. In [24], the authors obtained a 5-round conditional DL distinguisher with a bias of $2^{-5.415}$. However, in our paper, we do not inject any conditions on the 5-round DL distinguisher.

New 6-round distinguisher. Using the approach in Sect.4.3, we obtain a new 6-round distinguisher with a bias of $2^{-22.43}$, which is the best theoretical results up to date. We first introduce a 5-round DL distinguisher of Ascon, and then extend one round to construct the 6-round DL distinguisher. Let us decompose a 5-round reduced variant E' into $E' = E'_1 \circ E'_m \circ E'_0$, where E'_0 consists of round 2-5, E'_m consists of round 6, and $E'_1 = Id$. For E'_0, we use a truncated differential of form $\Delta'_0 \xrightarrow[4\ rounds]{p'} \Delta'_4$. The value of Δ'_0 is shown in Table 5.

For the 4-round truncated differential $\Delta'_0 \xrightarrow[4\ rounds]{p'_1} \Delta'_4$, we are only interested in the entries of the DLCT of S-box no.57. In our value of Δ'_4, one of the input bits to S-box no.57 are known to be zero; specifically, the input is of the form $*0**$. The probability of the 4-round truncated differential is $p'_1 = 2^{-17.43}$.

In this case, for E'_m, the output mask only consists of the MSB in the output of S-box no.57, and the relevant normalized entries of the DLCT of S satisfy the

Table 6. Assuming that each input difference of S occurs in Δ_5' with the same probability 2^{-4} and using Eq.(2) and Table 6, we obtain the estimate

$$\mathcal{E} = 4 \cdot 2^{-17.43} \cdot 2^{-4}(2^{-1} + 2^{-1} + 2^{-1} + 2^{-1} + 2^{-1} + 2^{-1} + 2^{-1} + 2^{-1}) \cdot (2^{-1})^2 = 2^{-19.43}$$

for the overall bias of the 5-round DL distinguisher. And then, we append one round with probability 2^{-3} before the 5-round DL distinguisher, and obtain a 6-round DL distinguisher with the bias of $2^{-22.43}$. This is the first theoretical result of Ascon's 6-round DL distinguisher.

Table 6: The relevant normalized entries of the DLCT in the S-box no.57

$\overline{\text{DLCT}}_S(0,16) = 2^{-1}$	$\overline{\text{DLCT}}_S(1,16) = 0$	$\overline{\text{DLCT}}_S(2,16) = 2^{-1}$	$\overline{\text{DLCT}}_S(3,16) = 0$
$\overline{\text{DLCT}}_S(4,16) = 0$	$\overline{\text{DLCT}}_S(5,16) = 2^{-1}$	$\overline{\text{DLCT}}_S(6,16) = 0$	$\overline{\text{DLCT}}_S(7,16) = 2^{-1}$
$\overline{\text{DLCT}}_S(16,16) = 0$	$\overline{\text{DLCT}}_S(17,16) = 2^{-1}$	$\overline{\text{DLCT}}_S(18,16) = 0$	$\overline{\text{DLCT}}_S(19,16) = 2^{-1}$
$\overline{\text{DLCT}}_S(20,16) = 2^{-1}$	$\overline{\text{DLCT}}_S(21,16) = 0$	$\overline{\text{DLCT}}_S(22,16) = 2^{-1}$	$\overline{\text{DLCT}}_S(23,16) = 0$

5.2 Applications to Serpent

The differential-linear distinguishers of Serpent The DL attack on the AES finalist Serpent [12] presented in [13] is based on a 9-round DL distinguisher with a bias of 2^{-60}, whihc starts with round 2. In [14], Dunkelman et al. performed experiments with reduced round variants of Serpent, and concluded that the actual bias of the 9-round DL distinguisher is $2^{-57.75}$ instead of 2^{-60}. In [10], Bar-On et al. recomputed the bias of the distinguisher using the DLCT and obtained an estimate of $2^{-57.68}$. In [24], Liu et al. recomputed the bias of the distinguisher using DATF and obtained an estimate of $2^{-57.736}$.

In this part, we revisit the analysis of the bias of the 9-round DL distinguisher using the approach in Sect.4.2, and show an estimate of $2^{-57.696}$, which is very close to the experimental result in [14].

We adopt the notations of [10,13]. The exact difference and mask values are obtained from [13]. The 9-round reduced variant of Serpent that starts with round 2 in [13] is denoted by E and decomposed as $E = E_1 \circ E_0$, where E_0 consists of rounds 2-4 and E_1 consists of 5-10. For E_0, the distinguisher uses a differential characteristic of the form

$$\Delta_0 \xrightarrow[L \circ S_2]{p_0 = 2^{-5}} \Delta_1 \xrightarrow[L \circ S_3]{p_1 = 2^{-1}} \Delta_2 \xrightarrow[L \circ S_4]{p_2 = 1} \Delta_3,$$

where Δ_2, Δ_3 are truncated differences. For E_1, the distinguisher uses a linear approximation of the form

$$\lambda_0 \xrightarrow[L \circ S_5]{q_0 = 2^{-5}} \lambda_1 \xrightarrow[L \circ S_6]{q_1 = 2^{-3}} \lambda_2 \xrightarrow[4\ rounds]{q_2 = 2^{-21}} \lambda_6,$$

where all nonzero bits of the mask λ_0 are included in the bits that are known to be zero in Δ_3. By experiments the authors of [13] found out that there are other differentials which also predict the difference in the bits of λ_0. Summing all these

differentials, they obtained that the probability that $\lambda_0 \cdot \Delta_3 = 0$ is $\frac{1}{2} + 2^{-7}$ and hence used it in their analysis. Using the complexity analysis of the classical DL framework, the authors of [13] concluded that the overall bias of the 9-round DL distinguisher is $2 \times 2^{-7} \times (2^{-27})^2 = 2^{-60}$.

The authors of [14] checked experimentally the first 4-round DL distinguisher in [13] (i.e., a 4-round distinguisher which starts with the difference Δ_0 and ends with the mask λ_1) and found that its bias is $2^{-13.75}$, instead of the estimate $2 \times 2^{-7} \times (2^{-5})^2 = 2^{-16}$. Therefore, they concluded that the bias of the 9-round distinguisher is $2^{-57.75}$ instead of 2^{-60}.

The authors of [10] considered a 3-round variant of Serpent that starts at round 3, denoted it by E', and computed the entry $\overline{\mathrm{DLCT}}_{E'}(\Delta_1, \lambda_1)$. They found that its bias is $2^{-8.68}$. Hence they concluded that the bias of the first 4-round DL distinguisher in [13] is $2^{-5} \times 2^{-8.68} = 2^{-13.68}$.

The authors of [24] revisited the 3-round variant of Serpent E' starting at round 3, and obtained a bias $\varepsilon = 2^{-8.736}$. They concluded that the bias of the first 4-round DL distinguisher in [13] is $2^{-5} \times 2^{-8.736} = 2^{-13.736}$. Furthermore, they concluded that the bias of the first 5-round DL distinguisher in [13] (i.e., a 5-round distinguisher which starts with the difference Δ_0 and ends with the mask λ_2) is $2^{-5} \times 2^{-12.736} = 2^{-17.736}$.

Revisiting the 4-round and 5-round DL distinguishers We apply the approach presented in Sect.4.2 to the 3-round variant of Serpent E' considered in [10] and obtain a bias $2^{-8.696}$. Therefore we conclude that the bias of the first 4-round DL distinguisher in [13] is $2^{-5} \times 2^{-8.696} = 2^{-13.696}$, which is very close to the experimental value.

We further revisit the first 5-round DL distinguisher in [13]. We denote the 5-round variant of Serpent that starts at round 2 by E' and decompose it as $E' = E'_1 \circ E_m \circ E'_0$, where E'_0 consists of round 2, E_m consists of rounds 3-5 and E'_1 consists round 6. For E'_0, the distinguisher uses a differential characteristic of the form $\Delta_0 \xrightarrow[LT \circ S_2]{p_0 = 2^{-5}} \Delta_1$. For E'_1, the distinguisher uses a linear approximation of the form $\lambda_1 \xrightarrow[LT \circ S_6]{q_1 = 2^{-3}} \lambda_2$. Since we can obtain that the bias of E_m is $2^{-8.696}$, we can get the bias of the 5-round distinguisher in [24] is $4 \times 2^{-5} \times 2^{-8.696} \times (2^{-3})^2 = 2^{-17.696}$. Therefore, we can obtain the bias of the 9-round DL distinguisher in [13] is $4 \times 2^{-5} \times 2^{-8.696} \times (2^{-23})^2 = 2^{-57.696}$.

New and Better DL distinguishers In this part, we show that the distinguishers can be improved by using another combination of a differential characteristic and a linear approximation, which leads to a higher bias. Table 7 summarizes our results for Serpent up to 9 rounds. The column labeled 'start round' indicates the round at which the DL distinguisher begins. Note that for 9-round distinguishers, we consider two cases: the first case, the theoretial bias is as high as possible (i.e., the row of 9^1 in Table 7), and the second case, the number of active S-boxes of the input difference and the output mask is as low as

possible (i.e., the row of 9^2 in Table 7), which are both better than the 9-round DL distinguisher in [13]. Note that for the 4-round and 5-round DL distinguishers, the biases we obtain are much better than those reported in [13]. The details of each DL distinguisher are summarized in Table 12.

Table 7: The differential-linear bias of Serpent

RN	start round	R_0	R_m	R_1	theoretical bias	experimental result
3	5	0	3	0	$2^{-1.415}$	$2^{-1.415}$
4	7	1	3	0	$2^{-5.415}$	$2^{-5.30}$
5	5	1	3	1	$2^{-11.415}$	$2^{-11.44}$
6	4	1	3	2	$2^{-19.61}$	
7	1	1	3	3	$2^{-29.45}$	
8	1	1	3	4	$2^{-39.45}$	
9^1	1	2	2	5	2^{-52}	
9^2	2	1	3	5	$2^{-55.43}$	

5.3 Applications to KNOT

In 2019, KNOT was selected by NIST as one of the 32 candidates in the second round of lightweight cryptography (LWC) standardization process. The KNOT AEAD family has 4 members. In this paper, we consider the primary AEAD member–KNOT-AEAD(128,256,64) (short as KNOT256), which has a state size of 256.

An experimental validation Firstly, we perform some statistical tests to verify our approach in Sect.4.2. Given the input difference Δ and the output mask λ (shown in Table 8(a)), we evaluate the biases which consists of different number of rounds, the experimental results are shown in Table 8(b). As we can see, there is no gap between our theoretical values and the experimental results. Note that in the hexadecimal presentation of states, the top row denotes a_0, the bottom row denotes a_3, and the arrangement of S-boxes is from right to left.

The DL distinguishers In this paper, we focus on the analysis of the round-reduced initialization phase, so the input differences are limited to the nonce, and the output masks are limited to the rate part. Using the approach in Sect.4.2, we find that the biases of the optimal DL distinguishers are all 2^{-1} when the number of rounds of E_m is from 2 to 8, so we construct distinguishers starting from 9 rounds.

We first compute the bias of E_m covered over multiple rounds while constraining the number of active S-boxes in the input differences and the output

Table 8: The experimental validation

(a) The input difference Δ and output mask λ in statistical tests

input difference Δ	0000000000000000
	0000000000000000
	0000000000000001
	0000000000000000
output mask λ	0000000000000000
	0000000000000000
	0000000000010000
	0000000000000000

(b) The verification results

RN	Theoretical results	Experimental results
3	2^{-3}	2^{-3}
4	$2^{-2.415}$	$2^{-2.415}$
5	$2^{-1.83}$	$2^{-1.83}$
6	$2^{-1.476}$	$2^{-1.476}$
7	$2^{-1.272}$	$2^{-1.272}$
8	$2^{-1.154}$	$2^{-1.154}$
9	$2^{-1.24}$	$2^{-1.24}$

linear masks of E_m to be 1. We then try all possibilities and then extend them with differential and linear trails to evaluate the overall bias of the DL distinguishers. Using this approach, we can construct the DL distinguishers for up to 13 rounds.

To construct longer DL distinguishers, we initially search for differential and linear characteristics, where the differential characteristics have one active S-box in the input difference, while minimizing the number of active S-boxes in the output difference. Additionally, for the linear approximations, we look for one active S-box in the output mask, and minimize the number of active S-boxes in the input mask. We utilize DL distinguishers that consist of number of rounds as high as possible to connect the differential characteristics and the linear approximations. Because of rotation invariance, we obtain numerous combinations for each combination of differential and linear trails. Consequently, we employ our approaches to assess the overall bias of all feasible combinations and select the best one. Table 13 summarizes the selected DL distinguishers.

5.4 Applications to AES

The *Advanced Encryption Standard* (AES) [36] is the most widely adopted block cipher in the world today. The AES is a *Substitution-Permutation network* that supports key size of 128, 192 and 256 bits. The number of rounds are 10/12/14 for AES-128/192/256, respectively. An AES round applies four operations to the state matrix, which can be seen as $R = AK \circ MC \circ SR \circ SB$, where the non-linear layer is $\mathcal{S} = SB$, the linear layer is $\mathcal{L} = AK \circ MC \circ SR$.

We first exhaust all possible 3-round DL distinguishers where the number of active S-boxes in the first round and the third round is both 1. The bias of the best 3-round DL distinguisher we obtained is $2^{-8.66}$.

Table 9: The differential-linear bias of KNOT256

RN	R_0	R_m	R_1	theoretical bias	experimental result
9	0	9	0	$2^{-1.20}$	$2^{-1.20}$
10	1	9	0	$2^{-3.66}$	$2^{-3.27}$
11	1	9	1	$2^{-6.38}$	$2^{-4.31}$
12	2	9	1	$2^{-9.27}$	$2^{-9.91}$
13	3	9	1	$2^{-12.27}$	$2^{-14.04}$
14	4	9	1	$2^{-16.23}$	
15	4	8	3	$2^{-23.31}$	
16	4	8	4	$2^{-30.52}$	

For the 4-round and 5-round DL distinguisher, we firstly construct a 3-round DL distinguisher where the number of active S-box in the first round is 1, and the number of active S-box in the third round is 4, which has the bias of $2^{-21.85}$. A 4-round DL distinguisher is obtained by appending a 1-round linear approximation after the 3-round DL distinguisher. The 4-round DL distinguisher has a bias of $2^{-27.85}$, where the bias of the 1-round linear approximation is 2^{-4}, and the number of active S-box in the fourth round is 1. After that, if we perform forward extension on the 4-round DL distinguisher by adding a one round differential trail with probability 2^{-24}, a 5-round DL distinguisher is constructed. The overall bias of the 5-round distinguisher is $2^{-51.85}$. The composition for different DL distinguishers is shown in Table 10, and the details are presented in Table 14 in the Appendix.

Table 10: The differential-linear bias of AES

RN	R_0	R_m	R_1	theoretical bias	experimental result
2	0	2	0	2^{-1}	2^{-1}
3	0	3	0	$2^{-8.66}$	$2^{-8.66}$
4	1	3	0	$2^{-27.85}$	
5	1	3	1	$2^{-51.85}$	

5.5 Applications to CLEFIA

CLEFIA is a 128-bit block cipher with variable key lengths of 128, 192 and 256, which takes a 4-branch generalized Feistel network [35]. The number of rounds are 18/22/26 for CLEFIA-128/192/256, respectively. The procedure of encryption is described in [35].

In this paper, we construct the DL distinguishers up to 9 rounds. Since the biases of the optimal DL distinguishers are all 2^{-1} when the number of rounds

of E_m is less than 5, we construct DL distinguishers starting from 5 rounds. We first search two 5-round DL distinguishers with the same biases, and then perform forward and backward extension to construct long DL distinguishers.

When searching 5-round DL distinguishers, for the input difference, we limit that the number of active S-box is only one. For the output mask, there are two cases. In the first case, we limit that the output mask has one active S-box. In the second case, to make the number of active S-box in the linear trails as low as possible. After that, we search the differential characteristics with high-probabilities and linear approximations with high-biases. This paper presents the best combinations of a differential characteristic and linear approximation. The compositions for different rounds of DL distinguishers and the bias are shown in Table 11. The details are presented in Table 15 in the Appendix.

Table 11: The differential-linear bias of CLEFIA

RN	R_0	R_m	R_1	theoretical bias	experimental result
4	0	4	0	2^{-1}	2^{-1}
5	0	5	0	$2^{-2.54}$	$2^{-2.54}$
6	1	5	0	$2^{-7.54}$	$2^{-7.54}$
7	1	5	1	$2^{-12.37}$	
8	1	5	2	$2^{-33.59}$	
9	2	5	2	$2^{-55.84}$	

6 Conclusion

Based on the TDT and a relation between differential-linear and truncated differential cryptanalysis, we propose two new approaches to estimate the bias of a differential-linear distinguisher. In practical applications, the choice of which approach to use is worth discussing. Here are our suggestions. For ciphers where the $\mathcal{L}$ layer is weak and the bias of E_m consisting of multiple rounds is easy to estimate, the approach in Sect.4.2 is generally a better option; for ciphers where the $\mathcal{L}$ layer is strong, and the bias estimate of E_m consisting of multiple rounds is computationally too demanding by using the approach in Sect.4.2, then the approach in Sect.4.3 is preferred.

We demonstrate the accuracy and efficiency of our new approaches by applying to 5 symmetric-key primitives: the LWC winner Ascon, the AES finalist Serpent, and the LWC candidate KNOT, AES, and CLEFIA:

- For Ascon, we revisit the previous 4-round and 5-round DL distinguishers, our results closely match the experimental results. Then, we improve the number of DL distinguisher's rounds from 5 to 6.

- For Serpent, we revisit the bias estimate of a known 9-round DL distinguisher. We show that our estimate of the bias is very close to the experimental result for the 4-round distinguisher. The bias of the 5-round DL distinguisher is also revisited. Furthermore, two completely new 9-round distinguishers with higher biases are presented.
- For KNOT, we search DL distinguishers up to 16 rounds. Our 16-round DL distinguisher has a bias of $2^{-30.52}$, which is the best distinguisher of KNOT256 with respect to the number of rounds, and possibly used to mount better DL attacks on reduced-round KNOT256.
- For AES, we search DL distinguishers up to 5 rounds. Our 5-round DL distinguisher has a bias of $2^{-51.85}$, which is the best distinguisher of AES with respect to the number of rounds.
- For CLEFIA, we search DL distinguishers up to 9 rounds. Our 9-round DL distinguisher has a bias of $2^{-55.84}$, which is one of the best distinguishers of CLEFIA.

Acknowledgements We are grateful to the anonymous reviewers of Asiacrypt 2023, Eurocrypt 2024 and CRYPTO 2024 for their valuable feedback, which helped us to improve our work. The research was supported by National key research and development program (No. 2023YFB4503203) and by the Chinese Academy of Sciences(CAS) Project for Young Scientists in Basic Research under Grant YSBR-035 and by the National Natural Science Foundation of China (No. 61379138).

References

1. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1990).
2. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard, Springer 1993.
3. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1993).
4. Langford, S.K., Hellman, M.E.: Differential-linear cryptanalysis. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 17–25. Springer, Heidelberg (1994).
5. Biham, E., Dunkelman, O., Keller, N.: Enhancing differential-linear cryptanalysis. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 254–266. Springer, Heidelberg (2002).
6. Lu, J.: A methodology for differential-linear cryptanalysis and its applications - (extended abstract). In: Canteaut, A. (ed.) FSE 2012. LNCS, vol.7549, pp. 69–89. (2012).
7. Blondeau, C., Leander, G., Nyberg, K.: Differential-linear cryptanalysis revisited. J. Cryptol. 30(3), 859–888 (2017).
8. Blondeau, C., Nyberg, K.: New Links between Differential and Linear Cryptanalysis. In: Johansson, T., Nguyen, P.Q. (eds) EUROCRYPT 2013. LNCS, vol 7881, pp. 388-404. Springer, Heidelberg (2013).

9. Chabaud, F., Vaudenay, S.: Links between differential and linear cryptanalysis. In: De Santis, A. (eds) EUROCRYPT 1994. LNCS, vol 950, pp. 356-365 Springer, Heidelberg (1994).
10. Bar-On, A., Dunkelman, O., Keller, N., Weizman, A.: DLCT: a new tool for differential-linear cryptanalysis. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11476, pp. 313–342. Springer, Cham (2019).
11. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: Boomerang Connectivity Table: A New Cryptanalysis Tool. In: Nielsen, J., Rijmen, V. (eds) EUROCRYPT 2018. LNCS, vol 10821, pp. 683-714 Springer, Cham (2018).
12. Biham, E., Anderson, R., Knudsen, L.: Serpent: A New Block Cipher Proposal. In: Vaudenay, S. (eds) FSE 1998. LNCS, vol 1372, pp. 222-238 Springer, Heidelberg (1998).
13. Biham, E., Dunkelman, O., Keller, N.: Differential-linear cryptanalysis of Serpent. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 9–21. Springer, Heidelberg (2003).
14. Dunkelman, O., Indesteege, S., Keller, N.: A differential-linear attack on 12-round Serpent. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 308–321. Springer, Heidelberg (2008).
15. Biham, E., Dunkelman, O., Keller, N.: Linear cryptanalysis of reduced round Serpent. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 16–27. Springer, Heidelberg (2002).
16. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Cryptanalysis of Ascon. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 371–387. Springer, Cham (2015).
17. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. 21(1), 131–147 (2008).
18. Eichlseder, M., Leander, G., Rasoolzadeh, S.: Computing Expected Differential Probability of (Truncated) Differentials and Expected Linear Potential of (Multidimensional) Linear Hulls in SPN Block Ciphers. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds) INDOCRYPT 2020. LNCS, vol 12578, pp. 345-369. Springer, Cham (2020).
19. Li, L., Jia, K., Wang, X., Dong, X.: Meet-in-the-Middle Technique for Truncated Differential and Its Applications to CLEFIA and Camellia. In: Leander, G. (eds) Fast Software Encryption. FSE 2015. LNCS, vol 9054. pp. 48-70. Springer, Heidelberg.
20. Dobraunig, C., Eichlseder, M., Mendel, F., Schl¨affer, M.: Ascon v1.2. Submission to the CAESAR Competition (2016)
21. Dobraunig, C., Eichlseder, M.,Mendel, F., Schl¨affer, M.: Ascon v1.2. Submission to the NIST Lightweight Cryptography competition (2019)
22. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. In: De Santis, A. (eds) EUROCRYPT 1994. LNCS, vol 950, pp. 366-375. Springer, Heidelberg (1995).
23. Beierle, C., Leander, G., Todo, Y.: Improved differential-linear attacks with applications to ARX ciphers. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12172, pp. 329–358. Springer, Cham (2020).
24. Liu, M., Lu, X., Lin, D.: Differential-linear cryptanalysis from an algebraic perspective. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part III. LNCS, vol. 12827, pp. 247–277. Springer Cham (2021).
25. Zhang, W., et al.: KNOT: algorithm specifications and supporting document.
26. Zhang, W., Ding, T., Zhou, C., Ji, F.: Security Analysis of KNOT-AEAD and KNOT-Hash, Fourth Lightweight Cryptography Workshop, NIST, 2020

27. Knudsen, L.R.: Truncated and higher order differentials. In: Preneel, B. (eds) FSE 1994. LNCS, vol 1008, pp. 196-211. Springer, Heidelberg (1995).
28. Wang, S., Hou, S., Liu, M., Lin, D.: Differential-Linear Cryptanalysis of the Lightweight Cryptographic Algorithm KNOT. In: Yu, Y., Yung, M. (eds) Inscrypt 2021. LNCS, vol 13007. 171-190 Springer, Cham (2021).
29. Knudsen, L.R., Robshaw, M.J.: Truncated Differentials. In: The block cipher companion, Information security and cryptography, pp. 154–159. Springer Berlin Heidelberg (2011)
30. Canteaut, A., et al.: On the differential-linear connectivity table of vectorial Boolean functions. CoRR, abs/1908.07445 (2019)
31. Canteaut, A., K¨olsch, L., Wiemer, F.: Observations on the DLCT and absolute indicators. IACR Cryptol. ePrint Arch. 2019, 848 (2019)
32. Liu, Z., Gu, D., Zhang, J., Li, W.: Differential-multiple linear cryptanalysis. In: Bao, F., Yung, M., Lin, D., Jing, J. (eds.) Inscrypt 2009. LNCS, vol. 6151, pp. 35–49. Springer, Heidelberg (2010).
33. Jiqiang, L.: A methodology for differential-linear cryptanalysis and its applications. Des. Codes Cryptogr. 77(1), 11–48 (2015).
34. Hu, K., Peyrin, T., Tan, Q.Q., Yap, T.: Revisiting Higher-Order Differential-Linear Attacks from an Algebraic Perspective. ASIACRYPT 2023. LNCS, vol 14440. Springer, Singapore.
35. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit block cipher CLEFIA (extended abstract). In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 181–195. Springer, Heidelberg (2007)
36. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer, 2002.
37. Lai, X., Massey, J.L.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)

A The Differential-Linear Distinguishers of Serpent

In Table 12, we summarizes our results that DL distinguishers for 3 to 9 rounds of Serpent.

Table 12: The DL distinguishers of Serpent

#R	the differential-linear distinguisher	bias
3	start round: 5, $r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-1.415}$	$2^{-1.415}$
	$\Delta_O = 0x00000003000000000000000000000000 \quad \lambda_I = 0x08010812a00a200400000010000a0000$	
4	start round: 7, $p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-2},\ r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-3.415}$	$2^{-5.415}$
	$\Delta_I = 0x00600000000000000000000000000000 \quad \Delta_O = 0x02000000000000000000000000001000$	
	$\lambda_I = 0x00000000000000008010812a00a2004$	
	$\Delta_I = 0x06000000000000000000000000000000 \quad \Delta_O = 0x20000000000000000000000000010000$	
	$\lambda_I = 0x00000000000000008010812a00a20040$	
	$\Delta_I = 0x09000000000000000000000000000000 \quad \Delta_O = 0x20000000000000000000000000010000$	
	$\lambda_I = 0x0000010000a000008010812a00a20040$	
	$\Delta_I = 0x00900000000000000000000000000000 \quad \Delta_O = 0x02000000000000000000000000001000$	
	$\lambda_I = 0x00000010000a000008010812a00a2004$	
5	start round: 5, $p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-2.415},\ q(\lambda_I \xrightarrow{R_1=1} \lambda_O) = 2^{-3}$	$2^{-11.415}$
	$\Delta_I = 0xc0000000000000000000000000000100 \quad \Delta_O = 0x00000040000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a0000 \quad \lambda_O = 0x400014000b0000a00100023000220200$	
	$\Delta_I = 0x0000000000000000000000000000100c \quad \Delta_O = 0x00000040000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a00000 \quad \lambda_O = 0xc3a0b160000000000000000004d002$	
	$\Delta_I = 0x0000000000000000000000000000100c0 \quad \Delta_O = 0x00000040000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a000000 \quad \lambda_O = 0x3a0b160000000000000000004d002c$	
	$\Delta_I = 0x0000000000000000000000000000100c00 \quad \Delta_O = 0x00004000000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a0000000 \quad \lambda_O = 0xa0b160000000000000000004d002c3$	
	$\Delta_I = 0x0000000000000000000000000000100c000 \quad \Delta_O = 0x00040000000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a00000000 \quad \lambda_O = 0x0b160000000000000000004d002c3a$	
	$\Delta_I = 0x0000000000000000000000000000100c0000 \quad \Delta_O = 0x00400000000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a000000000 \quad \lambda_O = 0xb160000000000000000004d002c3a0$	
	$\Delta_I = 0x0000000000000000000000000000100c00000 \quad \Delta_O = 0x04000000000000000000000000000000$	
	$\lambda_I = 0x00000000000000000000010000a0000000000 \quad \lambda_O = 0x160000000000000000004d002c3a0b$	
6	start round: 4, $p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-4},\ r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-3.61},\ q(\lambda_I \xrightarrow{R_1=2} \lambda_O) = 2^{-7}$	$2^{-19.61}$
	$\Delta_I = 0x00900000000000b0000000000000000 \quad \Delta_O = 0x00000000000000000000000000000100$	
	$\lambda_I = 0x00000000000000010000a00000000000 \quad \lambda_O = 0x92002b8810800a00104000a2002acbb0$	
	$\Delta_I = 0x09000000000000b0000000000000000 \quad \Delta_O = 0x00000000000000000000000000001000$	
	$\lambda_I = 0x00000000000000010000a000000000000 \quad \lambda_O = 0x2b0130820a01b000400b02100060a200$	
7	start round: 1, $p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-8.45},\ q(\lambda_I \xrightarrow{R_1=3} \lambda_O) = 2^{-9}$	$2^{-29.45}$
	$\Delta_I = 0x00000100a0000000000000000000000 \quad \Delta_O = 0x00000000000000040000000000000000$	
	$\lambda_I = 0x00000000000000000000000800000000 \quad \lambda_O = 0x1080230000206004020140000b0000a00$	
	$\Delta_I = 0x00000100a0000000000000000000000 \quad \Delta_O = 0x00000000000000040000000000000000$	
	$\lambda_I = 0x00000000000000000000000800000000 \quad \lambda_O = 0x00810900a0206004020140000b0000a80$	
	$\Delta_I = 0x0000100a0000000000000000000000 \quad \Delta_O = 0x00000000000000040000000000000000$	
	$\lambda_I = 0x00000000000000000000000800000000 \quad \lambda_O = 0x0802300002060040201400 0b0000a001$	

	$\Delta_I = 0x0000100a0000000000000000000000$ $\Delta_O = 0x00000000000000400000000000000000$ $\lambda_I = 0x00000000000000000000008000000000$ $\lambda_O = 0x0810900a02060040201400 0b0000a800$	
8	start round: 1, $p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-8.45},\ q(\lambda_I \xrightarrow{R_1=4} \lambda_O) = 2^{-14}$	$2^{-39.45}$
	$\Delta_I = 0x00000100a0000000000000000000000$ $\Delta_O = 0x00000000000000400000000000000000$ $\lambda_I = 0x00000000000000000000000800000000$ $\lambda_O = 0x010000a000008010012a082010000a04$	
	$\Delta_I = 0x00000100a0000000000000000000000$ $\Delta_O = 0x00000000000000400000000000000000$ $\lambda_I = 0x00000000000000000000000800000000$ $\lambda_O = 0x00000000000008010812a082210400a04$	
	$\Delta_I = 0x0000100a0000000000000000000000$ $\Delta_O = 0x00000000000000400000000000000000$ $\lambda_I = 0x00000000000000000000008000000000$ $\lambda_O = 0x10000a000018010a12a08a0000020040$	
	$\Delta_I = 0x0000100a0000000000000000000000$ $\Delta_O = 0x00000000000000400000000000000000$ $\lambda_I = 0x00000000000000000000008000000000$ $\lambda_O = 0x10000b0000b8010a92b00b2a00a00000$	
9	start round: 1, $p(\Delta_I \xrightarrow{R_0=2} \Delta_O) = 2^{-7},\ r(\Delta_O \xrightarrow{R_m=2} \lambda_I) = 2^{-7},\ q(\lambda_I \xrightarrow{R_1=5} \lambda_O) = 2^{-20}$	2^{-52}
	$\Delta_O = 0x0000000000000000200a000000000000$ $\Delta_O = 0x00000000000000010000810000200440$ $\lambda_I = 0x04000000000000000000000000000020$ $\lambda_O = 0x2a00308043090ab2e02a240400801 08a$	
9	start round: 2, $p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-4},\ r(\Delta_O \xrightarrow{R_m=3} \lambda_I) = 2^{-7.33},\ q(\lambda_I \xrightarrow{R_1=5} \lambda_O) = 2^{-23}$	$2^{-55.33}$
	$\Delta_I = 0x0000000004007000000000000000000$ $\Delta_O = 0x00000000000000000000400000000000$ $\lambda_I = 0x00000000000000000000000080000000$ $\lambda_O = 0x000b0000b0000300 00b0200e00000010$	
	$\Delta_I = 0x0000000004007000000000000000000$ $\Delta_O = 0x00000000000000000000400000000000$ $\lambda_I = 0x00000000000000000000000800000000$ $\lambda_O = 0x00b0000b0000300 00b0200e000000100$	

B The Differential-Linear Distinguishers of KNOT256

In Table 13, for each bias in Table 9, we provide their exact differences and masks. Note that in the hexadecimal presentation of states, the top row denotes a_0, the bottom row denotes a_3, and the arrangement of S-boxes is from right to left.

Table 13: The differential-linear distinguishers of KNOT256

RN	the differential-linear distinguisher	biases
9	$r(\Delta_O \xrightarrow{R_m=9} \lambda_I) = 2^{-1.20}$	$2^{-1.20}$
	$\Delta_O = \begin{matrix}0000000000000000\\0000000000000000\\0000000000000001\\0000000000000001\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000100\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
10	$p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-2},\ r(\Delta_O \xrightarrow{R_m=9} \lambda_I) = 2^{-1.66}$	$2^{-3.66}$
	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0000000000000001\\0000000000000001\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000000\\0000000000000000\\0000000000000000\\0000000000000001\end{matrix}$ $\lambda_I = \begin{matrix}0100000000000000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
11	$p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-2},\ r(\Delta_O \xrightarrow{R_m=9} \lambda_I) = 2^{-2.38},\ q(\lambda_I \xrightarrow{R_1=1} \lambda_O) = 2^{-2}$	$2^{-6.38}$

	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0000000000000001\\0000000000000001\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000001\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000000\\0000000000020000\\0000000000000000\\0000000000020000\end{matrix}$ $\lambda_O = \begin{matrix}0000000000020000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
12	$p(\Delta_I \xrightarrow{R_0=2} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=9} \lambda_I) = 2^{-2.27},\ q(\lambda_I \xrightarrow{R_1=1} \lambda_O) = 2^{-2}$	$2^{-9.27}$
	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0100000000000000\\0100000000000000\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000000\\0000000000000000\\0000000000000001\\0000000000000000\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000000\\0000000200000000\\0000000000000000\\0000000200000000\end{matrix}$ $\lambda_O = \begin{matrix}0000000200000000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
13	$p(\Delta_I \xrightarrow{R_0=3} \Delta_O) = 2^{-8},\ r(\Delta_O \xrightarrow{R_m=9} \lambda_I) = 2^{-2.27},\ q(\lambda_I \xrightarrow{R_1=1} \lambda_O) = 2^{-2}$	$2^{-12.27}$
	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0080000000000000\\0080000000000000\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000000\\0000000000000000\\0000000000000001\\0000000000000000\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000000\\0000000200000000\\0000000000000000\\0000000200000000\end{matrix}$ $\lambda_O = \begin{matrix}0000000200000000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
14	$p(\Delta_I \xrightarrow{R_0=4} \Delta_O) = 2^{-10},\ r(\Delta_O \xrightarrow{R_m=9} \lambda_I) = 2^{-2.23},\ q(\lambda_I \xrightarrow{R_1=1} \lambda_O) = 2^{-3}$	$2^{-16.23}$
	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0040000000000000\\0040000000000000\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000001\\0000000000000000\\0000000001000000\\0000000000000000\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000000\\0000010000000000\\0000000000000000\\0000000000000000\end{matrix}$ $\lambda_O = \begin{matrix}0000010000000000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
15	$p(\Delta_I \xrightarrow{R_0=4} \Delta_O) = 2^{-10},\ r(\Delta_O \xrightarrow{R_m=8} \lambda_I) = 2^{-3.31},\ q(\lambda_I \xrightarrow{R_1=3} \lambda_O) = 2^{-6}$	$2^{-23.31}$
	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0040000000000000\\0040000000000000\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000001\\0000000000000000\\0000000001000000\\0000000000000000\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000000\\0000000000020000\\0100000000000000\\0100000000020000\end{matrix}$ $\lambda_O = \begin{matrix}0000000002000000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	
16	$p(\Delta_I \xrightarrow{R_0=4} \Delta_O) = 2^{-10},\ r(\Delta_O \xrightarrow{R_m=8} \lambda_I) = 2^{-2.52},\ q(\lambda_I \xrightarrow{R_1=4} \lambda_O) = 2^{-10}$	$2^{-30.52}$
	$\Delta_I = \begin{matrix}0000000000000000\\0000000000000000\\0040000000000000\\0040000000000000\end{matrix}$ $\Delta_O = \begin{matrix}0000000000000001\\0000000000000000\\0000000001000000\\0000000000000000\end{matrix}$ $\lambda_I = \begin{matrix}0000000000000000\\0000000000020000\\0100000000000000\\0100000000020000\end{matrix}$ $\lambda_O = \begin{matrix}0000000000010000\\0000000000000000\\0000000000000000\\0000000000000000\end{matrix}$	

C The Differential-Linear Distinguishers of AES

In Table 14, for each bias in Table 10, we provide a few DL distinguishers, where the arrangement follows the design document of AES.

Table 14: The DL distinguishers of AES

#R	the differential-linear distinguisher	bias
	$r(\Delta_O \xrightarrow{R_m=2.5} \lambda_I) = 2^{-8.66},\ q(\lambda_I \xrightarrow{R_1=0.5} \lambda_O) = 2^{-1}$	
	$\Delta_O = \begin{bmatrix}7b & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$ $\lambda_I = \begin{bmatrix}35 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$ $\lambda_O = \begin{bmatrix}0d & 0 & 0 & 0\\ ec & 0 & 0 & 0\\ 52 & 0 & 0 & 0\\ 86 & 0 & 0 & 0\end{bmatrix}$	

3 rounds $2^{-8.66}$

	$\Delta_O = \begin{bmatrix} 83&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_I = \begin{bmatrix} 0&e6&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_O = \begin{bmatrix} 0&13&0&0\\ 0&40&0&0\\ 0&22&0&0\\ 0&97&0&0 \end{bmatrix}$	
	$\Delta_O = \begin{bmatrix} a6&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_I = \begin{bmatrix} 0&96&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_O = \begin{bmatrix} 0&85&0&0\\ 0&7d&0&0\\ 0&24&0&0\\ 0&4a&0&0 \end{bmatrix}$	
	$\Delta_O = \begin{bmatrix} b4&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_I = \begin{bmatrix} 0&0&67&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_O = \begin{bmatrix} 0&0&5c&0\\ 0&0&bc&0\\ 0&0&f5&0\\ 0&0&72&0 \end{bmatrix}$	
	$\Delta_O = \begin{bmatrix} b4&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_I = \begin{bmatrix} 0&0&0&d4\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_O = \begin{bmatrix} 0&0&0&34\\ 0&0&0&9d\\ 0&0&0&53\\ 0&0&0&2e \end{bmatrix}$	
4 rounds	$r(\Delta_O \xrightarrow{R_m=2.5} \lambda_I) = 2^{-21.85},\ q(\lambda_I \xrightarrow{R_1=1.5} \lambda_O) = 2^{-4}$	
	$\Delta_O = \begin{bmatrix} 1&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_I = \begin{bmatrix} 2&0&0&0\\ 0&3&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{bmatrix}$ $\lambda_O = \begin{bmatrix} 38&0&0&0\\ 2c&0&0&0\\ 34&0&0&0\\ 24&0&0&0 \end{bmatrix}$	$2^{-27.85}$
5 rounds	$p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-24},\ r(\Delta_O \xrightarrow{R_m=2.5} \lambda_I) = 2^{-21.85},\ q(\lambda_I \xrightarrow{R_1=1.5} \lambda_O) = 2^{-4}$	
	$\Delta_I = \begin{bmatrix} b3&0&0&0\\ 0&58&0&0\\ 0&0&45&0\\ 0&0&0&0f \end{bmatrix}$ $\Delta_O = \begin{bmatrix} 1&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$ $\lambda_I = \begin{bmatrix} 2&0&0&0\\ 0&3&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{bmatrix}$ $\lambda_O = \begin{bmatrix} 38&0&0&0\\ 2c&0&0&0\\ 34&0&0&0\\ 24&0&0&0 \end{bmatrix}$	$2^{-51.85}$

D The Differential-Linear Distinguishers of CLEFIA

In Table 15, for each bias in Table 11, we provide a few DL distinguishers, where the first column presents the input differences, and the second column presents the output masks. In the hexadecimal presentation of states, the arrangement is from left to right. Every two nibbles represent an S-box.

Table 15: The differential-linear distinguishers of CLEFIA

#R	the differential-linear distinguisher	bias
5	$r(\Delta_O \xrightarrow{R_m=5} \lambda_I) = 2^{-2.54}$	$2^{-2.54}$
	$\Delta_O = 0x000000000000009c0000000000000000 \quad \lambda_I = 0x000000002c3a0b160000000000000000$	

	$\Delta_O = 0x000000000009c00000000000000000000 \quad \lambda_I = 0x000000000b162c3a0000000000000000$	
6	$p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=5} \lambda_I) = 2^{-2.54}$	$2^{-7.54}$
	$\Delta_I = 0x0000000000000000009c0000452b1356 \quad \Delta_O = 0x000000000000009c0000000000000000$ $\lambda_I = 0x000000002c3a0b160000000000000000$	
	$\Delta_I = 0x00000000000000000000009c1356452b \quad \Delta_O = 0x000000000009c0000000000000000000$ $\lambda_I = 0x000000000b162c3a0000000000000000$	
7	$p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=5} \lambda_I) = 2^{-2.54},\ q(\lambda_I \xrightarrow{R_1=1} \lambda_O) = 2^{-3.415}$	$2^{-12.37}$
	$\Delta_I = 0x0000000000000000009c0000452b1356 \quad \Delta_O = 0x000000000000009c0000000000000000$ $\lambda_I = 0x000000002c3a0b160000000000000000 \quad \lambda_O = 0x0b162c3a00000000000000004d000000$	
	$\Delta_I = 0x00000000000000000000009c1356452b \quad \Delta_O = 0x000000000009c0000000000000000000$ $\lambda_I = 0x000000000b162c3a0000000000000000 \quad \lambda_O = 0x2c3a0b160000000000000000004d00$	
8	$p(\Delta_I \xrightarrow{R_0=1} \Delta_O) = 2^{-5},\ r(\Delta_O \xrightarrow{R_m=5} \lambda_I) = 2^{-2.54},\ q(\lambda_I \xrightarrow{R_1=2} \lambda_O) = 2^{-14.02}$	$2^{-33.59}$
	$\Delta_I = 0x0000000000000000009c0000452b1356 \quad \Delta_O = 0x000000000000009c0000000000000000$ $\lambda_I = 0x000000002c3a0b160000000000000000 \quad \lambda_O = 0x00000000535d01ee8e0000000b162c3a$	
	$\Delta_I = 0x00000000000000000000009c1356452b \quad \Delta_O = 0x000000000009c0000000000000000000$ $\lambda_I = 0x000000000b162c3a0000000000000000 \quad \lambda_O = 0x0000000001ee535d00008e002c3a0b16$	
9	$p(\Delta_I \xrightarrow{R_0=2} \Delta_O) = 2^{-27.415},\ r(\Delta_O \xrightarrow{R_m=5} \lambda_I) = 2^{-2.83},\ q(\lambda_I \xrightarrow{R_1=2} \lambda_O) = 2^{-13.80}$	$2^{-55.84}$
	$\Delta_I = 0xba501aa0cf68a5fa00000000000b10000 \quad \Delta_O = 0x000000000b1000000000000000000000$ $\lambda_I = 0x00000000625396310000000000000000 \quad \lambda_O = 0x00000000ef6904b90000ee0062539631$	
	$\Delta_I = 0x82572cae90838bb10000000000b10000 \quad \Delta_O = 0x000000000b1000000000000000000000$ $\lambda_I = 0x00000000625396310000000000000000 \quad \lambda_O = 0x00000000ef6904b90000ee0062539631$	
	$\Delta_I = 0x1aa0ba50a5facf6800000000000000b1 \quad \Delta_O = 0x00000000000000b100000000000000000$ $\lambda_I = 0x00000009631625300000000000000000 \quad \lambda_O = 0x0000000004b9ef69ee00000096316253$	
	$\Delta_I = 0x2cae82578bb1908300000000000000b1 \quad \Delta_O = 0x00000000000000b100000000000000000$ $\lambda_I = 0x00000009631625300000000000000000 \quad \lambda_O = 0x0000000004b9ef69ee00000096316253$	