Des. Codes Cryptogr. (2015) 77:11–48
DOI 10.1007/s10623-014-9985-x

A methodology for differential-linear cryptanalysis and its applications

Jiqiang Lu

Received: 13 September 2012 / Revised: 16 January 2014 / Accepted: 21 May 2014 /
Published online: 7 June 2014

Abstract Differential and linear cryptanalyses are powerful techniques for analysing the security of a block cipher. In 1994 Langford and Hellman published a combination of differential and linear cryptanalysis under two default independence assumptions, known as differential-linear cryptanalysis, which is based on the use of a differential-linear distinguisher constructed by concatenating a linear approximation with a (truncated) differential with probability 1. In 1995 Langford gave a general version of differential-linear cryptanalysis, so that a differential with a probability smaller than 1 can also be used to construct a differential-linear distinguisher; the general version was published in 2002 by Biham, Dunkelman and Keller with an elaborate explanation using an additional assumption. In this paper, we introduce a new methodology for differential-linear cryptanalysis under the original two assumptions, without using the additional assumption of Biham et al. The new methodology is more reasonable and more general than Langford and Biham et al.'s methodology; and apart from this advantage it can lead to some better cryptanalytic results than Langford and Biham et al.'s methodology and Langford and Hellman's methodology. As examples, we apply it to 13 rounds of the DES block cipher, 10 rounds of the CTC2 block cipher and 12 rounds of the Serpent block cipher. The new methodology can be used to cryptanalyse other block ciphers, and block cipher designers should pay attention to this new methodology when designing a block cipher.

Communicated by V. Rijmen and C. Mitchell.

An earlier version of this work appeared in the pre-proceedings of the 19th International Workshop on Fast Software Encryption (FSE 2012, Washington, USA, 19–21 March 2012), an extended abstract of which was published in the post-proceedings of FSE 2012 [40]; and an even earlier version of this work appeared in 2010 as part of Cryptology ePrint Archive Report 2010/025 [39] while the author was with Eindhoven University of Technology (The Netherlands).

J. Lu (✉)
Institute for Infocomm Research, Agency for Science, Technology and Research,
1 Fusionopolis Way, Singapore 138632, Singapore
e-mail: lvjiqiang@hotmail.com; jlu@i2r.a-star.edu.sg

Keywords Cryptology · Block cipher · CTC2 · DES · Serpent · Differential-linear cryptanalysis

Mathematics Subject Classification 94A60

1 Introduction

The block cipher is an important primitive in secret-key cryptography. A block cipher is an algorithm that transforms a fixed-length data block, called a plaintext block, into another data block of the same length, called a ciphertext block, under the control of a secret user key. One main purpose of a block cipher is to provide confidentiality for data transmitted in insecure communication environments. A block cipher can be used to build other secret-key cryptographic primitives, such as hash functions, message authentication codes and cryptographic pseudorandom number generators. Block ciphers are also often used as a component in various security areas. In practice, almost all block ciphers are constructed by iterating a simple function many times, known as the iterated method, and the repeated function is called the round function. Most block ciphers are examples of one of two special types of iterated ciphers, known as Feistel ciphers and Substitution-Permutation Networks (SPNs). In a Feistel cipher, the plaintext is split into two halves. The round function is applied to one half, and the output of the round function is bitwise exored with the other half; finally, the two halves are swapped, and become the two halves of the next round. In an SPN cipher, the round function is applied to the whole block, and its output becomes the input of the next round. The Data Encryption Standard (DES) [44] block cipher is an example of a Feistel cipher, and the CTC2 [16] and Serpent [1,2] block ciphers are examples of an SPN. DES has a 64-bit block size, a 56-bit user key, and a total of 16 rounds; CTC2 has a variable block size, a variable length key and a variable number of rounds; and Serpent has a 128-bit block size, a variable length user key of up to 256 bits, and a total of 32 rounds.

Differential cryptanalysis was introduced in 1990 by Biham and Shamir [5], which is based on the use of one or more so-called differentials. Linear cryptanalysis was introduced in 1992 by Matsui and Yamagishi [43], which is based on the use of one or more so-called linear approximations. Both the cryptanalytic methods were used to attack the full DES cipher faster than exhaustive key search [7,41,42]. In the field of block cipher cryptanalysis, a cryptanalytic attack is commonly regarded as effective if it is faster (i.e. it has lower time complexity) than exhaustive key search and requires a smaller memory than the dictionary attack.

In 1994 Langford and Hellman [37] published a combination of differential and linear cryptanalysis under two default independence assumptions, known as differential-linear cryptanalysis, and they applied it to break 8-round DES; and later Langford applied it to attack greater numbers of rounds of DES [36]. Such an attack is constructed on a so-called differential-linear distinguisher; a differential-linear distinguisher treats a block cipher as a cascade of two sub-ciphers, and it uses a linear approximation for a sub-cipher and, for the other sub-cipher it uses a differential (or a truncated differential [31]) with a one probability that does not affect the bit(s) concerned by the input mask of the linear approximation. In 1995 Langford [36] gave a general version of differential-linear cryptanalysis, so that a differential with a probability smaller than 1 can also be used to construct a differential-linear distinguisher. Langford did not give much explanation to this general version, and did not describe any application. The general version was published in 2002 by Biham et al. [10] with an elaborate explanation using an additional assumption, and they applied it to break 9-round DES and mentioned that it could attack DES with up to 10 rounds. Below we refer to

this general version as Langford and Biham et al.'s methodology. Differential-linear cryptanalysis has been used to yield the best currently published cryptanalytic results for a number of state-of-the-art block ciphers [10,12,22,23], and is thus clearly of importance.

In this paper, we present a new methodology for differential-linear cryptanalysis under the two default assumptions implicitly used by Langford and Hellman, without using the additional assumption due to Biham et al. The new methodology is more reasonable and more general than Langford and Biham et al.'s methodology, and it can lead to some better cryptanalytic results than Langford and Biham et al.'s and Langford and Hellman's methodologies. As examples, we apply the new methodology to mount differential-linear attacks on 10-round CTC2 with a 255-bit block size and key, 13-round DES and 12-round Serpent. In particular, in terms of the numbers of attacked rounds, the 10-round CTC2 attack is the first published cryptanalytic result on this version of CTC2, and the 12-round Serpent attack matches the best previously published cryptanalytic result for Serpent, that was obtained under Langford and Biham et al.'s methodology. Table 1 summarises both our and previous main cryptanalytic results on CTC2, DES and Serpent.

The remainder of the paper is organised as follows. In the next section we give the notation used throughout the paper and briefly describe differential and linear cryptanalysis. In Sect. 3 we review Langford and Hellman's and Langford and Biham et al.'s methodologies and give our methodology for differential-linear cryptanalysis. In Sects. 4–6 we present our cryptanalytic results on DES, CTC2 and Serpent, respectively. In Sect. 7 we discuss a few possible extensions and variants of our methodology. Section 8 concludes this paper.

2 Preliminaries

In this section we describe the notation, differential and linear cryptanalysis.

2.1 Notation

In the following descriptions, we assume that a number without a prefix is in decimal notation, and a number with prefix $0x$ is in hexadecimal notation, unless otherwise stated. The bits of a value are numbered from right to left, the leftmost bit is the most significant bit, and the rightmost bit is the least significant bit, except in the case of DES, where we use the same numbering notation as in FIPS-46 [44]. We use the following notation.

$\oplus$	Bitwise logical exclusive OR (XOR) of two bit strings of the same length
$\odot$	Dot product of two bit strings of the same length
$\|\|$	String concatenation
$\ll$	Left shift of a bit string
$\lll$	Left rotation of a bit string
$\circ$	Functional composition. When composing functions X and Y, X $\circ$ Y denotes the function obtained by first applying X and then applying Y
e_j	A 255-bit value with zeros everywhere except for bit position j, $(0 \leq j \leq 254)$
$e_{i_0,\cdots,i_j}$	The 255-bit value equal to $e_{i_0} \oplus \cdots \oplus e_{i_j}$, $(0 \leq i_0, \cdots, i_j \leq 254)$
$\mathbb{E}$	An n-bit block cipher when used with a specific user key
$\Phi(\cdot)$	The cumulative distribution function of the standard normal distribution

Table 1 Our and previous main cryptanalytic results on CTC2, DES and Serpent

Cipher	Attack technique	Rounds	Data	Memory	Time	Success rate	Source
CTC2	Algebraic [17]	6	4CP	[a]	2^{253}Enc.	[a]	[16,22]
(255-bit	Differential	8[b]	2^{60}CP	[a]	2^{60}Enc.[c]	[a]	[22]
version)	Differential-linear	8[b]	2^{37}CP	2^{12}	2^{37}Enc.[c]	70 %	[22]
		10	$2^{144.84}$CP	$2^{54.4}$	2^{207}Enc.	99 %	Sect. 5.4
DES	Differential	Full	$2^{47.2}$CP	2^{14}	$2^{47.2}$Enc.	60 %	[7]
	Linear	Full	2^{43}KP	$2^{14.3}$	2^{43}Enc.	85 %	[42]
		Full	2^{42}CP	$2^{15.2}$	2^{42}Enc.	86 %	[32]
	Davies' attack [19,20]	Full	2^{50}KP	2^{30}	2^{50}Enc.	51 %	[4]
		Full	2^{45}CP	2^{36}	2^{45}Enc.	50 %	[34]
	Differential-linear	8	768CP	2^{11}	$2^{14.6}$Enc.[c]	95 %	[37]
		9	$2^{15.75}$CP	$2^{19.75}$	$2^{29.17}$Enc.[c]	89 %	[10,21]
		10	$2^{29.46}$CP	$2^{11.9}$	2^{46}Enc.	99 %	Sect. 4.4
		12	2^{30}CP	2^{30}	$2^{30.8}$Enc.[c]	73 %	[36]
		13	$2^{52.1}$CP	$2^{27.4}$	$2^{54.2}$Enc.	99 %	Sect. 4.5
Serpent	Differential	8	2^{84}CP	2^{88}	2^{213}MA	[a]	[8]
	Amplified boomerang [28]	9	2^{110}CP	2^{212}	2^{252}Enc.	[a]	[33]
	Boomerang [48]	10	2^{128}KP	2^{100}	$2^{165.3}$Enc.	[a]	[11]
	Rectangle [8]	10	$2^{126.3}$CP	$2^{130.3}$	$2^{165.3}$Enc.	[a]	[11]
	Linear	11	2^{118}KP	2^{88}	2^{178}Enc.	[a]	[14]
	Differential-linear	11	$2^{113.7}$CP	2^{99}	$2^{137.7}$MA	93 %	[23]
		12	$2^{123.5}$CP	$2^{128.5}$	$2^{249.4}$Enc.	84 %	[23]
		12	$2^{124.5}$CP	$2^{129.5}$	$2^{244.9}$Enc.	99 %	Sect. 6.3

CP chosen plaintexts, *KP* known plaintexts, *MA* memory accesses, *Memory unit* bytes, *Enc.* encryption operations of the relevant version of CTC2, DES or Serpent

[a] This term was not specified

[b] There is a flaw (see Sect. 5.2)

[c] The complexity is for retrieving only a portion of the full key bits

2.2 Differential cryptanalysis

Differential cryptanalysis [5] takes advantage of how a specific difference in a pair of inputs of a cipher can affect a difference in the pair of outputs of the cipher, where the pair of outputs are obtained by encrypting the pair of inputs using the same key. The notion of difference can be defined in several ways; the most widely discussed is with respect to the XOR operation. The difference between the inputs is called the input difference, and the difference between the outputs of a function is called the output difference. The combination of the input difference and the output difference is called a differential. The probability of a differential is defined as follows.

Definition 1 (*from* [38]) If α and β are n-bit blocks, then the probability of the differential (α, β) for $\mathbb{E}$, written $\Delta\alpha \rightarrow \Delta\beta$, is defined to be

$$\Pr_{\mathbb{E}}(\Delta\alpha \rightarrow \Delta\beta) = \Pr_{P \in \{0,1\}^n}(\mathbb{E}(P) \oplus \mathbb{E}(P \oplus \alpha) = \beta).$$

The following result follows trivially from Definition 1:

Proposition 1 (from [38]) *If α and β are n-bit blocks, then*

$$\Pr_{\mathbb{E}}(\Delta\alpha \rightarrow \Delta\beta) = \frac{|\{x|\mathbb{E}(x) \oplus \mathbb{E}(x \oplus \alpha) = \beta, x \in \{0,1\}^n\}|}{2^n}.$$

For a random permutation, the expected probability of a differential for any pair (α, β) is 2^{-n}. Therefore, if $\Pr_{\mathbb{E}}(\Delta\alpha \rightarrow \Delta\beta)$ is larger than 2^{-n}, we can use the differential to distinguish $\mathbb{E}$ from a random permutation, given a sufficient number of chosen plaintext pairs.

2.3 Linear cryptanalysis

Linear cryptanalysis [41,43] exploits correlations between a particular linear function of the input blocks and a second linear function of the output blocks. The combination of the two linear functions is called a linear approximation. The most widely used linear function involves computing the bitwise dot product operation of the block with a specific binary vector (the specific value combined with the input blocks may be different from the value applied to the output blocks). The value combined with the input blocks is called the input mask, and the value applied to the output blocks is called the output mask. The probability of a linear approximation is defined as follows.

Definition 2 (from [38]) If α and β are n-bit blocks, then the probability of the linear approximation (α, β) for $\mathbb{E}$, written $\Gamma\alpha \rightarrow \Gamma\beta$, is defined to be

$$\Pr_{\mathbb{E}}(\Gamma\alpha \rightarrow \Gamma\beta) = \Pr_{P \in \{0,1\}^n}(P \odot \alpha = \mathbb{E}(P) \odot \beta).$$

We refer to below the dot product $P \odot \alpha$ as the input parity, and the dot product $\mathbb{E}(P) \odot \beta$ as the output parity. The following result follows trivially from Definition 2:

Proposition 2 *(*from [38]) *If α and β are n-bit blocks, then*

$$\Pr_{\mathbb{E}}(\Gamma\alpha \rightarrow \Gamma\beta) = \frac{|\{x|x \odot \alpha = \mathbb{E}(x) \odot \beta, x \in \{0,1\}^n\}|}{2^n}.$$

For a random permutation, the expected probability of a linear approximation for any pair (α, β) is $\frac{1}{2}$. The bias of a linear approximation $\Gamma\alpha \rightarrow \Gamma\beta$, denoted by ϵ, is defined to be $\epsilon = |\Pr_{\mathbb{E}}(\Gamma\alpha \rightarrow \Gamma\beta) - \frac{1}{2}|$. Thus, if the bias ϵ is sufficiently large, we can use the linear approximation to distinguish $\mathbb{E}$ from a random permutation, given a sufficient number of matching plaintext-ciphertext pairs.

It is particularly worthy to note that in 2008 Selçuk [46] formulated the success probability of a linear cryptanalysis attack, as follows.

Theorem 1 (interpreted from [46]) *For a linear attack on m key bits that uses a linear approximation with probability p and N known plaintexts and ranks the correct m-bit key value among the top r out of the 2^m possible key values, if m and N are sufficiently large, then under the assumption that the linear approximation's probability to hold is independent for each key tried and is equal to $\frac{1}{2}$ for all wrong key values, the success probability of the linear attack, denoted by P_S, is*

$$P_S = \Phi\left(|2p-1|\sqrt{N} - \Phi^{-1}\left(1 - \frac{r}{2^{m+1}}\right)\right),$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

2.4 General assumptions used in practice

Propositions 1 and 2 give the accurate probability values of a differential and a linear approximation from a theoretical point of view. However, it is usually hard to apply them in practice to a block cipher with a large block size, for example, $n = 64$ or 128 which is currently being widely used in reality, and even harder when the differential or linear approximation operates on many rounds of the cipher. In practice, for a Markov block cipher [35], a multi-round differential characteristic (or linear approximation) is usually obtained by concatenating a few one-round differential characteristics (respectively, linear approximations), and the probability of the multi-round differential (or linear approximation) is regarded as the product (respectively, the piling-up function [41]) of the probabilities of the one-round differential characteristics (respectively, linear approximations) under the following Assumption 1.

Assumption 1 (*interpreted from* [35]) The inputs as well as the subkeys to the involved rounds are independent and uniformly distributed, and the probability of a differential or linear approximation is approximately identical for almost all subkey values.

We refer the reader to [35] for a mathematical description of Assumption 1 with respect to differential cryptanalysis, (in particular, the second part of Assumption 1 is known as the stochastic equivalence assumption [35]). We note that one may argue the correctness of Assumption 1 and may use a different assumption, for the round keys are actually dependent generally, being generated from a global user key under the key schedule algorithm of the cipher. Anyway, all such assumptions require us to treat the involved rounds as independent. As mentioned in [25], this is "most often not exactly the case, but as often it is a good approximation".

Differential and linear cryptanalyses generally treat a basic unit of input (i.e. a chosen-plaintext pair for differential cryptanalysis; a known-plaintext for linear cryptanalysis) as a random variable, and assume that given a set of inputs of the basic unit, the inputs that satisfy the required property can be approximated by an independent distribution, as followed in [6,41].

3 Differential-linear cryptanalysis: previous and our methodologies

In this section we first review previous methodologies on differential-linear cryptanalysis, namely Langford and Hellman's methodology and Langford and Biham et al.'s methodology, and then give our new methodology, followed by a few implications. First observe that for simplicity we assume that the probability for a linear approximation with bias ϵ is $\frac{1}{2} + \epsilon$ in all the following descriptions; but the same results can be obtained when the probability is $\frac{1}{2} - \epsilon$.

3.1 Langford and Hellman's methodology

In 1994 Langford and Hellman [37] introduced differential-linear cryptanalysis as a combination of differential and linear cryptanalysis, which is based on the use of a differential-linear distinguisher. To construct a differential-linear distinguisher, they treated $\mathbb{E}$ as a cascade of two sub-ciphers $\mathbb{E}_0$ and $\mathbb{E}_1$, where $\mathbb{E} = \mathbb{E}_0 \circ \mathbb{E}_1$. A differential-linear distinguisher uses a (truncated) differential $\Delta\alpha \rightarrow \Delta\beta$ with probability 1 for $\mathbb{E}_0$ and a linear approximation $\Gamma\gamma \rightarrow \Gamma\delta$ with bias ϵ for $\mathbb{E}_1$, where the output difference β of the (truncated) differential has a zero value in the bit positions concerned by the input mask of the linear approximation

(thus $\beta \odot \gamma = 0$ holds). Let P be a plaintext chosen uniformly at random from $\{0, 1\}^n$. Thus, we have $\mathbb{E}_0(P) \odot \gamma = \mathbb{E}_0(P \oplus \alpha) \odot \gamma$ with probability 1, where $\mathbb{E}_0(P)$ and $\mathbb{E}_0(P \oplus \alpha)$ are the results of encrypting P and $P \oplus \alpha$ with $\mathbb{E}_0$, respectively. The differential-linear distinguisher is concerned with the event $\delta \odot \mathbb{E}(P) = \delta \odot \mathbb{E}(P \oplus \alpha)$; and under Assumption 1 and the following Assumption 2 it has a probability of $\Pr(\delta \odot \mathbb{E}(P) = \delta \odot \mathbb{E}(P \oplus \alpha)) = (\frac{1}{2} + \epsilon) \times (\frac{1}{2} + \epsilon) + (\frac{1}{2} - \epsilon) \times (\frac{1}{2} - \epsilon) = \frac{1}{2} + 2\epsilon^2$.

Assumption 2 (*interpreted from* [37]) The two inputs $\mathbb{E}_0(P)$ and $\mathbb{E}_0(P \oplus \alpha)$ of the linear approximation for $\mathbb{E}_1$ behave as independent inputs with respect to the linear approximation.

Note that $\mathbb{E}(P) = \mathbb{E}_1(\mathbb{E}_0(P))$ and $\mathbb{E}(P \oplus \alpha) = \mathbb{E}_1(\mathbb{E}_0(P \oplus \alpha))$ in the above descriptions. Assumption 2 is somewhat like assuming an independent distribution for plaintext pairs generated from a particular plaintext structure with certain property in differential cryptanalysis.

By contrast, for a random permutation, the expected probability of a differential-linear distinguisher is $\frac{1}{2}$. Therefore, if the bias $|\Pr(\delta \odot \mathbb{E}(P) = \delta \odot \mathbb{E}(P \oplus \alpha)) - \frac{1}{2}| = 2\epsilon^2$ is sufficiently large, we can distinguish $\mathbb{E}$ from a random permutation.

3.2 Langford and Biham et al.'s methodology

In 1995 Langford [36] gave a general version of differential-linear cryptanalysis, where a differential-linear distinguisher could be built on a differential with a probability smaller than 1; specifically, she wrote: "Suppose that we have a differential characteristic which occurs with probability p_D and a linear relation which hold with probability $p_L = 1/2 + \varepsilon_L$. Then the probability that two texts with the correct input pattern will have output parities which agree is $p_{DL} = 1/2 + 2p_D\varepsilon_L^2$." This probability formula is exactly the same as that Biham, Dunkelman and Keller published in 2002, except that different symbols were used. Langford did not give much explanation to this general version, and did not describe any application. The general version was published in 2002 by Biham et al. [10] with an elaborate explanation using an additional assumption, plus a few applications. As mentioned earlier, we refer to this general version as Langford and Biham et al.'s methodology.

Langford and Biham et al.'s general version includes the case when the (truncated) differential $\Delta\alpha \to \Delta\beta$ has a smaller probability than 1, p say, with β meeting the condition $\beta \odot \gamma = 0$.[1] A slightly revised version was given in [21]. Biham et al. applied Langford and Hellman's analysis described above when $\mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta$, and used the following Assumption 3 for the cases where $\mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) \neq \beta$:[2]

Assumption 3 (*interpreted from* [10]) The output parities $\delta \odot \mathbb{E}(P)$ and $\delta \odot \mathbb{E}(P \oplus \alpha)$ have a uniform and independent distribution in $\{0, 1\}$ for the cases where $\mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) \neq \beta$.

As a result, under Assumptions 1, 2 and 3, Biham et al. got $\Pr(\delta \odot \mathbb{E}(P) = \delta \odot \mathbb{E}(P \oplus \alpha)) = p \times (\frac{1}{2} + 2\epsilon^2) + (1 - p) \times \frac{1}{2} = \frac{1}{2} + 2p\epsilon^2$.

Finally, Biham et al. concluded that if the bias $2p\epsilon^2$ is sufficiently large, the distinguisher can be used as the basis of a differential-linear attack to distinguish $\mathbb{E}$ from a random permutation. Roughly, the attack has a data complexity of about $O(p^{-2}\epsilon^{-4})$.

[1] A more general condition is $\beta \odot \gamma = c$, where $c \in \{0, 1\}$ is a constant. Without loss of generality, we consider the case with $c = 0$ throughout this paper.

[2] We note that Biham et al. used a different assumption when reviewing the enhanced version in a few other papers, [13] say, where they assumed that $\mathbb{E}_0(P) \odot \gamma = \mathbb{E}_0(P \oplus \alpha) \odot \gamma$ holds with half a chance for the cases where $\mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) \neq \beta$, yielding the same probability value $\frac{1}{2} + 2p\epsilon^2$ as under Assumption 3. We treat this assumption as Assumption 3, though they are different.

3.3 Our methodology

In summary, the differential-linear distinguishers described above are concerned with the correlation between a pair of output parities, where the pair of output parities are obtained by applying a linear function (e.g. bitwise dot product with δ) to the outputs of a pair of input blocks with difference α (under the same key). The combination of the input difference and the linear function is called a differential-linear distinguisher. More formally, we define the probability of the differential-linear distinguisher as follows.

Definition 3 If α and δ are n-bit blocks, then the probability of the differential-linear distinguisher (α, δ) for $\mathbb{E}$, written $\Delta\alpha \to \Gamma\delta$, is defined to be

$$\Pr_{\mathbb{E}}(\Delta\alpha \to \Gamma\delta) = \Pr_{P\in\{0,1\}^n}(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta).$$

The following result follows trivially from Definition 3:

Proposition 3 *If α and δ are n-bit blocks, then*

$$\Pr_{\mathbb{E}}(\Delta\alpha \to \Gamma\delta) = \frac{|\{x|\mathbb{E}(x) \odot \delta = \mathbb{E}(x \oplus \alpha) \odot \delta, x \in \{0,1\}^n\}|}{2^n}.$$

For a random permutation, the expected probability of a differential-linear distinguisher for any combination (α, δ) is $\frac{1}{2}$. The bias of the differential-linear distinguisher $\Delta\alpha \to \Gamma\delta$ is defined to be $|\Pr_{\mathbb{E}}(\Delta\alpha \to \Gamma\delta) - \frac{1}{2}|$. Thus, if the bias is sufficiently large, we can use the differential-linear distinguisher to distinguish $\mathbb{E}$ from a random permutation, given a sufficient number of chosen plaintext pairs.

In practice, it is usually infeasible to compute the accurate probability of a differential-linear distinguisher $\Delta\alpha \to \Gamma\delta$ by Proposition 3, and we have to make use of some assumptions to approximate it, like Langford and Biham et al.'s methodology described in Sect. 3.2. However, Langford and Biham et al.'s methodology uses the three assumptions as hypotheses and works only when Assumption 3 holds; otherwise it may give probability values that are highly inaccurate in some situations; for example, let's intuitively consider the naive situation where the differential $\Delta\alpha \to \Delta\beta$ has probability $\frac{1}{2}$ and meets $\beta \odot \gamma = 0$, and all the other possible differentials $\{\Delta\alpha \to \Delta\widehat{\beta}\}$ meet $\widehat{\beta} \odot \gamma = 1$. Such an example can be easily built for a practical block cipher, DES say. The differential $\Delta\alpha \to \Delta\beta$ contributes $\frac{1}{2}[(\frac{1}{2}+\epsilon)\times(\frac{1}{2}+\epsilon)+(\frac{1}{2}-\epsilon)\times(\frac{1}{2}-\epsilon)] = \frac{1}{4}+\epsilon^2$ to the probability of the distinguisher, and the other differentials $\{\Delta\alpha \to \Delta\widehat{\beta}\}$ contribute $\frac{1}{2}[(\frac{1}{2}+\epsilon)\times(\frac{1}{2}-\epsilon)+(\frac{1}{2}-\epsilon)\times(\frac{1}{2}+\epsilon)] = \frac{1}{4}-\epsilon^2$, which also cause a bias, but in a negative way, canceling the bias due to $\Delta\alpha \to \Delta\beta$. So the real bias of the distinguisher is 0, that is, the distinguisher has no cryptanalytic significance (in the sense of Langford and Biham et al.'s methodology). But if we applied Langford and Biham et al.'s methodology in this situation, the distinguisher would seem to have a bias of $2 \times \frac{1}{2} \times \epsilon^2 = \epsilon^2$, and thus the distinguisher would be useful (if ϵ^2 is large enough); but it is useless in fact. Notice that this case is not truly a counterexample to Langford and Biham et al.'s methodology, for it is clear that Assumption 3 does not hold for it, and one cannot apply Langford and Biham et al.'s methodology, but it suggests that we should be cautious about using Assumption 3 and actually, we should be careful with using any assumption, and it is preferable to use as few assumptions as possible.

Biham, Dunkelman and Keller used a heuristic way to approximate the probability of a differential-linear distinguisher. We make an analysis for the probability of a differential-linear distinguisher from a mathematical point, and obtain a new methodology under only Assumptions 1 and 2. Our result is given as Theorem 2, followed by a proof.

Theorem 2 *An n-bit block cipher $\mathbb{E}$ is represented as a cascade of two sub-ciphers $\mathbb{E}_0$ and $\mathbb{E}_1$, where $\mathbb{E} = \mathbb{E}_0 \circ \mathbb{E}_1$. If α ($\neq 0$) is an input difference for $\mathbb{E}_0$, $\Gamma\gamma \rightarrow \Gamma\delta$ is a linear approximation with bias ϵ for $\mathbb{E}_1$, and the sum of the probabilities for the differentials $\{\Delta\alpha \rightarrow \Delta\beta | \Pr_{\mathbb{E}_0}(\Delta\alpha \rightarrow \Delta\beta) > 0, \gamma \odot \beta = 0, \beta \in \{0,1\}^n\}$ is $\widehat{p}$ ($= \sum_{\beta \in \{0,1\}^n, \gamma \odot \beta = 0} \Pr_{\mathbb{E}_0}(\Delta\alpha \rightarrow \Delta\beta)$), then under Assumptions 1 and 2 the probability of the differential-linear distinguisher $\Delta\alpha \rightarrow \Gamma\delta$ is*

$$\Pr_{P \in \{0,1\}^n}(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta) = \frac{1}{2} + 2(2\widehat{p} - 1)\epsilon^2.$$

Proof Given the input difference α for $\mathbb{E}_0$, there are one or more possible output differences $\{\beta | \Pr_{\mathbb{E}_0}(\Delta\ \alpha \rightarrow \Delta\beta) > 0, \beta \in \{0,1\}^n\}$; these output differences can be classified into two sets: one is $\{\beta | \gamma \odot \beta = 0, \Pr_{\mathbb{E}_0}(\Delta\alpha \rightarrow \Delta\beta) > 0, \beta \in \{0,1\}^n\}$, and the other is $\{\beta | \gamma \odot \beta = 1, \Pr_{\mathbb{E}_0}(\Delta\alpha \rightarrow \Delta\beta) > 0, \beta \in \{0,1\}^n\}$.

Let P be a plaintext chosen uniformly at random from $\{0,1\}^n$. Then, under Assumptions 1 and 2 we have

$$\begin{aligned}
&\Pr(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta | \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta, \gamma \odot \beta = 0) \\
&= \Pr(\mathbb{E}_0(P) \odot \gamma = \mathbb{E}(P) \odot \delta, \mathbb{E}_0(P \oplus \alpha) \odot \gamma = \mathbb{E}(P \oplus \alpha) \odot \delta | \\
&\quad \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta, \gamma \odot \beta = 0) \\
&\quad + \Pr(\mathbb{E}_0(P) \odot \gamma \neq \mathbb{E}(P) \odot \delta, \mathbb{E}_0(P \oplus \alpha) \odot \gamma \neq \mathbb{E}(P \oplus \alpha) \odot \delta | \\
&\quad \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta, \gamma \odot \beta = 0) \\
&= \left(\frac{1}{2} + \epsilon\right) \times \left(\frac{1}{2} + \epsilon\right) + \left[1 - \left(\frac{1}{2} + \epsilon\right)\right] \times \left[1 - \left(\frac{1}{2} + \epsilon\right)\right] \\
&= \frac{1}{2} + 2\epsilon^2,
\end{aligned}$$

and

$$\begin{aligned}
&\Pr(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta | \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta, \gamma \odot \beta = 1) \\
&= \Pr(\mathbb{E}_0(P) \odot \gamma = \mathbb{E}(P) \odot \delta, \mathbb{E}_0(P \oplus \alpha) \odot \gamma \neq \mathbb{E}(P \oplus \alpha) \odot \delta | \\
&\quad \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta, \gamma \odot \beta = 1) \\
&\quad + \Pr(\mathbb{E}_0(P) \odot \gamma \neq \mathbb{E}(P) \odot \delta, \mathbb{E}_0(P \oplus \alpha) \odot \gamma = \mathbb{E}(P \oplus \alpha) \odot \delta | \\
&\quad \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta, \gamma \odot \beta = 1) \\
&= \left(\frac{1}{2} + \epsilon\right) \times \left[1 - \left(\frac{1}{2} + \epsilon\right)\right] + \left[1 - \left(\frac{1}{2} + \epsilon\right)\right] \times \left(\frac{1}{2} + \epsilon\right) \\
&= \frac{1}{2} - 2\epsilon^2.
\end{aligned}$$

Next, under Assumptions 1 and 2 we can compute the probability of the differential-linear distinguisher as follows.

$$\begin{aligned}
&\Pr(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta) \\
&= \sum_{\beta \in \{0,1\}^n, Y \in \{0,1\}} \Pr(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta, \mathbb{E}_0(P) \odot \gamma \oplus \\
&\quad \mathbb{E}_0(P \oplus \alpha) \odot \gamma = Y, \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta) \\
&= \sum_{\beta \in \{0,1\}^n, Y \in \{0,1\}} \Pr(\mathbb{E}(P) \odot \delta = \mathbb{E}(P \oplus \alpha) \odot \delta | \mathbb{E}_0(P) \odot \gamma \oplus \\
&\quad \mathbb{E}_0(P \oplus \alpha) \odot \gamma = Y, \mathbb{E}_0(P) \oplus \mathbb{E}_0(P \oplus \alpha) = \beta) \times
\end{aligned}$$

$$
\begin{aligned}
&\Pr(\mathbb{E}_0(P)\odot\gamma\oplus\mathbb{E}_0(P\oplus\alpha)\odot\gamma=Y,\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta)\\
=&\sum_{\beta\in\{0,1\}^n}\Pr(\mathbb{E}(P)\odot\delta=\mathbb{E}(P\oplus\alpha)\odot\delta|\mathbb{E}_0(P)\odot\gamma\oplus\mathbb{E}_0(P\oplus\alpha)\odot\gamma=0,\\
&\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta)\times\Pr(\mathbb{E}_0(P)\odot\gamma\oplus\mathbb{E}_0(P\oplus\alpha)\odot\gamma=0,\\
&\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta)+\\
&\sum_{\beta\in\{0,1\}^n}\Pr(\mathbb{E}(P)\odot\delta=\mathbb{E}(P\oplus\alpha)\odot\delta|\mathbb{E}_0(P)\odot\gamma\oplus\mathbb{E}_0(P\oplus\alpha)\odot\gamma=1,\\
&\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta)\times\Pr(\mathbb{E}_0(P)\odot\gamma\oplus\mathbb{E}_0(P\oplus\alpha)\odot\gamma=1,\\
&\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta) \qquad (1)\\
=&\left(\frac{1}{2}+2\epsilon^2\right)\times\sum_{\beta\in\{0,1\}^n,\gamma\odot\beta=0}\Pr(\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta)+\\
&\left(\frac{1}{2}-2\epsilon^2\right)\times\sum_{\beta\in\{0,1\}^n,\gamma\odot\beta=1}\Pr(\mathbb{E}_0(P)\oplus\mathbb{E}_0(P\oplus\alpha)=\beta)\\
=&\frac{1}{2}+2(2\widehat{p}-1)\epsilon^2.
\end{aligned}
$$

□

Consequently, the bias of the differential-linear distinguisher $\Delta\alpha\rightarrow\Gamma\delta$ is

$$\left|\Pr_{P\in\{0,1\}^n}(\mathbb{E}(P)\odot\delta=\mathbb{E}(P\oplus\alpha)\odot\delta)-\frac{1}{2}\right|=2\,|2\widehat{p}-1|\,\epsilon^2.$$

□

3.4 Implications

Langford and Biham et al.'s methodology requires Assumptions 1, 2 and 3, while our methodology requires only Assumptions 1 and 2. Thus, our methodology is more reasonable than Langford and Biham et al.'s methodology.

Langford and Biham et al.'s methodology holds only when Assumption 3 holds, and under the situation we have $\widehat{p}=p+(1-p)\frac{1}{2}=\frac{1}{2}+\frac{p}{2}$, meaning that the probability value obtained using Langford and Biham et al.'s methodology equals that obtained using our methodology. Thus, when Langford and Biham et al.'s methodology holds, our methodology always holds. However, our methodology holds under some situations where Langford and Biham et al.'s methodology does not hold, for example, it works for the naive situation discussed in Sect. 3.3 where $\widehat{p}=p=\frac{1}{2}$. Therefore, our methodology is more general than Langford and Biham et al.'s methodology. (When Langford and Hellman's methodology holds, our methodology always holds as well).

Our methodology still requires Assumptions 1 and 2. In practice, Assumption 1 is extensively used in differential and linear cryptanalysis and is commonly regarded as necessary. Assumption 2 seems irremovable to get such a simple and practical probability formula; otherwise, the formula could not be so simple, but a more accurate version can be easily obtained from our above reasonings, for instance, from Eq. 1, though it is complicated and appears to be hardly applicable in practice. The assumptions mean that, in some cases, the probability of a differential-linear distinguisher may be overestimated or underestimated, and so is the success probability of the attack; however, computer experiments [12,32,37,41,42] have shown that the assumptions work well in practice for some block ciphers. Anyway, it seems reasonable to take the worst case assumption from the point of the user of a cipher.

We suggest that if possible an attacker should check the validity of these assumptions when applying them to a specific cipher.

Our result shows that using only one (truncated) differential satisfying $\beta \odot \gamma = 0$ is insufficient in most situations, and it is likely to be not sufficient in the general situation; we should use all the differentials satisfying $\beta \odot \gamma = 0$ instead. This makes the distinguisher harder and even impossible to construct in practice, due to a large number of possible output differences. Anyway, we should use at least those differentials with a significant contribution to reduce the deviation if we are able to do so, (which means there is a mathematical approximation in this scenario).

Langford and Biham et al.'s methodology suggests that if the bias of the linear approximation keeps constant, the larger p is, the bigger is the bias of the distinguisher. Now, we know that may be not true in the general situation: A differential with a bigger probability will not necessarily result in a distinguisher with a bigger bias.

When constructing a differential-linear distinguisher, in Langford and Biham et al.'s methodology the attacker first chooses a (truncated) differential that meets the condition (as followed in [10,12,22,23], in practice the output difference of the differential usually has zeros in the bit positions concerned by the input mask of the linear approximation, so that $\beta \odot \gamma = 0$; a rare case is from Biham et al.'s attack [10] on the full COCONUT98 [47] block cipher, where $\beta \odot \gamma$ is unknown but constant), then calculates the probability of the differential, and finally takes this probability as the value of p. Our new methodology suggests a different format, that is, computing $\widehat{p}$. Once the linear approximation and the input difference of the differentials are chosen, that how many rounds can be constructed for a distinguisher depends to some extent on the computational power available for the attacker. Thus, our methodology is more computationally intensive in some situations.

Our new methodology can lead to some better differential-linear cryptanalytic results than Langford and Biham et al.'s methodology as well as Langford and Hellman's methodology, as to be demonstrated by its applications to the block ciphers DES, CTC2 and Serpent in the following sections. Before further proceeding, observe that DES would be a Markov cipher under the XOR difference notion [35], and similarly we can learn that both CTC2 and Serpent would be Markov ciphers under the XOR difference notion, if the subkeys of each cipher were independent.

It is worthy to note that Langford and Biham et al.'s methodology may be applicable in some situations where it is impossible to compute the output difference β, so it is infeasible to apply our new methodology; and such an example is Biham et al.'s attack [10] on COCONUT98, where $\beta \odot \gamma$ is unknown but constant.

At last, to be conservative, we would like to suggest that one should pay attention to all these methodologies, for a real situation is usually hard to predict, and it may make Assumption 3 for Langford and Biham et al.'s methodology hold.

4 Application to the DES block cipher

The DES block cipher is well known to both academia and industry, which has a 64-bit block size, a 56-bit user key, and a total of 16 rounds. We refer the reader to [44] for the specifications of DES.

In 1994, under the two default Assumptions 1 and 2 Langford and Hellman [37] used their methodology to obtain a 6-round differential-linear distinguisher of DES, and finally applied it to break 8-round DES; the attack recovers 16 key bits with a time complexity of $2^{14.6}$ 8-round DES encryptions, so it would take 2^{40} encryptions to recover the remaining

40 key bits with an exhaustive search, meaning that a total of approximately 2^{40} 8-round DES encryptions are required to recover the whole 56 key bits (Note that there might exist a more efficient way to obtain the remaining key bits). In 1995, Langford [36] applied it to greater numbers of DES, in particular, 12-round DES was breakable with a data complexity of 2^{30} chosen plaintexts. In 2002, under Assumptions 1, 2 and 3, Biham, Dunkelman and Keller [10] described a 7-round differential-linear distinguisher of DES using their enhanced methodology, and finally gave differential-linear attacks on 8- and 9-round DES; (besides, they mentioned that Langford and Biham et al.'s methodology could attack DES with up to 10 rounds). An improved version of the 9-round attack appeared in pages 108–111 of [21]. Their attack recovers 18 key bits with a time complexity of $2^{29.17}$ 9-round DES encryptions, the remaining 38 key bits would take 2^{38} encryptions to recover with a key exhaustion, and thus it has a total of approximately 2^{38} 9-round DES encryptions to recover the whole 56 key bits.

In this section, we show that our new methodology enables us to construct 7- and 8-round differential-linear distinguishers of DES based on the 3-round linear approximation used in previous differential-linear cryptanalysis of DES [10,37]; and the 8-round distinguisher allows us to break 10-round DES. More importantly, we are able to construct an 11-round differential-linear distinguisher of DES, and finally use it as the basis of a differential-linear attack on 13-round DES. Below we describe these new differential-linear cryptanalytic results for DES. We write the subkey used in the S_l S-box of Round m as $K_{m,l}$, where $1 \leq m \leq 16$, $1 \leq l \leq 8$.

4.1 A 7-round differential-linear distinguisher with bias $2^{-7.94}$

We choose the input difference for the 7-round differential-linear distinguisher to be $\Delta\alpha = 0x4000000000000000$, so that there is a one probability in the first round, thus making things easier by reducing the number of possible output differences after 4-round DES. As mentioned earlier, we use the same 3-round linear approximation $\Gamma\gamma \rightarrow \Gamma\delta$ as used in Langford and Hellman's and Biham et al.'s differential-linear cryptanalytic results of DES [10,37]: $0x21040080000080\ 00 \rightarrow 0x0000800021040080$ with bias 0.195, (an illustration of the linear approximation without the last swap was given in Fig. 2 of [10]). Let's compute the probability of the new 7-round differential-linear distinguisher. The 7-round differential-linear distinguisher as well as the following computation is partially depicted in Fig. 1.

After the expansion **E** operation of the second round, $0x4$ in the input difference becomes $0x8$, which enters the S_1 S-box of the second round and generates 11 differences after the S-box: $\{\omega|\omega = 0x3, 0x5, 0x6, 0x7, 0x9, 0xA, 0xB, 0xC, 0xD, 0xE, 0xF\}$; the probabilities for these output differences are given in the second column of Table 2. We represent ω as a concatenation of four one-bit variables $a||b||c||d$, where $a, b, c, d \in \{0, 1\}$. Thus, the right half of the third round has the input difference $00000000a0000000b00000c0000000d0$ in binary notation, and this input difference can make at most 6 S-boxes of the third round active: $S_2, S_3, S_4, S_5, S_6, S_8$. After a simple analysis, we know that the left half of the input mask $\Gamma\gamma$ concerns the four bits of the output difference of the S_5 S-box of the third round, and we denote the four-bit output difference as $e||f||g||h$, where $e, f, g, h \in \{0, 1\}$. The right half of the input mask concerns the second most significant bit of the output difference of the S_1 S-box of the fourth round plus b. The input difference of the S_1 S-box of the fourth round depends on:

- The second least significant bit of the output difference of the S_2 S-box of the third round, and we label the bit m;

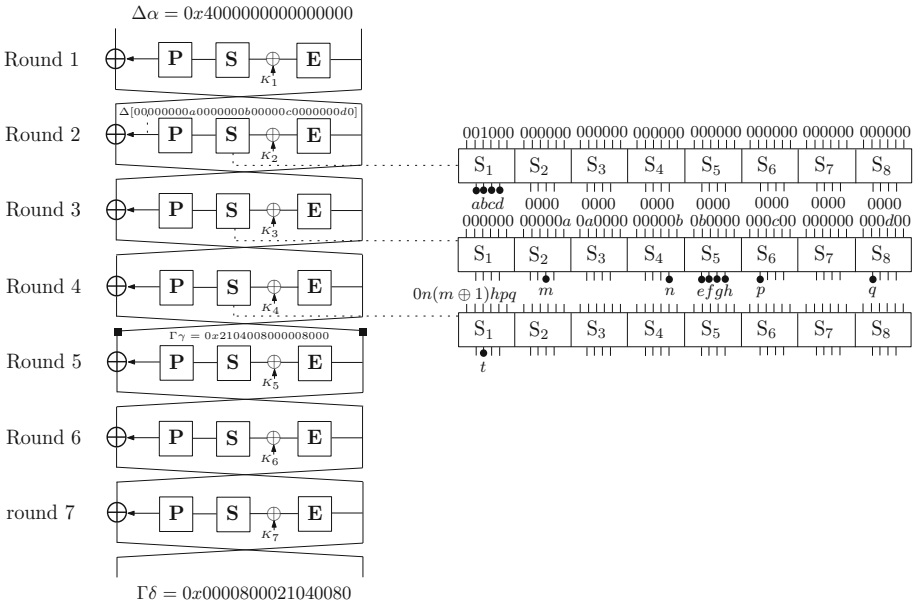

Fig. 1 A 7-round differential-linear distinguisher of DES

Table 2 Probabilities for the eleven output differences in $\{\omega\}$, where $\mathcal{A}$ denotes the event $\Delta 0x8 \rightarrow \Delta\omega$ (under the S_1 S-box)

ω	$\Pr_{S_1}(\mathcal{A})$	$\Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0 \mid \mathcal{A})$	$\Pr(\Delta\widehat{\beta}_\omega \odot \Gamma\gamma = 0 \mid \mathcal{A})$	$\Pr(\Delta\widetilde{\beta}_\omega \odot \Gamma\widetilde{\gamma} = 0 \mid \mathcal{A})$
$0x3$	$\frac{12}{64}$	0.55859375	0.50328584527596831	0.49779944866895676
$0x5$	$\frac{8}{64}$	0.50439453125	0.49747781828045845	0.49595199525356293
$0x6$	$\frac{8}{64}$	0.51708984375	0.50507303327322006	0.50433863041689619
$0x7$	$\frac{4}{64}$	0.50457763671875	0.49877615783771034	0.50256029706542904
$0x9$	$\frac{6}{64}$	0.578125	0.50051539158448577	0.50855094581311278
$0xA$	$\frac{2}{64}$	0.537109375	0.50116461620200425	0.50591027818154544
$0xB$	$\frac{8}{64}$	0.56123046875	0.49983475663202626	0.50239421910760029
$0xC$	$\frac{8}{64}$	0.4735107421875	0.49967876038863324	0.49929085310759547
$0xD$	$\frac{2}{64}$	0.4891510009765625	0.49995220528766993	0.49968796220765910
$0xE$	$\frac{2}{64}$	0.50665283203125	0.50015277066222552	0.50061782109781916
$0xF$	$\frac{4}{64}$	0.50272369384765625	0.50010005129477086	0.50005227406592345

- The least significant bit of the output difference of the S_4 S-box of the third round, and we label the bit n;
- The least significant bit (i.e., h) of the output difference of the S_5 S-box of the third round;
- The most significant bit of the output difference of the S_6 S-box of the third round, and we label the bit p;
- The most significant bit of the output difference of the S_8 S-box of the third round, and we label the bit q; and
- The one-bit difference in $\Delta\alpha$.

Table 3 Conditional probabilities $\Pr_{S_1}(\Delta t = 0|\Delta(0||n||(m \oplus 1)||h||p||q) = \xi)$

ξ	$0x0$	$0x1$	$0x2$	$0x3$	$0x4$	$0x5$	$0x6$	$0x7$	$0x8$	$0x9$	$0xA$	$0xB$	$0xC$	$0xD$	$0xE$	$0xF$
Pr	1	$\frac{32}{64}$	$\frac{28}{64}$	$\frac{36}{64}$	$\frac{20}{64}$	$\frac{28}{64}$	$\frac{28}{64}$	$\frac{36}{64}$	$\frac{28}{64}$	$\frac{28}{64}$	$\frac{32}{64}$	$\frac{32}{64}$	$\frac{24}{64}$	$\frac{40}{64}$	$\frac{36}{64}$	$\frac{24}{64}$
ξ	$0x10$	$0x11$	$0x12$	$0x13$	$0x14$	$0x15$	$0x16$	$0x17$	$0x18$	$0x19$	$0x1A$	$0x1B$	$0x1C$	$0x1D$	$0x1E$	$0x1F$
Pr	$\frac{24}{64}$	$\frac{36}{64}$	$\frac{32}{64}$	$\frac{32}{64}$	$\frac{32}{64}$	$\frac{32}{64}$	$\frac{40}{64}$	$\frac{32}{64}$	$\frac{32}{64}$	$\frac{32}{64}$	$\frac{20}{64}$	$\frac{28}{64}$	$\frac{36}{64}$	$\frac{28}{64}$	$\frac{36}{64}$	$\frac{36}{64}$

In summary, the five bits of the output difference concerned by the input mask $\Gamma\gamma$ depend on a total of 12 indeterminate one-bit differences: $a, b, c, d, e, f, g, h, m, n, p, q$. And the input difference of the S_1 S-box of the fourth round is $0||n||(m \oplus 1)||h||p||q$ in binary notation.

In the third round, the S_2 S-box has an input difference $00000a$ in binary notation, the S_4 S-box has an input difference $00000b$ in binary notation, the S_5 S-box has an input difference $0b0000$ in binary notation, the S_6 S-box has an input difference $000c00$ in binary notation, and the S_8 S-box has an input difference $000d00$ in binary notation. By the difference distribution tables of the S-boxes (see [6]), we compute the possible values as well as their probabilities for $m, n, (e||f||g||h), p, q$, as follows:

$$
\begin{array}{ll}
\Pr_{S_2}(m = 0|a = 0) = 1, & \Pr_{S_2}(m = 0|a = 1) = \frac{28}{64}, \\
\Pr_{S_2}(m = 1|a = 1) = \frac{36}{64}, & \Pr_{S_4}(n = 0|b = 0) = 1, \\
\Pr_{S_4}(n = 0|b = 1) = \frac{32}{64}, & \Pr_{S_4}(n = 1|b = 1) = \frac{32}{64}, \\
\Pr_{S_6}(p = 0|c = 0) = 1, & \Pr_{S_6}(p = 0|c = 1) = \frac{16}{64}, \\
\Pr_{S_6}(p = 1|c = 1) = \frac{48}{64}, & \Pr_{S_8}(q = 0|d = 0) = 1, \\
\Pr_{S_8}(q = 0|d = 1) = \frac{24}{64}, & \Pr_{S_8}(q = 1|d = 1) = \frac{40}{64}, \\
\Pr_{S_5}((e||f||g||h) = 0x0|b = 0) = 1, & \Pr_{S_5}((e||f||g||h) = 0x5|b = 1) = \frac{4}{64}, \\
\Pr_{S_5}((e||f||g||h) = 0x6|b = 1) = \frac{4}{64}, & \Pr_{S_5}((e||f||g||h) = 0x7|b = 1) = \frac{12}{64}, \\
\Pr_{S_5}((e||f||g||h) = 0x9|b = 1) = \frac{2}{64}, & \Pr_{S_5}((e||f||g||h) = 0xA|b = 1) = \frac{8}{64}, \\
\Pr_{S_5}((e||f||g||h) = 0xB|b = 1) = \frac{10}{64}, & \Pr_{S_5}((e||f||g||h) = 0xC|b = 1) = \frac{4}{64}, \\
\Pr_{S_5}((e||f||g||h) = 0xD|b = 1) = \frac{6}{64}, & \Pr_{S_5}((e||f||g||h) = 0xE|b = 1) = \frac{12}{64}, \\
\Pr_{S_5}((e||f||g||h) = 0xF|b = 1) = \frac{2}{64}. &
\end{array}
$$

We denote by t the second most significant bit of the output difference of the S_1 S-box of the fourth round, and by the difference distribution table of the S_1 S-box we compute the probability of $t = 0$ and 1 for all the input differences of the S-box; and the conditional probabilities $\Pr_{S_1}(\Delta t|\Delta(0||n||(m \oplus 1)||h||p||q))$ are given in Table 3. Note that $\Pr_{S_1}(\Delta t = 1|\Delta(0||n||(m \oplus 1)||h||p||q) = \xi) = 1 - \Pr_{S_1}(\Delta t = 0|\Delta(0||n||(m \oplus 1)||h||p||q) = \xi)$.

For each difference ω, we denote by β_ω the output difference(s) of the 4-round DES, and now we can compute the probability that the XOR of the concerned five bits of β_ω (whose values are $e, f, g, h, b \oplus t$) is zero (i.e., $\Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0|\Delta 0x8 \to \Delta\omega)$) by performing a computer program on all the possible (truncated) differential characteristics. These probabilities are given in the third column of Table 2. The largest number of possible differential characteristics happens when $\omega = 0xF$, which is $10 \times 2 \times 2 \times 2 \times 2 \times 2 \approx 2^{11.3}$, and a straightforward implementation takes a few seconds on a (general) personal computer.

Finally, we have $\widehat{p} = \sum_\omega \Pr_{S_1}(\Delta 0x8 \to \Delta\omega) \times \Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0|\Delta 0x8 \to \Delta\omega) = 0.526823616027832015625$. As a result, by Theorem 2 we know that the probability of the 7-round distinguisher $\Delta\alpha \to \Gamma\delta$ is $\frac{1}{2} + 2 \times 0.195^2 \times (2 \times 0.526823616027832015625 - 1) \approx \frac{1}{2} + 2 \times 0.195^2 \times 2^{-4.22} \approx \frac{1}{2} + 2^{-7.94}$. Therefore, the 7-round distinguisher $\Delta\alpha \to \Gamma\delta$ has a bias of $2^{-7.94}$.

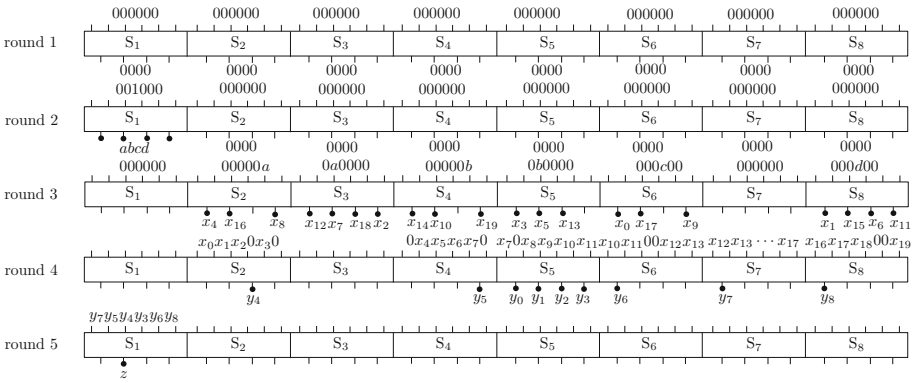

Fig. 2 Concerned differences in the S-boxes of the 5-round differentials for the 8-round differential-linear distinguisher of DES

4.2 An 8-round differential-linear distinguisher with bias $2^{-12.83}$

Further, appending one more round at the end of the four rounds covered by the differentials $\{\Delta\alpha \rightarrow \beta_\omega\}$ in the above 7-round distinguisher can lead to an 8-round differential-linear distinguisher of DES. We depict the concerned differences in the S-boxes of the 5-round differentials in Fig. 2 when computing the probability of the 8-round differential-linear distinguisher.

A detailed analysis reveals that the left half of the input mask $\Gamma\gamma$ concerns the four bits of the output difference of the S_5 S-box of the fourth round, and the right half of the input mask concerns the second most significant bit of the output difference of the S_1 S-box of the fifth round. Let $y_0, y_1, y_2, y_3, z \in \{0, 1\}$ be one-bit variables; we denote by $y_0||y_1||y_2||y_3$ the output difference of the S_5 S-box of the fourth round, and denote by z the second most significant bit of the output difference of the S_1 S-box of the fifth round. The input difference of the S_1 S-box of the fifth round depends on:

- The second least significant bit of the output difference of the S_2 S-box of the fourth round, and we label the bit y_4;
- The least significant bit of the output difference of the S_4 S-box of the fourth round, and we label the bit y_5;
- The least significant bit (i.e., y_3) of the output difference of the S_5 S-box of the fourth round;
- The most significant bit of the output difference of the S_6 S-box of the fourth round, and we label the bit y_6;
- The most significant bit of the output difference of the S_7 S-box of the fourth round, and we label the bit y_7; and
- The most significant bit of the output difference of the S_8 S-box of the fourth round, and we label the bit y_8.

The input difference of the S_2 S-box of the fourth round is $x_0||x_1||x_2||0||x_3||0$ (in binary notation), where x_0 denotes the most significant bit of the output difference of the S_6 S-box of the third round, x_1 denotes the most significant bit of the output difference of the S_8 S-box of the third round, x_2 denotes the least significant bit of the output difference of the S_3 S-box of the third round, and x_3 denotes the most significant bit of the output difference of the S_5 S-box of the third round. The input difference of the S_4 S-box of the

fourth round is $0||x_4||x_5||x_6||x_7||0$, where x_4 denotes the most significant bit of the output difference of the S_2 S-box of the third round, x_5 denotes the second most significant bit of the output difference of the S_5 S-box of the third round, x_6 denotes the second least significant bit of the output difference of the S_8 S-box of the third round, and x_7 denotes the second most significant bit of the output difference of the S_3 S-box of the third round. The input difference of the S_5 S-box of the fourth round is $x_7||0||x_8||x_9||x_{10}||x_{11}$, where x_8 denotes the least significant bit of the output difference of the S_2 S-box of the third round, x_9 denotes the least significant bit of the output difference of the S_6 S-box of the third round, x_{10} denotes the second most significant bit of the output difference of the S_4 S-box of the third round, and x_{11} denotes the least significant bit of the output difference of the S_8 S-box of the third round. The input difference of the S_6 S-box of the fourth round is $x_{10}||x_{11}||0||0||x_{12}||x_{13}$, where x_{12} denotes the most significant bit of the output difference of the S_3 S-box of the third round, and x_{13} denotes the second least significant bit of the output difference of the S_5 S-box of the third round. The input difference of the S_7 S-box of the fourth round is $x_{12}||x_{13}||x_{14}||x_{15}||x_{16}||x_{17}$, where x_{14} denotes the most significant bit of the output difference of the S_4 S-box of the third round, x_{15} denotes the second most significant bit of the output difference of the S_8 S-box of the third round, x_{16} denotes the second most significant bit of the output difference of the S_2 S-box of the third round, and x_{17} denotes the second most significant bit of the output difference of the S_6 S-box of the third round. The input difference of the S_8 S-box of the fourth round is $x_{16}||x_{17}||x_{18}||0||0||x_{19}$, where x_{18} denotes the second least significant bit of the output difference of the S_3 S-box of the third round, and x_{19} denotes the least significant bit of the output difference of the S_4 S-box of the third round.

The differential characteristics for the first two rounds are the same as in the 7-round distinguisher. In summary, the five bits of the output difference concerned by the input mask $\Gamma\gamma$ depend on a total of 33 indeterminate one-bit differences: $a, b, c, d, x_0, x_1, \cdots, x_{19}, y_0, y_1, \cdots, y_8$. For each difference ω, we denote by $\widehat{\beta}_\omega$ the output difference(s) of the 5-round DES. Now, similar to that described for the 7-round distinguisher we can compute the probability that the XOR of the concerned five bits of $\widehat{\beta}_\omega$ (whose values are y_0, y_1, y_2, y_3, z) is zero (i.e., $\Pr(\Delta\widehat{\beta}_\omega \odot \Gamma\gamma = 0|\Delta 0x8 \rightarrow \Delta\omega)$) by performing a computer program on all the possible (truncated) differential characteristics. These probabilities are given in the fourth column of Table 2. The largest number of possible differential characteristics happens also when $\omega = 0xF$, which is roughly $7 \times 10 \times 4 \times 6 \times 6 \times 10 \times 2^4 \times 2 \times 2 \times 2 \times 2 \times 2 \approx 2^{25.6}$; and it takes a few seconds to check on a personal computer.

As a result, we have $\widehat{p} = \sum_\omega \Pr_{S_1}(\Delta 0x8 \rightarrow \Delta\omega) \times \Pr(\Delta\widehat{\beta}_\omega \odot \Gamma\gamma = 0|\Delta 0x8 \rightarrow \Delta\omega) = 0.500891897847608796875$, and thus by Theorem 2 the probability of the 8-round distinguisher $\Delta\alpha \rightarrow \Gamma\delta$ is $\frac{1}{2} + 2 \times 0.195^2 \times (2 \times 0.500891897847608796875 - 1) \approx \frac{1}{2} + 2 \times 0.195^2 \times 2^{-9.11} \approx \frac{1}{2} + 2^{-12.83}$. Therefore, the 8-round distinguisher $\Delta\alpha \rightarrow \Gamma\delta$ has a bias of $2^{-12.83}$.

We have checked the possibility of extending the 8-round distinguisher $\Delta\alpha \rightarrow \Gamma\delta$ to a 9-round distinguisher by appending one more round at the end of the five rounds covered by the differentials $\{\Delta\alpha \rightarrow \widehat{\beta}_\omega\}$. Now the five bits of the output difference concerned by the input mask $\Gamma\gamma$ depend on a total of 65 indeterminate one-bit differences, and there are roughly $2^{55.6}$ possible differential characteristics for $\omega = 0xF$. This is computationally infeasible for a general personal computer.

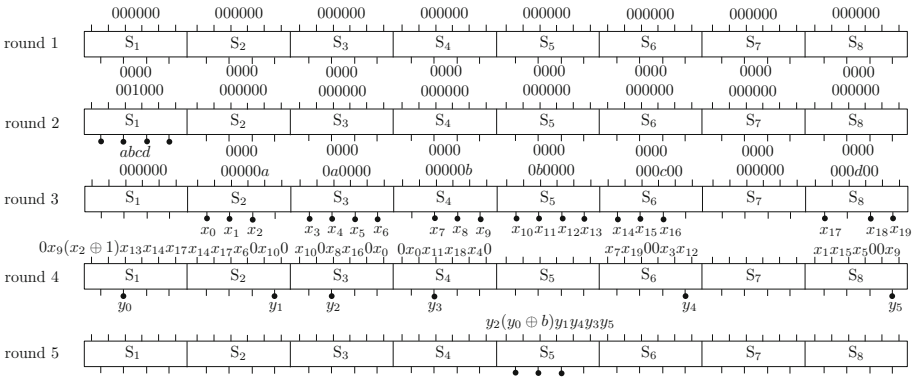

Fig. 3 Concerned differences in the S-boxes of the 5-round differentials for the 11-round differential-linear distinguisher of DES

4.3 An 11-round differential-linear distinguisher with bias $2^{-24.05}$

More importantly, the new methodology enables us to construct an 11-round differential-linear distinguisher of DES. The 11-round differential-linear distinguisher is made up of a 6-round linear approximation $\Gamma\widetilde{\gamma} \rightarrow \Gamma\widetilde{\delta}$ with bias $1.95 \times 2^{-9} \approx 2^{-8.04}$ and all the 5-round differentials $\{\Delta\alpha \rightarrow \Delta\widetilde{\beta}\}$ with $\Delta\alpha = 0x4000000000000000$. The 6-round linear approximation $\Gamma\widetilde{\gamma} \rightarrow \Gamma\widetilde{\delta}$ is $0x0000000001040080 \rightarrow 0x0000800021040080$, (which is the best 6-round linear approximation given in [41], where the last swap is not included). Below we compute the probability of the 11-round differential-linear distinguisher using the new methodology, where the concerned differences in the S-boxes of the 5-round differentials are depicted in Fig. 3.

We first consider the 5-round differentials $\{\Delta\alpha \rightarrow \Delta\widetilde{\beta}\}$. There is a one probability in the first round, meaning that the first round is bypassed by the differential characteristic with probability 1. After the **E** expansion operation of the second round, $0x4$ in $\Delta\alpha$ becomes $0x8$, which enters the S_1 S-box of the second round and generates 11 differences after the S-box: $\{\omega|\omega = 0x3, 0x5, 0x6, 0x7, 0x9, 0xA, 0xB, 0xC, 0xD, 0xE, 0xF\}$; the probabilities for these output differences are given in the second column of Table 2. We represent ω as a concatenation of four one-bit variables $a||b||c||d$, where $a, b, c, d \in \{0, 1\}$. Thus, the right half of the third round has the input difference $00000000a\ 0000000b00000c0000000d0$ in binary notation, and this input difference can make at most 6 S-boxes of the third round active: $S_2, S_3, S_4, S_5, S_6, S_8$.

In the third round, the S_2 S-box has an input difference $00000a$ in binary notation, the S_3 S-box has an input difference $0a0000$ in binary notation, the S_4 S-box has an input difference $00000b$ in binary notation, the S_5 S-box has an input difference $0b0000$ in binary notation, the S_6 S-box has an input difference $000c00$ in binary notation, and the S_8 S-box has an input difference $000d00$ in binary notation. We denote respectively by x_0, x_1, x_2 the most significant bit, the second most significant bit and the second least significant bit of the output difference of the S_2 S-box, by $x_3||x_4||x_5||x_6$ the output difference of the S_3 S-box, by x_7, x_8, x_9 the second most significant bit, the second least significant bit and the least significant bit of the output difference of the S_4 S-box, by $x_{10}||x_{11}||x_{12}||x_{13}$ the output difference of the S_5 S-box, by x_{14}, x_{15}, x_{16} the most significant bit, the second most significant bit and the second least significant bit of the output difference of the S_6 S-box,

and by x_{17}, x_{18}, x_{19} the most significant bit, the second least significant bit and the least significant bit of the output difference of the S_8 S-box.

In the fourth round, the S_1 S-box has the input difference $0||x_9||(x_2 \oplus 1)||x_{13}||x_{14}||x_{17}$, and we denote by y_0 the second most significant bit of its output difference; the S_2 S-box has the input difference $x_{14}||x_{17}||x_6||0||x_{10}||0$, and we denote by y_1 the least significant bit of its output difference; the S_3 S-box has the input difference $x_{10}||0||x_8||x_{16}||0||x_0$, and we denote by y_2 the second most significant bit of its output difference; the S_4 S-box has the input difference $0||x_0||x_{11}||x_{18}||x_4||0$, and we denote by y_3 the second most significant bit of its output difference; the S_6 S-box has the input difference $x_7||x_{19}||0||0||x_3||x_{12}$, and we denote by y_4 the least significant bit of its output difference; the S_8 S-box has the input difference $x_1||x_{15}||x_5||0||0||x_9$, and we denote by y_5 the least significant bit of its output difference. Thus, we have that the input difference of the S_5 S-box of the fifth round is $y_2||(y_0 \oplus b)||y_1||y_4||y_3||y_5$.

A simple analysis reveals that the three bits concerned by the input mask $\Gamma\gamma$ depend on:

- x_{10}, x_{11} and x_{12}; and
- The three most significant bits of the output difference of the S_5 S-box of the fifth round; and we denote the XOR of the three bits by z.

For each difference ω, we denote by $\widetilde{\beta}_\omega$ the output difference(s) of the 5-round DES. Now, by the difference distribution tables of the S-boxes (see [6]) we can compute the probability that the XOR of the concerned three bits of $\widetilde{\beta}_\omega$ (i.e., $x_{10} \oplus x_{11} \oplus x_{12} \oplus z$) is zero by performing a computer program on all the possible (truncated) differential characteristics. These probabilities are given in the fifth column of Table 2. The largest number of possible differential characteristics happens also when $\omega = 0xF$, which is $7 \times 10 \times 4 \times 10 \times 6 \times 7 \times 2^6 \times 2 \approx 2^{23.9}$; and it takes a few seconds to check on a personal computer.

Finally, we have $\widehat{p} = \sum_\omega \Pr_{S_1}(\Delta 0x8 \to \Delta\omega) \times \Pr(\Delta\widetilde{\beta}_\omega \odot \Gamma\gamma = 0 | \Delta 0x8 \to \Delta\omega) = 0.500993547648294625$. As a result, by Theorem 2 we have that the probability of the 11-round distinguisher $\Delta\alpha \to \Gamma\widetilde{\delta}$ is $\frac{1}{2} + 2 \times (2^{-8.04})^2 \times (2 \times 0.500993547648294625 - 1) \approx \frac{1}{2} + 2 \times (2^{-8.04})^2 \times 2^{-8.97} = \frac{1}{2} + 2^{-24.05}$. Therefore, the 11-round distinguisher $\Delta\alpha \to \Gamma\widetilde{\delta}$ has a bias of $2^{-24.05}$.

4.4 Differential-linear attack on 10-round DES

The 8-round distinguisher $\Delta\alpha \to \Gamma\delta$ enables us to construct a differential-linear attack breaking 10 rounds of DES. We assume that the attacked rounds are the first ten rounds from Rounds 1 to 10. A simple analysis on the key schedule of DES reveals that $K_{1,1}$ and $K_{10,1}$ overlap in 2 bits (i.e. bits 17 and 51 of the user key), and thus given $K_{1,1}$ we know 2 bits of $K_{10,1}$. The attack procedure is as follows.

1. Choose $2^{24.46}$ structures $\mathcal{S}_i$, $(i = 1, 2, \ldots, 2^{24.46})$, where a structure is defined to be a set of 2^4 plaintexts $P_{i,j}$ with bits (9, 17, 23, 31) of the left half taking all the possible values, bit (2) of the right half fixed to 0 and the other 59 bits fixed, $(j = 1, 2, \ldots, 2^4)$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^4 plaintexts in each of the $2^{24.46}$ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$.
2. Choose $2^{24.46}$ structures $\widehat{\mathcal{S}}_i$, $(i = 1, \ldots, 2^{24.46})$, where a structure $\widehat{\mathcal{S}}_i$ is obtained by setting 1 to bit (2) of the right half of all the plaintexts $P_{i,j}$ in $\mathcal{S}_i$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^4 plaintexts in each $\widehat{\mathcal{S}}_i$.
3. Initialize 2^{10} counters to zero, which correspond to all the possible values of the subkey $(K_{1,1}, K_{10,1})$.
4. Guess a value for $K_{1,1}$, and do as follows.

(a) Partially encrypt every plaintext $P_{i,j}$ with the guessed $K_{1,1}$ to get its intermediate value immediately after Round 1 ; we denote it by $\varepsilon_{i,j}$.
(b) Partially decrypt $\varepsilon_{i,j} \oplus 0x4000000000000000$ with the guessed $K_{1,1}$ to get its plaintext, and find the plaintext in $\widehat{\mathcal{S}}_i$; we denote it by $\widehat{P}_{i,j}$, and denote by $\widehat{C}_{i,j}$ the corresponding ciphertext for $\widehat{P}_{i,j}$. Store $(C_{i,j}, \widehat{C}_{i,j})$ in a table.

5. Guess a value for the remaining 4 bits of $K_{10,1}$, and do as follows for every ciphertext pair $(C_{i,j}, \widehat{C}_{i,j})$.
   (a) Partially decrypt $C_{i,j}$ and $\widehat{C}_{i,j}$ with $K_{10,1}$ to get bit (17) of the left half of their intermediate values immediately before Round 10.
   (b) Check whether the XOR of the five bits for $C_{i,j}$ — bit (17) of the left half and bits (3,8,14,25) of the right half of its intermediate value immediately before Round 10 — is equal to the XOR of the corresponding five bits for $\widehat{C}_{i,j}$. If yes, add one to the counter corresponding to the guessed $(K_{1,1}, K_{10,1})$.
6. Output the guess for $(K_{1,1}, K_{10,1})$ with the highest deviation from $2^{27.46}$.

The attack requires $2^{29.46}$ chosen plaintexts. We start to collect another pair of structures of plaintexts only after testing a pair of structures of plaintexts, so that we can reuse the memory for storing the pair of structures of plaintexts, hence the required memory of the attack is dominated by the storage of the 2^{10} counters, which is $2^{10} \times \frac{28.46}{8} \approx 2^{11.9}$ bytes of memory. (If we would like to collect all the plaintexts at once, then it would require a storage of $2^{29.46} \times 8 \times 2 = 2^{33.46}$ bytes.) Step 4 has a time complexity of approximately $2 \times 2^{28.46} \times 2^6 \times \frac{1}{8\times 10} \approx 2^{29.14}$ 10-round DES encryptions. Step 5 has a time complexity of approximately $2 \times 2^{28.46} \times 2^{10} \times \frac{1}{8\times 10} \approx 2^{33.14}$ 10-round DES encryptions. There are $2^{28.46}$ plaintext pairs $(P_{i,j}, \widehat{P}_{i,j})$ for a guess of $(K_{1,1}, K_{10,1})$. Following Theorem 1, we learn that the attack has a success probability of about $\Phi(|2(\frac{1}{2}+2^{-12.83})-1|\sqrt{2^{28.46}}-\Phi^{-1}(1-\frac{1}{2^{10}})) \approx \Phi(5.28-\Phi^{-1}(0.99902)) \approx \Phi(5.28-3.1) \approx 99\%$.[3] Hence, the attack has a time complexity of $2^{29.46} + 2^{29.14} + 2^{33.14} \approx 2^{33.33}$ 10-round DES encryptions to recover 10 key bits.

A slightly more efficient attack, which is similar to the 13-round DES attack given in the next subsection, has a time complexity of $2^{29.46} + 2 \times 2^{28.46} \times 2^6 \times \frac{1}{8\times 10} + \frac{2^{28.46}\times 2^6}{10} + 2 \times 2^6 \times 2^4 \times 2^{20} \times \frac{1}{8\times 10} \approx 2^{31.78}$ 10-round DES encryptions to recover the 10 key bits.

The remaining 46 key bits can be found by exhaustive search, which means that it takes a total time complexity of 2^{46} 10-round DES encryptions to recover the full 56-bit user key. We note that it may be possible to recover the full key with a lower time complexity by exploiting one or more differential-linear attacks to recover the remaining 46 key bits, instead of exhaustive search on them. Anyway, the attack shows that our methodology enables us to break 10-round DES when using the same 3-round linear approximation as in those previous differential-linear cryptanalysis of DES.

4.5 Differential-linear attack on 13-round DES

The 11-round distinguisher $\Delta\alpha \rightarrow \Gamma\widetilde{\delta}$ can be used to break 13-round DES. We assume that the attacked rounds are the first thirteen rounds from Rounds 1 to 13. A simple analysis on the key schedule of DES reveals that $K_{1,1}$ and $K_{13,1}$ overlap in 2 bits (i.e. bits 17 and 34

[3] Note that here a differential-linear distinguisher is treated as a linear approximation, so that we can use Theorem 1 for a differential-linear attack, but here we can replace $m + 1$ in Theorem 1 with m, (since the term on the XOR of concerned key bits is cancelled in a differential-linear distinguisher). The same statement applies to the subsequent attacks, although we do not make any further explicit statements.

of the user key), and thus given $K_{1,1}$ we know 2 bits of $K_{13,1}$. The attack procedure is as follows.

1. Choose $2^{47.1}$ structures $\mathcal{S}_i$, $(i = 1, 2, \ldots, 2^{47.1})$, where a structure is defined to be a set of 2^4 plaintexts $P_{i,j}$ with bits (9, 17, 23, 31) of the left half taking all the possible values, bit (2) of the right half fixed to 0 and the other 59 bits fixed, $(j = 1, 2, \ldots, 2^4)$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^4 plaintexts in each of the $2^{47.1}$ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$.
2. Choose $2^{47.1}$ structures $\widehat{\mathcal{S}}_i$, $(i = 1, \ldots, 2^{47.1})$, where a structure $\widehat{\mathcal{S}}_i$ is obtained by setting 1 to bit (2) of the right half of all the plaintexts $P_{i,j}$ in $\mathcal{S}_i$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^4 plaintexts in each $\widehat{\mathcal{S}}_i$.
3. Guess a value for $K_{1,1}$, and do as follows.
   (a) Initialize 2^{20} counters to zero, which correspond to the 2^{20} possible pairs consisting of the possible values for a couple of the 10 ciphertext bits: bit (17) of the left half and bits (1, 2, 3, 4, 5, 8, 14, 25, 32) of the right half.
   (b) Partially encrypt every plaintext $P_{i,j}$ with the guessed $K_{1,1}$ to get its intermediate value immediately after Round 1; we denote it by $\varepsilon_{i,j}$.
   (c) Partially decrypt $\varepsilon_{i,j} \oplus 0x4000000000000000$ with the guessed $K_{1,1}$ to get its plaintext, and find the plaintext in $\widehat{\mathcal{S}}_i$; we denote it by $\widehat{P}_{i,j}$, and denote by $\widehat{C}_{i,j}$ the corresponding ciphertext for $\widehat{P}_{i,j}$. Store $(C_{i,j}, \widehat{C}_{i,j})$ in a table.
   (d) For every ciphertext pair $(C_{i,j}, \widehat{C}_{i,j})$, add one to the counter corresponding to the pair of the above-mentioned 10 ciphertext bits from $(C_{i,j}, \widehat{C}_{i,j})$.
   (e) Guess a value for the unknown 4 bits of $K_{13,1}$, and do as follows.
      (i) For each of the 2^{20} pairs of the concerned 10 ciphertext bits, partially decrypt it with the guessed $K_{13,1}$ to get the pair of the 5 bits concerned by the output mask $\Gamma\delta$, and compute the XOR of the pair of the 5 bits (concerned by the output mask).
      (ii) Count the number of the ciphertext pairs $(C_{i,j}, \widehat{C}_{i,j})$ such that the XOR of the pair of the 5 bits concerned by $\Gamma\delta$ is zero, and compute its deviation from $2^{50.1}$.
      (iii) If the guess for $(K_{1,1}, K_{13,1})$ is the first guess for $(K_{1,1}, K_{13,1})$, then record the guess and the deviation computed in Step 3(e)(ii); otherwise, record the guess and its deviation only when the deviation is larger than that of the previously recorded guess, and remove the guess with the smaller deviation.
4. For the $(K_{1,1}, K_{13,1})$ recorded in Step 3(e)(iii), exhaustively search for the remaining 46 key bits with two known plaintext-ciphertext pairs. If a 56-bit key is suggested, output it as the user key of the 13-round DES.

The attack requires $2^{52.1}$ chosen plaintexts. Steps 1 and 2 have a time complexity of $2^{52.1}$ 13-round DES encryptions. Steps 3(b) and 3(c) have a time complexity of $2 \times 2^{51.1} \times 2^6 \times \frac{1}{8\times 13} \approx 2^{51.4}$ 13-round DES encryptions. Step 3(d) has a time complexity of $2^{51.1} \times 2^6 = 2^{57.1}$ memory accesses. Roughly, an extremely conservative estimate from a theoretical viewpoint is: 13 memory accesses equal a 13-round DES encryption in terms of time, assuming that the 13-round DES is implemented with 8 parallel S-box lookups per round and one round is equivalent to one memory access. So the time complexity of Step 3(d) is equivalent to $\frac{2^{57.1}}{13} \approx 2^{53.4}$ 13-round DES encryptions. The time complexity of Step 3(e) is dominated by the time complexity of Step 3(e)(i), which is $2 \times 2^6 \times 2^4 \times 2^{20} \times \frac{1}{8\times 13} \approx 2^{24.3}$ 13-round DES encryptions. Step 4 has a time complexity of 2^{46} 13-round DES encryptions. Therefore, the attack has a total time complexity of approximately $2^{54.2}$ 13-round DES encryptions, faster than exhaustive key search.

If we would like to collect all the plaintexts at once, then the required memory for the attack is dominated by the storage of the plaintexts and ciphertexts, which is $2^{52.1} \times 16 = 2^{56.1}$ bytes; however, we can implement the attack procedure in an optimised way: We initialize 2^{26} counters, which correspond to the possible values of the combination of $K_{1,1}$ and the couple of the 10 ciphertext bits described in Step 3(a); and we start to collect another pair of structures of plaintexts only after testing a pair of structures of plaintexts. Thus, the required memory of the optimised attack is dominated by the storage of the 2^{26} counters, which is $2^{26} \times \frac{20}{8} \approx 2^{27.4}$ bytes of memory.

There are $2^{51.1}$ plaintext pairs $(P_{i,j}, \widehat{P}_{i,j})$ for a guess of $(K_{1,1}, K_{13,1})$, and thus following Theorem 1, we have that the attack has a success probability of about $\Phi(|2(\frac{1}{2} + 2^{-24.05}) - 1|\sqrt{2^{51.1}} - \Phi^{-1}(1 - \frac{1}{2^{10}})) \approx \Phi(5.66 - \Phi^{-1}(0.99902)) \approx \Phi(2.56) \approx 99\%$.

5 Application to the CTC2 block cipher

The CTC2 [16] block cipher was designed to show the strength of algebraic cryptanalysis [17] on block ciphers by the proposer of algebraic cryptanalysis, who described an algebraic attack on 6 rounds of the version of CTC2 that uses a 255-bit block size and a 255-bit key.

In 2009, using Langford and Biham et al.'s methodology Dunkelman and Keller [22] described 6- and 7-round differential-linear distinguishers for the version of CTC2, and finally presented differential-linear attacks on 7 and 8 rounds of CTC2 (with a 255-bit block size and key). The 8-round attack is known as the best previously published cryptanalytic result on the version of CTC2 in terms of the numbers of attacked rounds.

In this section, we show a flaw in the previous differential-linear cryptanalytic results for CTC2. Then, under the new methodology we construct an 8.5-round differential-linear distinguisher with bias $2^{-69.42}$ for the CTC2 with a 255-bit block size and key, and finally give a differential-linear attack on 10-round CTC2 with a 255-bit block size and key. Notice that in this section, sometimes we simply write $\Delta\alpha \xrightarrow{\mathbb{T}} \Delta\beta$ to denote the differential $\Delta\alpha \to \Delta\beta$ for some operation $\mathbb{T}$, and write $\Gamma\alpha \xrightarrow{\mathbb{T}} \Gamma\beta$ to denote the linear approximation $\Gamma\alpha \to \Gamma\beta$ for $\mathbb{T}$. We first briefly describe the CTC2 block cipher.

5.1 The CTC2 block cipher

The CTC2 [16] block cipher has a variable block size, a variable length key and a variable number of rounds. As in [22], we only consider the version of CTC2 that uses a 255-bit block size and a 255-bit key. CTC2 uses the following two elementary operations to construct its round function.

- **S** is a non-linear substitution operation constructed by applying the same 3×3-bit bijective S-box 85 times in parallel to an input.
- **D** is a linear diffusion operation, which takes a 255-bit block $Y = (Y_{254}, \ldots, Y_1, Y_0)$ as input, and outputs a 255-bit block $Z = (Z_{254}, \ldots, Z_1, Z_0)$, computed as defined below, where $i = 0, 1, 3, 4, \cdots, 254$.

$$\begin{cases} Z_{151} = Y_2 \oplus Y_{139} \oplus Y_{21} \\ Z_{(i\times 202+2) \bmod 255} = Y_i \oplus Y_{(i+137) \bmod 255} \end{cases}$$

CTC2 takes as input a 255-bit plaintext block P, and its encryption procedure for N_r rounds is, where $Z_0, X_i, Y_i, Z_i, X_{N_r}, Y_{N_r}, Z_{N_r}$ are 255-bit variables, and K_0, K_i, K_{N_r} are round keys generated from a user key K as $K_j = K \lll j$ in our notation, $(0 \le j \le N_r)$.

1. $Z_0 = P$.
2. For $i = 1$ to $N_r - 1$:
   - $X_i = Z_{i-1} \oplus K_{i-1}$,
   - $Y_i = \mathbf{S}(X_i)$,
   - $Z_i = \mathbf{D}(Y_i)$.
3. $X_{N_r} = Z_{N_r-1} \oplus K_{N_r-1}$, $Y_{N_r} = \mathbf{S}(X_i)$, $Z_{N_r} = \mathbf{D}(Y_{N_r})$.
4. Ciphertext $= Z_{N_r} \oplus K_{N_r}$.

To keep in accordance with [16], the ith iteration of Step 2 in the above description is referred to as Round i, $(1 \le i \le N_r - 1)$, and the transformations in Steps 3 and 4 are referred to as Round N_r. We number the 85 S-boxes in a round from 0 to 84 from right to left.

5.2 A flaw in previous differential-linear cryptanalysis of CTC2

Observe that Dunkelman and Keller used the 0.5-round differential $e_{30,151} \xrightarrow{\mathbf{D}} e_2$ with probability 1 in their differential-linear attacks presented in [22]. However, we find that this 0.5-round differential $e_{30,151} \xrightarrow{\mathbf{D}} e_2$ is not correct: For the **D** operation, given the input difference $e_{30,151}$, we cannot get the output difference e_2; and the correct output difference should be $e_{25,63,159,197}$. On the other hand, for the **D** operation, given the output difference e_2, the input difference has over fifty non-zero bits, much more than the number two in $e_{30,151}$. As a consequence, the differential-linear cryptanalytic results are flawed.

Note that Dunkelman and Keller also described differential attacks on 5-, 6- 7- and 8-round CTC2 in [22], and the 0.5-round differential $e_{30,151} \xrightarrow{\mathbf{D}} e_2$ with probability 1 was also used and played a very important role in the differential results; thus they are flawed, too.

An anonymous referee pointed out that the flaw on CTC2 might be because a wrong linear operation was used and thus there might also exist a flaw in the 3.5-round linear approximation given in [22]: $e_{14,104,134,241} \xrightarrow{\mathbf{S}} e_{14,104,132,241} \xrightarrow{\mathbf{D}} e_{38,154} \xrightarrow{\mathbf{S}} e_{36,154} \xrightarrow{\mathbf{D}} e_0 \xrightarrow{\mathbf{S}} e_2 \xrightarrow{\mathbf{D}} e_{30,151} \xrightarrow{\mathbf{S}} e_{32,151}$. We checked this 3.5-round linear approximation and found that the input mask $e_{14,104,132,241}$ for the first **D** operation was not correct, which should be $e_{26,33,163,170}$; and thus the input mask $e_{14,104,134,241}$ for the first **S** operation is not correct, either, but the remaining 2.5-round linear approximation $e_{38,154} \xrightarrow{\mathbf{S}} e_{36,154} \xrightarrow{\mathbf{D}} e_0 \xrightarrow{\mathbf{S}} e_2 \xrightarrow{\mathbf{D}} e_{30,151} \xrightarrow{\mathbf{S}} e_{32,151}$ is correct, in particular it is correct for the part $e_2 \xrightarrow{\mathbf{D}} e_{30,151}$ corresponding to the part $e_{30,151} \xrightarrow{\mathbf{D}} e_2$ concerned by the flaw in the above mentioned differential and differential-linear cryptanalysis results. We can easily obtain a correct 3.5-round linear approximation with the same bias by correcting the first round of the above 3.5-round linear approximation from [22]. In the next subsection we build a 8.5-round differential-linear distinguisher with bias $2^{-69.42}$ based on a corrected 3.5-round linear approximation.

5.3 An 8.5-round differential-linear distinguisher with bias $2^{-69.42}$

The 8.5-round differential-linear distinguisher is made up of a 5.5-round linear approximation $\Gamma\gamma \to \Gamma\delta$ with bias 2^{-33} and all the 3-round differentials $\{\Delta\alpha \to \Delta\beta\}$ with $\Delta\alpha = e_0$. The 5.5-round linear approximation $\Gamma\gamma \to \Gamma\delta$ is $e_{3,42,46,66,78,98,102,140,141,160,164,196,203,235,239} \xrightarrow{\mathbf{S}} e_{3,4,42,46,66,78,98,102,140,141,160,\ 164,196,203,235,239} \xrightarrow{\mathbf{D}} e_{45,69,74,98,163,192,206,235} \xrightarrow{\mathbf{S}} e_{45,69,74,98,163,192,206,235} \xrightarrow{\mathbf{D}} e_{26,\ 33,163,170} \xrightarrow{\mathbf{S}} e_{26,33,163,170} \xrightarrow{\mathbf{D}} e_{38,154} \xrightarrow{\mathbf{S}} e_{36,154} \xrightarrow{\mathbf{D}} e_0 \xrightarrow{\mathbf{S}}$

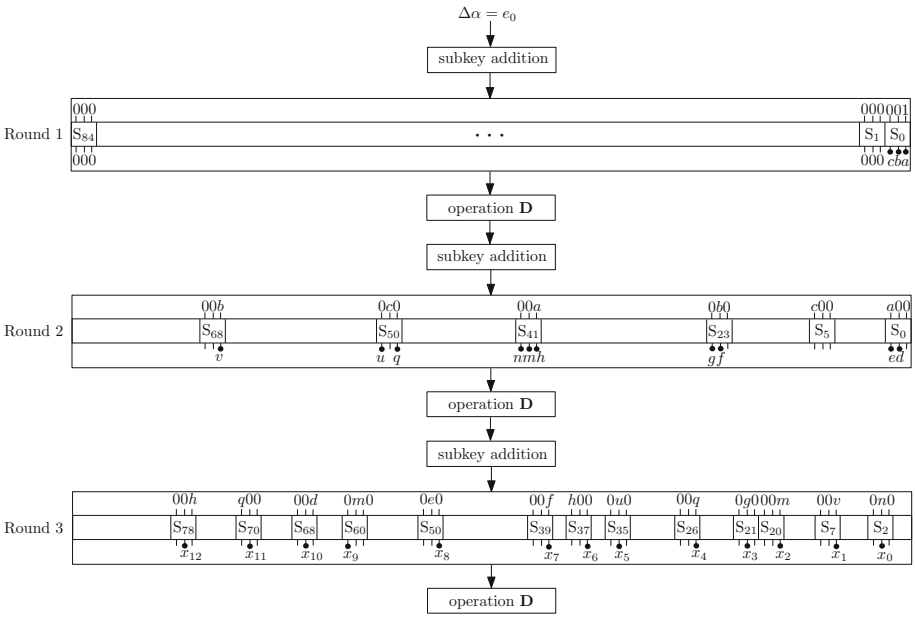

Fig. 4 Concerned differences in the S-boxes of the 3-round differentials for the 8.5-round differential-linear distinguisher of CTC2

$e_2 \xrightarrow{\mathbf{D}} e_{30,151} \xrightarrow{\mathbf{S}} e_{32,151}$. The input difference α is chosen so that there are only 16 active bit positions after being applied $\mathbf{D}^{-1}$. This enables us to conduct a differential-linear attack on 10-round CTC2 that will be presented in Sect. 5.4. For any other one-bit difference except e_0, there are more than 50 active bit positions after applying $\mathbf{D}^{-1}$ to it, and thus the resulting distinguisher cannot be used to break 10-round CTC2, because too many subkey bits are to be guessed.

We now compute the probability of the 8.5-round differential-linear distinguisher, and Fig. 4 depicts the concerned differences in the S-boxes of the 3-round differentials for the 8.5-round differential-linear distinguisher. Without loss of generality, we assume that the 3-round differentials $\{\Delta\alpha \rightarrow \Delta\beta\}$ operate on Rounds 1 to 3, and the 5.5-round linear approximation $\Gamma\gamma \rightarrow \Gamma\delta$ operates on Rounds 4 to 9 (just before the **D** operation of Round 9). By the **D** operation, we learn that the input mask $\Gamma\gamma$ concerns the following 30 bit positions of the output difference of the **S** operation of Round 3: Bits 3, 7, 13, 21, 46, 54, 60, 64, 74, 78, 84, 87, 107, 111, 117, 140, 144, 150, 158, 164, 172, 178, 182, 205, 211, 215, 221, 235, 244 and 248. The 30 concerned bit positions are covered in 30 S-boxes of Round 3: S-boxes 1, 2, 4, 7, 15, 18, 20, 21, 24, 26, 28, 29, 35, 37, 39, 46, 48, 50, 52, 54, 57, 59, 60, 68, 70, 71, 73, 78, 81 and 82; let Ω be the set of the 30 S-boxes.

On the other direction, the input difference $\Delta\alpha$ generates 4 possible differences after the **S** operation of Round 1: $\{\omega | \omega = e_0, e_1, e_2, e_{0,1,2}\}$, each with probability 2^{-2} as shown in the second row of Table 4. We represent the least significant three bits of ω as a concatenation of three one-bit variables $c||b||a$, where $a, b, c \in \{0, 1\}$. After the **D** operation of Round 1, a difference ω causes at most 6 active S-boxes of Round 2: S-boxes 0, 5, 23, 41, 50 and 68; the input difference for S-box 0 is $a||0||0$ in binary notation, the input difference for S-box 5 is $c||0||0$ in binary notation, the input difference for S-box 23 is $0||b||0$ in binary notation, the

Table 4 Probabilities for the four output differences in $\{\omega\}$

Difference (ω)	e_0	e_1	e_2	$e_{0,1,2}$
$\Pr_{\mathbf{S}}(\Delta\alpha \to \Delta\omega)$	2^{-2}	2^{-2}	2^{-2}	2^{-2}
$\Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0 \vert \Delta\alpha \to \Delta\omega)$	0.46875	0.5	0.625	0.5

input difference for S-box 41 is $0||0||a$ in binary notation, the input difference for S-box 50 is $0||c||0$ in binary notation, and the input difference for S-box 68 is $0||0||b$ in binary notation. The 18-bit output difference of the 6 active S-boxes of Round 2 gets involved in a total of 34 bits of the input difference of the S-box operation of Round 3: Bits 2, 7, 17, 21, 25, 36, 40, 49, 60, 64, 70, 78, 93, 102, 106, 113, 117, 121, 123, 128, 151, 155, 159, 170, 174, 181, 185, 204, 212, 223, 227, 234, 238 and 242. Among the 34 bits, only 13 bits are involved in the 30 S-boxes in Ω: Bits 7, 21, 60, 64, 78, 106, 113, 117, 151, 181, 204, 212 and 234; and they are for S-boxes 2, 7, 20, 21, 26, 35, 37, 39, 50, 60, 68, 70 and 78. The values for the 13 bits depend on 10 bits of the output difference of 5 active S-boxes of Round 2:

- The most significant two bits of the output difference of S-box 0, and we denote them by $e||d$;
- The most significant two bits of the output difference of S-box 23, and we denote them by $g||f$;
- The three bits of the output difference of S-box 41, and we denote them by $n||m||h$;
- The most and least significant bits of the output difference of S-box 50, and we denote them by $u||q$; and
- The least significant bit of the output difference of S-box 68, and we label the bit v.

As a result, we have the following results in Round 3: S-box 2 has an input difference $0||n||0$, and the second least significant bit, labeled by x_0, of its output difference is concerned by $\Gamma\gamma$; S-box 7 has an input difference $0||0||v$, and the least significant bit, labeled by x_1, of its output difference is concerned by $\Gamma\gamma$; S-box 20 has an input difference $0||0||m$, and the least significant bit, labeled by x_2, of its output difference is concerned by $\Gamma\gamma$; S-box 21 has an input difference $0||g||0$, and the second least significant bit, labeled by x_3, of its output difference is concerned by $\Gamma\gamma$; S-box 26 has an input difference $0||0||q$, and the least significant bit, labeled by x_4, of its output difference is concerned by $\Gamma\gamma$; S-box 35 has an input difference $0||u||0$, and the most significant bit, labeled respectively by x_5, of its output difference is concerned by $\Gamma\gamma$; S-box 37 has an input difference $h||0||0$, and the least significant bit, labeled by x_6, of its output difference is concerned by $\Gamma\gamma$; S-box 39 has an input difference $0||0||f$, and the least significant bit, labeled by x_7, of its output difference is concerned by $\Gamma\gamma$; S-box 50 has an input difference $0||e||0$, and the least significant bit, labeled by x_8, of its output difference is concerned by $\Gamma\gamma$; S-box 60 has an input difference $0||m||0$, and the most significant bit, labeled by x_9, of its output difference is concerned by $\Gamma\gamma$; S-box 68 has an input difference $0||0||d$, and the second least significant bit, labeled by x_{10}, of its output difference is concerned by $\Gamma\gamma$; S-box 70 has an input difference $q||0||0$, and the second least significant bit, labeled by x_{11}, of its output difference is concerned by $\Gamma\gamma$; and S-box 78 has an input difference $0||0||h$, and the second least significant bit, labeled by x_{12}, of its output difference is concerned by $\Gamma\gamma$.

Now, whether $\beta \odot \gamma = 0$ is equivalent to whether $\bigoplus_{i=0}^{12} x_i = 0$.

By the difference distribution table of the S-box, we get the possible values for $(e||d)$, $(g||f)$, $(n||m||h)$, $(u||q)$, v, $x_0, x_1, \ldots, x_{12}$ and the conditional probabilities, as follows:

$\Pr((e||d) = 0|a = 0) = 1$, $\Pr((e||d) = 0|a = 1) = 0.25$,
$\Pr((e||d) = 1|a = 1) = 0.25$, $\Pr((e||d) = 2|a = 1) = 0.25$,
$\Pr((e||d) = 3|a = 1) = 0.25$, $\Pr((g||f) = 0|b = 0) = 1$,
$\Pr((g||f) = 1|b = 1) = 0.5$, $\Pr((g||f) = 3|b = 1) = 0.5$,
$\Pr((n||m||h) = 0|a = 0) = 1$, $\Pr((n||m||h) = 1|a = 1) = 0.25$,
$\Pr((n||m||h) = 2|a = 1) = 0.25$, $\Pr((n||m||h) = 4|a = 1) = 0.25$,
$\Pr((n||m||h) = 7|a = 1) = 0.25$, $\Pr((u||q) = 0|c = 0) = 1$,
$\Pr((u||q) = 0|c = 1) = 0.25$, $\Pr((u||q) = 1|c = 1) = 0.25$,
$\Pr((u||q) = 2|c = 1) = 0.25$, $\Pr((u||q) = 3|c = 1) = 0.25$,
$\Pr(v = 0|b = 0) = 1$, $\Pr(v = 0|b = 1) = 0.5$,
$\Pr(v = 1|b = 1) = 0.5$, $\Pr(x_0 = 0|n = 0) = 1$,
$\Pr(x_0 = 1|n = 1) = 1$, $\Pr(x_1 = 0|v = 0) = 1$,
$\Pr(x_1 = 0|v = 1) = 0.5$, $\Pr(x_1 = 1|v = 1) = 0.5$,
$\Pr(x_2 = 0|m = 0) = 1$, $\Pr(x_2 = 0|m = 1) = 0.5$,
$\Pr(x_2 = 1|m = 1) = 0.5$, $\Pr(x_3 = 0|g = 0) = 1$,
$\Pr(x_3 = 1|g = 1) = 1$, $\Pr(x_4 = 0|q = 0) = 1$,
$\Pr(x_4 = 0|q = 1) = 0.5$, $\Pr(x_4 = 1|q = 1) = 0.5$,
$\Pr(x_5 = 0|u = 0) = 1$, $\Pr(x_5 = 0|u = 1) = 0.5$,
$\Pr(x_5 = 1|u = 1) = 0.5$, $\Pr(x_6 = 0|h = 0) = 1$,
$\Pr(x_6 = 1|h = 1) = 1$, $\Pr(x_7 = 0|f = 0) = 1$,
$\Pr(x_7 = 0|f = 1) = 0.5$, $\Pr(x_7 = 1|f = 1) = 0.5$,
$\Pr(x_8 = 0|e = 0) = 1$, $\Pr(x_8 = 0|e = 1) = 0.5$,
$\Pr(x_8 = 1|e = 1) = 0.5$, $\Pr(x_9 = 0|m = 0) = 1$,
$\Pr(x_9 = 0|m = 1) = 0.5$, $\Pr(x_9 = 1|m = 1) = 0.5$,
$\Pr(x_{10} = 0|d = 0) = 1$, $\Pr(x_{10} = 0|d = 1) = 0.5$,
$\Pr(x_{10} = 1|d = 1) = 0.5$, $\Pr(x_{11} = 0|q = 0) = 1$,
$\Pr(x_{11} = 0|q = 1) = 0.5$, $\Pr(x_{11} = 1|q = 1) = 0.5$,
$\Pr(x_{12} = 0|h = 0) = 1$, $\Pr(x_{12} = 0|h = 1) = 0.5$,
$\Pr(x_{12} = 1|h = 1) = 0.5$.

For each difference ω, we denote by β_ω the output difference(s) immediately after Round 3, and using the above conditional probabilities we compute the probability of $\bigoplus_{i=0}^{12} x_i = 0$ by performing a program on all the possible (truncated) differential characteristics, which takes a few seconds on a personal computer. These probabilities are given in the third row of Table 4.

Thus, we have $\widehat{p} = \sum_\omega \Pr_{\mathbf{S}}(\Delta 0x1 \to \Delta\omega) \times \Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0|\Delta 0x1 \to \Delta\omega) = 0.5234375$. By Theorem 2 we have that the probability of the 8.5-round distinguisher is $\frac{1}{2} + 2 \times (2^{-33})^2 \times [2 \times 0.5234375 - 1] \approx \frac{1}{2} + 2 \times 2^{-66} \times 2^{-4.42} \approx \frac{1}{2} + 2^{-69.42}$. Therefore, the 8.5-round differential-linear distinguisher $\Delta\alpha \to \Gamma\delta$ has a bias of $2^{-69.42}$.

5.4 Differential-linear attack on 10-round CTC2 with a 255-bit block size and key

The above 8.5-round distinguisher enables us to construct a differential-linear attack breaking the CTC2 when used with 10 rounds under a 255-bit block size and a 255-bit key.

We assume that the attacked rounds are the first ten rounds from Rounds 1 to 10; and we use the distinguisher from Rounds 2 until before the **D** operation of Round 10. As mentioned earlier, we learn that the input difference α propagates to 16 bit positions after the inverse of the **D** operation of Round 1: Bits 17, 21, 40, 59, 78, 97, 116, 135, 139, 154, 158, 177, 196, 215, 234 and 253. The 16 active bits correspond to 16 S-boxes of Round 0: S-boxes 5, 7, 13,

19, 26, 32, 38, 45, 46, 51, 52, 59, 65, 71, 78 and 84; let Θ be the set of the 16 S-boxes, and K_Θ be the 48 bits of K_0 corresponding to the 16 S-boxes in Θ. Another observation is that we do not need to guess the subkey bits from K_{10}, because the output mask $\Gamma\delta$ of the 8.5-round distinguisher concerns the intermediate value immediately after the **S** operation of Round 10, and for a pair of ciphertexts $(C, \widehat{C})$ the value of $\delta \odot \mathbf{D}^{-1}(C \oplus K_{10}) \oplus \delta \odot \mathbf{D}^{-1}(\widehat{C} \oplus K_{10})$ equals $\delta \odot \mathbf{D}^{-1}(C \oplus \widehat{C})$, which is independent of K_{10}. The attack procedure is as follows.

1. Choose $2^{96.84}$ structures $\mathcal{S}_i$, $(i = 0, 1, \ldots, 2^{96.84} - 1)$, where a structure is defined to be a set of 2^{48} plaintexts $P_{i,j}$ with the 48 bits for the S-boxes in Θ taking all the possible values and the other 207 bits fixed, $(j = 0, 1, \ldots, 2^{48} - 1)$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{48} plaintexts in each of the $2^{96.84}$ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$.
2. Initialize 2^{48} counters to zero, which correspond to all the possible values for K_Θ.
3. For every structures $\mathcal{S}_i$, guess a value for K_Θ, and do as follows.
   (a) Partially encrypt every (unpaired) plaintext $P_{i,j}$ with the guessed K_Θ to get its intermediate value immediately after the **S** operation of Round 1; we denote it by $\varepsilon_{i,j}$.
   (b) Take bitwise complements to bits (17, 21, 40, 59, 78, 97, 116, 135, 139, 154, 158, 177, 196, 215, 234, 253) of $\varepsilon_{i,j}$, and keep the other bits of $\varepsilon_{i,j}$ invariant; we denote the resulting value by $\widehat{\varepsilon}_{i,j}$.
   (c) Partially decrypt $\widehat{\varepsilon}_{i,j}$ with the guessed K_Θ to get its plaintext, and find the plaintext in $\mathcal{S}_i$; we denote it by $\widehat{P}_{i,j}$, and denote by $\widehat{C}_{i,j}$ the corresponding ciphertext for $\widehat{P}_{i,j}$.
   (d) For every ciphertext pair $(C_{i,j}, \widehat{C}_{i,j})$, compute the XOR of bits 32 and 151 of $\mathbf{D}^{-1}(C_{i,j} \oplus \widehat{C}_{i,j})$. If the XOR is zero, add one to the counter corresponding to the guessed K_Θ.
4. Output the guess for K_Θ with the highest deviation from $2^{142.84}$.

The attack requires $2^{144.84}$ chosen plaintexts. Note that we start to collect another structure of plaintexts only after testing a structure of plaintexts, so that we can reuse the memory for storing the structure of plaintexts, hence the required memory of the attack is dominated by the storage of the 2^{48} counters and a structure of 2^{48} plaintext-ciphertext pairs, which is $2^{48} \times \frac{143.84}{8} + 2 \times 2^{48} \times \frac{255}{8} \approx 2^{54.4}$ bytes of memory. The time complexity of Step 3 is dominated by the time complexity of Steps 3(a), 3(c) and 3(d), which is approximately $2 \times 2^{143.84} \times 2^{48} \times \frac{16}{85 \times 10} + 2^{143.84} \times 2^{48} \times \frac{1}{10} \approx 2^{188.98}$ 10-round CTC2 encryptions. There are $2^{143.84}$ plaintext pairs $(P_{i,j}, \widehat{P}_{i,j})$ for a guess of K_Θ. Following Theorem 1, we learn that the attack has a success probability of about $\Phi(|2(\frac{1}{2} + 2^{-69.42}) - 1|\sqrt{2^{143.84}} - \Phi^{-1}(1 - \frac{1}{2^{48}})) \approx 99\%$. Therefore, the attack has a time complexity of about $2^{188.98}$ 10-round CTC2 encryptions to find the 48 key bits.

The remaining 207 key bits can be found with exhaustive search, which means that it takes a total time complexity of approximately 2^{207} 10-round CTC2 encryptions to find the full user key. We note that it may be possible to recover the whole key with a lower time complexity by exploiting one or more differential-linear attacks to recover the remaining 207 key bits, instead of exhaustive search on them.

6 Application to the Serpent block cipher

The Serpent [1,2] block cipher is one of the five Advanced Encryption Standard (AES) finalists, second to the Rijndael [18] cipher that was selected as the AES [45]. It has a 128-bit block size, a variable length user key of up to 256 bits, and a total of 32 rounds. Serpent

was designed in a rather conservative way, and it was included in the GNU project [24] for possible use in real-world cryptographic applications.

In 2003, Biham et al. [12] described a 9-round differential-linear distinguisher of Serpent, and finally gave a differential-linear attack on 11-round Serpent (with a 256-bit key). In 2008 Dunkelman, Indesteege and Keller [23] presented an improved 9-round differential-linear distinguisher of Serpent, and finally used it as the basis for a differential-linear attack on 12-round Serpent. All these attacks are based on Langford and Biham et al.'s methodology, more or less. The 12-round attack is known as the best previously published cryptanalytic result on Serpent in terms of the numbers of attacked rounds.

In this section, we construct a 9-round differential-linear distinguisher with bias $2^{-59.41}$ under our new methodology, which can be used to break 12-round Serpent (with a 256-bit key) slightly faster than Dunkelman et al.'s attack at a higher success rate. We first briefly describe the Serpent block cipher.

6.1 The Serpent block cipher

The Serpent [1,2] block cipher has a 128-bit block size, a variable length key of up to 256 bits, and a total of 32 rounds; a shorter key can be used by appending one "1" bit to the most significant bit end, followed by as many "0" bits as required. Serpent uses the following elementary operations:

- **IP**/**FP** is the initial/final permutation; see [2] for their specifications.
- $\mathbf{S}_i$ is a non-linear substitution operation constructed by applying the same 4×4-bit bijective $S_{i \bmod 8}$ S-box 32 times in parallel to an input, $(0 \le i \le 31)$. Refer to [2] for specifications of the S-boxes $\mathrm{S}_0, \mathrm{S}_1, \ldots, \mathrm{S}_7$.
- **L** is a linear diffusion operation, which takes as input a 128-bit block of four 32-bit words $X = (X_3, X_2, X_1, X_0)$, and outputs a 128-bit block of four 32-bit words $Y = (Y_3, Y_2, Y_1, Y_0)$, computed as follows.
  - $X_0 = X_0 \lll 13$,
  - $X_2 = X_2 \lll 3$,
  - $X_1 = X_0 \oplus X_1 \oplus X_2$,
  - $X_3 = X_3 \oplus X_2 \oplus (X_0 \ll 3)$,
  - $X_1 = X_1 \lll 1$,
  - $X_3 = X_3 \lll 7$,
  - $X_0 = X_0 \oplus X_1 \oplus X_3$,
  - $X_2 = X_2 \oplus X_3 \oplus (X_1 \ll 7)$,
  - $X_0 = X_0 \lll 5$,
  - $X_2 = X_2 \lll 22$,
  - $Y = (X_3, X_2, X_1, X_0)$.

Serpent takes as input a 128-bit plaintext block P, and its encryption procedure is, where $\widehat{B}_0, \widehat{B}_1, \ldots, \widehat{B}_{32}$ are 128-bit variables, and $K_0, K_1, \ldots, K_{32}$ are round keys.

1. $\widehat{B}_0 = \mathbf{IP}(P)$.
2. For $i = 0$ to 30:
   - $\widehat{B}_{i+1} = \mathbf{L}(\mathbf{S}_i(\widehat{B}_i \oplus K_i))$.
3. $\widehat{B}_{32} = \mathbf{S}_{31}(\widehat{B}_{31} \oplus K_{31}) \oplus K_{32}$.
4. Ciphertext $= \mathbf{FP}(\widehat{B}_{32})$.

The ith iteration of Step 2 in the above description is referred to below as Round i, $(0 \le i \le 30)$, and the transformation in Steps 3 and 4 is referred to below as Round 31; this is

in accordance with [2]. We number the 32 S-boxes of a round from 0 to 31 from right to left. For simplicity, we describe a state S in a Serpent encryption as four 32-bit words (s_3, s_2, s_1, s_0), and write it as $(s_{3,31}||s_{2,31}||s_{1,31}||s_{0,31})||\cdots||(s_{3,1}||s_{2,1}||s_{1,1}||s_{0,1})||(s_{3,0}||s_{2,0}||s_{1,0}||s_{0,0})$, where $s_{j,l}$ is the lth bit of s_j, $(0 \leq j \leq 3, 0 \leq l \leq 31)$. We write $K_{i,m}$ for the 4-bit subkey of K_i that corresponds to S-box m of Round i, $(0 \leq m \leq 31)$. As the **IP** and **FP** operations are simply linear diffusion transformations, we omit them in our analysis.

6.2 A 9-round differential-linear distinguisher with bias $2^{-59.41}$

The 9-round differential-linear distinguisher with bias $2^{-59.41}$ is made up of a 6-round linear approximation $\Gamma\gamma \rightarrow \Gamma\delta$ with bias 2^{-27} for Rounds 5 to 10 and all the 3-round differentials $\{\Delta\alpha \rightarrow \Delta\beta\}$ for Rounds 2 to 4 with $\Delta\alpha = 0x000000A00000000000000000000000000$. The 6-round linear approximation $\Gamma\gamma \rightarrow \Gamma\delta$ is $0x00400000000000000000000000000002 \rightarrow 0x000B0000B\ 000030000B0200E00000010$, which is from the $\mathbf{S}_5$ operation until immediately before the last $\mathbf{S}_3$ operation of the 9-round linear approximation given in [9], with only the input mask slightly changed.[4] The input difference and linear approximation used in our 9-round differential-linear distinguisher are different from those for previous 9-round differential-linear distinguishers described in [12,23]. Below we use the new methodology to compute the probability of the 9-round differential-linear distinguisher, where the concerned differences in the S-boxes of the 3-round differentials for the 9-round differential-linear distinguisher are depicted in Fig. 5.

By the **L** operation we know that the input mask $\Gamma\gamma$ concerns a total of 3 bits of the output differences of three S_4 S-boxes in Round 4:

- The most significant bit of the output difference of S-box 0, and we label it x_0;
- The second most significant bit of the output difference of S-box 4, and we label it x_1; and
- The second most significant bit of the output difference of S-box 29, and we label it x_2.

Given the input difference $\Delta\alpha$, there is only one active S-box among the 32 S_2 S-boxes of Round 2, which generates 6 possible output differences: $\{\omega|\omega = 0x2, 0x4, 0x6, 0x8, 0xA, 0xE\}$; the probabilities for these output differences are given in the second column of Table 5, and the difference distribution tables of the eight S-boxes were presented in [49]. We write ω as $d||c||b||a$ in binary notation, where $a, b, c, d \in \{0, 1\}$.

The 6 possible output differences $\{\omega\}$ may affect at most 18 S-boxes of Round 3, and a simple analysis reveals that only fifteen of them relate to the input differences of the three S-boxes of Round 4 concerned by the input mask. We now focus on the fifteen S_3 S-boxes in Round 3. S-box 0 has an input difference $d000$ in binary notation, and we denote the most significant bit and the second most significant bit of its output difference by y_0, y_1, respectively. S-box 2 has an input difference $000c$ in binary notation, and we denote the second least significant bit of its output difference by y_2. S-box 3 has an input difference $c000$ in binary notation, and we denote the most significant bit, the second most significant bit and the second least significant bit of its output difference by y_3, y_4, y_5, respectively. S-box 4 has an input difference $0a00$ in binary notation, and we denote the second most significant bit and the least significant bit of its output difference by y_6, y_7, respectively. S-box 6 has an input difference $0a00$ in binary notation, and we denote the second least significant bit and the least significant bit of its output difference by y_8, y_9, respectively. S-box 7 has an input difference $00a0$ in binary notation, and we denote the most significant bit and the second most

[4] Note that there is a typo in the input mask of Round 7 of the 9-round linear approximation in [9], where the input mask should be $0x00000010000A00000000000000000000$.

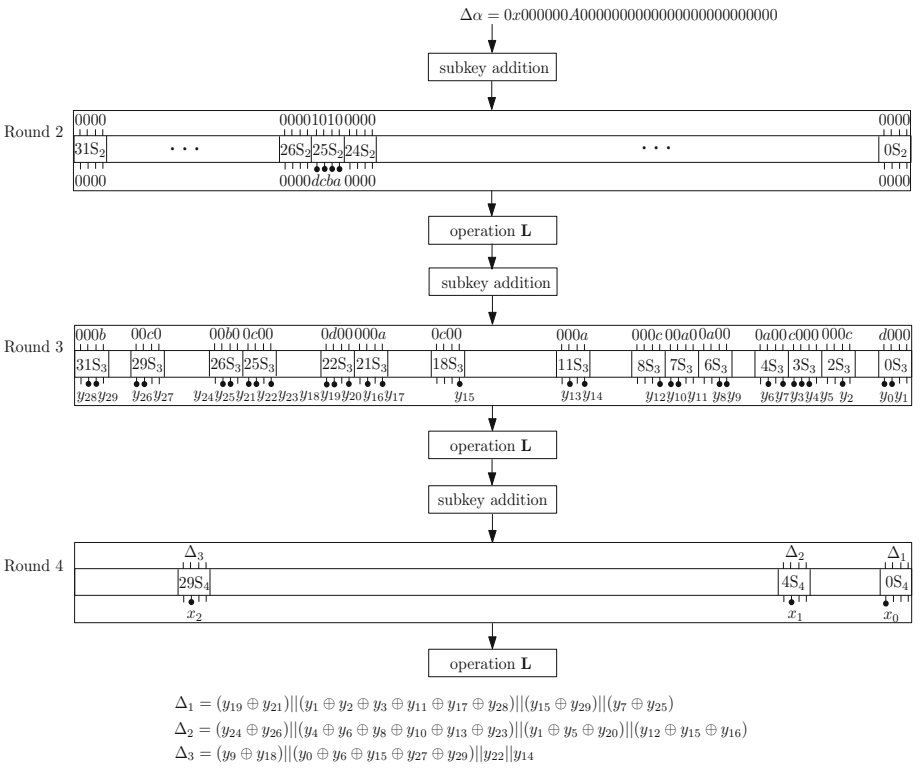

Fig. 5 Concerned differences in the S-boxes of the 3-round differentials for the 9-round differential-linear distinguisher of Serpent

Table 5 Probabilities for the six output differences in $\{\omega\}$

Difference (ω)	$\Pr_{S_2}(\Delta 0xA \rightarrow \Delta\omega)$	$\Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0 \mid \Delta 0xA \rightarrow \Delta\omega)$
$0x2$	2^{-3}	0.453125
$0x4$	2^{-3}	0.502197265625
$0x6$	2^{-2}	0.5
$0x8$	2^{-3}	0.49609375
$0xA$	2^{-2}	0.500732421875
$0xE$	2^{-3}	0.5

significant bit of its output difference by y_{10}, y_{11}, respectively. S-box 8 has an input difference $000c$ in binary notation, and we denote the least significant bit of its output difference by y_{12}. S-box 11 has an input difference $000a$ in binary notation, and we denote the second most significant bit and the least significant bit of its output difference by y_{13}, y_{14}, respectively. S-box 18 has an input difference $0c00$ in binary notation, and we denote the least significant bit of its output difference by y_{15}. S-box 21 has an input difference $000a$ in binary notation, and we denote the second most significant bit and the least significant bit of its output difference by y_{16}, y_{17}, respectively.S-box 22 has an input difference $0d00$ in binary notation, and we

denote the most significant bit, the second most significant bit and the least significant bit of its output difference by y_{18}, y_{19}, y_{20}, respectively. S-box 25 has an input difference $0c00$ in binary notation, and we denote the most significant bit, the second most significant bit and the least significant bit of its output difference by y_{21}, y_{22}, y_{23}, respectively. S-box 26 has an input difference $00b0$ in binary notation, and we denote the second most significant bit and the second least significant bit of its output difference by y_{24}, y_{25}, respectively. S-box 29 has an input difference $00c0$ in binary notation, and we denote the most significant bit and the second most significant bit of its output difference by y_{26}, y_{27}, respectively. S-box 31 has an input difference $000b$ in binary notation, and we denote the second most significant bit and the second least significant bit of its output difference by y_{28}, y_{29}, respectively.

As a result, we get the input differences of the concerned three S_4 S-boxes of Round 4:

- The input difference for S-box 0 is $(y_{19} \oplus y_{21})||(y_1 \oplus y_2 \oplus y_3 \oplus y_{11} \oplus y_{17} \oplus y_{28})||(y_{15} \oplus y_{29})||(y_7 \oplus y_{25})$;
- The input difference for S-box 4 is $(y_{24} \oplus y_{26})||(y_4 \oplus y_6 \oplus y_8 \oplus y_{10} \oplus y_{13} \oplus y_{23})||(y_1 \oplus y_5 \oplus y_{20})||(y_{12} \oplus y_{15} \oplus y_{16})$; and
- The input difference for S-box 29 is $(y_9 \oplus y_{18})||(y_0 \oplus y_6 \oplus y_{15} \oplus y_{27} \oplus y_{29})||y_{22}||y_{14}$.

By the difference distribution table of the S_3 S-box, we get all the possible values for $(y_0||y_1)$, y_2, $(y_3||y_4||y_5)$, $(y_6||y_7)$, $(y_8||y_9)$, $(y_{10}||y_{11})$, y_{12}, $(y_{13}||y_{14})$, y_{15}, $(y_{16}||y_{17})$, $(y_{18}||y_{19}||y_{20})$, $(y_{21}||y_{22}||y_{23})$, $(y_{24}||y_{25})$, $(y_{26}||y_{27})$ and $(y_{28}||y_{29})$ as well as their probabilities, as follows.

$\Pr((y_0||y_1) = 0x0|d = 1) = 0.125$, $\Pr((y_0||y_1) = 0x1|d = 1) = 0.125$,
$\Pr((y_0||y_1) = 0x2|d = 1) = 0.125$, $\Pr((y_0||y_1) = 0x3|d = 1) = 0.625$,
$\Pr(y_2 = 0x0|c = 1) = 0.5$, $\Pr(y_2 = 0x1|c = 1) = 0.5$,
$\Pr((y_3||y_4||y_5) = 0x1|c = 1) = 0.125$, $\Pr((y_3||y_4||y_5) = 0x3|c = 1) = 0.125$,
$\Pr((y_3||y_4||y_5) = 0x4|c = 1) = 0.125$, $\Pr((y_3||y_4||y_5) = 0x6|c = 1) = 0.375$,
$\Pr((y_3||y_4||y_5) = 0x7|c = 1) = 0.25$, $\Pr((y_6||y_7) = 0x0|a = 1) = 0.125$,
$\Pr((y_6||y_7) = 0x1|a = 1) = 0.125$, $\Pr((y_6||y_7) = 0x2|a = 1) = 0.375$,
$\Pr((y_6||y_7) = 0x3|a = 1) = 0.375$, $\Pr((y_8||y_9) = 0x0|a = 1) = 0.125$,
$\Pr((y_8||y_9) = 0x1|a = 1) = 0.125$, $\Pr((y_8||y_9) = 0x2|a = 1) = 0.375$,
$\Pr((y_8||y_9) = 0x3|a = 1) = 0.375$, $\Pr((y_{10}||y_{11}) = 0x1|a = 1) = 0.25$,
$\Pr((y_{10}||y_{11}) = 0x2|a = 1) = 0.5$, $\Pr((y_{10}||y_{11}) = 0x3|a = 1) = 0.25$,
$\Pr(y_{12} = 0x0|c = 1) = 0.25$, $\Pr(y_{12} = 0x1|c = 1) = 0.75$,
$\Pr((y_{13}||y_{14}) = 0x1|a = 1) = 0.25$, $\Pr((y_{13}||y_{14}) = 0x2|a = 1) = 0.25$,
$\Pr((y_{13}||y_{14}) = 0x3|a = 1) = 0.5$, $\Pr(y_{15} = 0x0|c = 1) = 0.5$,
$\Pr(y_{15} = 0x1|c = 1) = 0.5$, $\Pr((y_{16}||y_{17}) = 0x1|a = 1) = 0.25$,
$\Pr((y_{16}||y_{17}) = 0x2|a = 1) = 0.25$, $\Pr((y_{16}||y_{17}) = 0x3|a = 1) = 0.5$,
$\Pr((y_{18}||y_{19}||y_{20}) = 0x2|d = 1) = 0.25$, $\Pr((y_{18}||y_{19}||y_{20}) = 0x3|d = 1) = 0.25$,
$\Pr((y_{18}||y_{19}||y_{20}) = 0x4|d = 1) = 0.125$, $\Pr((y_{18}||y_{19}||y_{20}) = 0x5|d = 1) = 0.125$,
$\Pr((y_{18}||y_{19}||y_{20}) = 0x6|d = 1) = 0.125$, $\Pr((y_{18}||y_{19}||y_{20}) = 0x7|d = 1) = 0.125$,
$\Pr((y_{21}||y_{22}||y_{23}) = 0x2|c = 1) = 0.25$, $\Pr((y_{21}||y_{22}||y_{23}) = 0x3|c = 1) = 0.25$,
$\Pr((y_{21}||y_{22}||y_{23}) = 0x4|c = 1) = 0.125$, $\Pr((y_{21}||y_{22}||y_{23}) = 0x5|c = 1) = 0.125$,
$\Pr((y_{21}||y_{22}||y_{23}) = 0x6|c = 1) = 0.125$, $\Pr((y_{21}||y_{22}||y_{23}) = 0x7|c = 1) = 0.125$,
$\Pr((y_{24}||y_{25}) = 0x0|b = 1) = 0.125$, $\Pr((y_{24}||y_{25}) = 0x1|b = 1) = 0.375$,
$\Pr((y_{24}||y_{25}) = 0x2|b = 1) = 0.125$, $\Pr((y_{24}||y_{25}) = 0x3|b = 1) = 0.375$,
$\Pr((y_{26}||y_{27}) = 0x1|c = 1) = 0.25$, $\Pr((y_{26}||y_{27}) = 0x2|c = 1) = 0.5$,
$\Pr((y_{26}||y_{27}) = 0x3|c = 1) = 0.25$, $\Pr((y_{28}||y_{29}) = 0x1|b = 1) = 0.25$,
$\Pr((y_{28}||y_{29}) = 0x2|b = 1) = 0.5$, $\Pr((y_{28}||y_{29}) = 0x3|b = 1) = 0.25$.

Table 6 Probabilities for S_4

Difference (ω)	$0x0$	$0x1$	$0x2$	$0x3$	$0x4$	$0x5$	$0x6$	$0x7$	$0x8$	$0x9$	$0xA$	$0xB$	$0xC$	$0xD$	$0xE$	$0xF$
$\Pr(x_0 = 0\|\omega)$	1	0.25	0.5	0.5	0.5	0.25	0.5	0.5	0.25	0.75	0.5	0.5	0.25	0.75	0.5	0.5
$\Pr(x_1 = 0\|\omega)$	1	0.25	0.25	0.75	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.25	0.5	0.75	0.25
$\Pr(x_2 = 0\|\omega)$																

By the difference distribution table of S_4, we compute the conditional probability that for every possible input difference, the concerned bit of the output difference of each concerned S_4 S-box in Round 4 is 0 or 1; the probabilities are given in Table 6, and $\Pr(x_0 = 1|\omega) = 1-\Pr(x_0 = 0|\omega)$, $\Pr(x_1 = 1|\omega) = 1-\Pr(x_1 = 0|\omega)$, and $\Pr(x_2 = 1|\omega) = 1-\Pr(x_2 = 0|\omega)$.

For each difference ω, we denote by β_ω the output difference(s) of the 3-round Serpent. Subsequently, we compute the probability that the XOR of the concerned 3 bits of β_ω is zero (i.e., $\Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0|\Delta 0xA \to \Delta\omega)$) by performing a program on all the possible (truncated) differential characteristics. These probabilities are given in the third column of Table 5. A straightforward implementation takes several seconds on a personal computer.

Hence, by Theorem 2 we have $\widehat{p} = \sum_\omega \Pr_{S_2}(\Delta 0xA \to \Delta\omega) \times \Pr(\Delta\beta_\omega \odot \Gamma\gamma = 0|\Delta 0xA \to \Delta\omega) = 0.494110107421875$, and the probability of the 9-round differential-linear distinguisher $\Delta\alpha \to \Gamma\delta$ is $\frac{1}{2} + 2 \times (2^{-27})^2 \times (2 \times 0.494110107421875 - 1) \approx \frac{1}{2} - 2 \times 2^{-54} \times 2^{-6.41} = \frac{1}{2} - 2^{-59.41}$. Thus, the 9-round differential-linear distinguisher $\Delta\alpha \to \Gamma\delta$ has a bias of $2^{-59.41}$.

We use several strategies to find the 9-round distinguisher.

- First, 9 rounds are the most that we can build for a differential-linear distinguisher on a personal computer. As shown in [14,15], the best currently known 7-round linear approximation has bias 2^{-30}. Thus, if we aim to build a useful distinguisher operating on 10 (or more) rounds by using such a 7-round linear approximation, then the bias for the distinguisher is $2 \times |2\widehat{p} - 1| \times (2^{-30})^2 = |2\widehat{p} - 1| \cdot 2^{-59}$, so the margin for the value of $|2\widehat{p} - 1|$ is tough for using 3-round differentials. Alternatively, we may choose to use 4-round differentials, instead of using a 7-round linear approximation, however, there are a large number of possible differential characteristics for which calculating the probability of $\beta \odot \gamma = 0$ is beyond the computational power of a personal computer.
- Second, to use a 9-round distinguisher to attack 12-round Serpent, we should use such an input difference for the distinguisher that makes a small number of active S-boxes for the two rounds preceding the distinguisher, ideally less than 32, meaning that a small number of unknown key bits are required to guess; another way is to append two more rounds after the distinguisher, however this usually requires us to guess more subkey bits.
- Third, some 3-round differentials should involve as few as possible active bits, and the input mask should concern as few as possible output bits of the **S** operation of the preceding round. After having checked the biases of a number of 9-round distinguishers, we find: Generally speaking, the more active or concerned bits are involved, the smaller is the bias of the distinguisher, and the distinguisher is more likely to be ineffective.

The above 9-round differential-linear distinguisher is the best we have found under these strategies, where the input mask concerns only three output bits of the preceding $\mathbf{S}_4$ operation, and either of the two values $0x2$ and $0x8$ of ω makes only two active S_3 S-boxes in the following round.

6.3 Differential-linear attack on 12-round Serpent

We can use the 9-round differential-linear distinguisher as the basis for a differential-linear attack breaking 12-round Serpent (with a 256-bit key). We attack Rounds 0 to 11, and use the distinguisher from Rounds 2 to 10. The input difference α becomes $0x000000A20400080000000000000000000$ after being applied the inverse of the **L** operation of Round 1, and the 5 active bits correspond to S-boxes 18, 22, 24 and 25 of Round 1. It makes 27 active S-boxes of Round 0: S-boxes 0, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 15, 16, ..., 29 and 31; let Θ be the set of the 27 S-boxes, and K_Θ be the 108 bits of K_0 corresponding to the 27 S-boxes in Θ. The 16 bits concerned by the output mask correspond to S-boxes 1, 8, 11, 13, 18, 23 and 28 of Round 11. The attack procedure is as follows, where the values of parameters λ and ϕ will be specified in the subsequent analysis.

1. Choose λ structures $\mathcal{S}_i$, $(i = 0, 1, \ldots, \lambda - 1)$, where a structure is defined to be a set of 2^{108} plaintexts $P_{i,j}$ with the 108 bits for the 27 S-boxes in Θ taking all the possible values and the other 20 bits fixed, $(j = 0, 1, \ldots, 2^{108} - 1)$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{108} plaintexts in each of the λ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$.
2. Guess a value for $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25})$, and do as follows.
   (a) Initialize 2^{56} counters to zero, which correspond to the 2^{56} possible pairs of the 28 ciphertext bits corresponding to S-boxes 1, 8, 11, 13, 18, 23 and 28 of Round 11.
   (b) Partially encrypt every (unpaired) plaintext $P_{i,j}$ with the guessed $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25})$ to get its intermediate value immediately after the **S** operation of Round 1; we denote it by $\varepsilon_{i,j}$.
   (c) Compute $\varepsilon_{i,j} \oplus 0x000000A20400080000000000000000000$, and we denote the resulting value by $\widehat{\varepsilon}_{i,j}$.
   (d) Partially decrypt $\widehat{\varepsilon}_{i,j}$ with the guessed $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25})$ to get its plaintext, and find the plaintext in $\mathcal{S}_i$; we denote it by $\widehat{P}_{i,j}$, and denote by $\widehat{C}_{i,j}$ the corresponding ciphertext for $\widehat{P}_{i,j}$.
   (e) For every ciphertext pair $(C_{i,j}, \widehat{C}_{i,j})$, add one to the counter corresponding to the pair of the above-mentioned 28 ciphertext bits from $(C_{i,j}, \widehat{C}_{i,j})$.
   (f) Guess a value for $(K_{12,1}, K_{12,8}, K_{12,11}, K_{12,13}, K_{12,18}, K_{12,23}, K_{12,28})$, and do as follows.
      i For each of the 2^{56} pairs of the concerned 28 ciphertext bits, partially decrypt it with the guessed $(K_{12,1}, K_{12,8}, K_{12,11}, K_{12,13}, K_{12,18}, K_{12,23}, K_{12,28})$ to get the pair of the 16 bits concerned by the output mask, and compute the XOR of the pair of the 16 bits (concerned by the output mask).
      ii Count the number of the ciphertext pairs $(C_{i,j}, \widehat{C}_{i,j})$ such that the XOR of the pair of the 16 bits concerned by the output mask is zero, and compute its deviation from $\lambda \cdot 2^{106}$.
      iii If the guess for $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25}, K_{12,1}, K_{12,8}, K_{12,11}, K_{12,13}, K_{12,18}, K_{12,23}, K_{12,28})$ belongs to the first ϕ guesses for $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25}, K_{12,1}, K_{12,8}, K_{12,11}, K_{12,13}, K_{12,18}, K_{12,23}, K_{12,28})$, then record the guess and the deviation computed in Step 2(f)(ii); otherwise, record the guess and its deviation only when the deviation is larger than the smallest deviation of the previously recorded ϕ guesses, and remove the guess with the smallest deviation from the ϕ guesses.

3. For every recorded $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25})$ in Step 2(f)(iii), exhaustively search for the remaining 132 key bits with two known plaintext-ciphertext pairs. If a 256-bit key is suggested, output it as the user key of the 12-round Serpent.

The attack requires $\lambda \times 2^{108}$ chosen plaintexts. The required memory for the attack is dominated by the storage of the plaintexts and ciphertexts, which is $\lambda \times 2^{108} \times 32 = \lambda \times 2^{113}$ bytes. The time complexity of Step 2 is dominated by the time complexity of Steps 2(b), 2(d) and 2(f)(i), which is $\lambda \times 2 \times 2^{107} \times 2^{124} \times \frac{27+4}{32\times 12} + 2 \times 2^{124} \times 2^{28} \times 2^{56} \times \frac{7}{32\times 12} \approx \lambda \times 2^{228.37}$ 12-round Serpent encryptions. Step 3 has a time complexity of at most $\phi \times 2^{132}$ 12-round Serpent encryptions. There are $\lambda \times 2^{107}$ plaintext pairs $(P_{i,j}, \widehat{P}_{i,j})$ for a guess of $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25}, K_{12,1}, K_{12,8}, \ldots, K_{12,28})$. Following Theorem 1, we have that the probability that the correct guess of $(K_\Theta, K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25}, K_{12,1}, K_{12,8}, \ldots, K_{12,28})$ is recorded in Step 2(f)(iii) is about $\Phi(|2(\frac{1}{2} - 2^{-59.41}) - 1|\sqrt{\lambda \times 2^{107}} - \Phi^{-1}(1 - \frac{\phi}{2^{152}})) \approx 99\%$ when we set $\lambda = 2^{16.5}$ and $\phi = 2^{104}$. Thus, with a success probability of about 99% the attack requires $2^{124.5}$ chosen plaintexts, and has a total time complexity of approximately $2^{244.9}$ 12-round Serpent encryptions.

6.4 Notes

For the purpose of then AES submission requirements, the Serpent designers also considered the cases when using a 128/192-bit key, and we denote these versions by Serpent-128/192, respectively. There are some published cryptanalytic results on Serpent-128/192, and we are particularly interested in those differential-linear cryptanalytic results: Biham et al.'s and Dunkelman et al.'s differential-linear attacks on 10-round Serpent-128 and 11-round Serpent-192 given in [12,23]. All these attacks are based on Langford and Biham et al.'s methodology. A detailed analysis shows that the 9-round differential-linear distinguisher described in Sect. 6.2 can also be used to break 11-round Serpent-192 and 10-round Serpent-128, and we brief them below.

6.4.1 Note 1: Differential-linear attack on 11-round Serpent-192

The attack is basically the version of the above 12-round Serpent attack when the first round is removed. Let $\phi = 1$, then we get similarly that with a success probability of about 99% the attack requires $2^{107.2}$ structures of 2^{16} plaintexts with the 16 bits for S-boxes 18, 22, 24 and 25 (of Round 1) taking all the possible values and the other 112 bits fixed, and has a time complexity of $2^{123.2} + 2 \times 2^{122.2} \times 2^{16} \times \frac{4}{32\times 11} + 2 \times 2^{16} \times 2^{28} \times 2^{56} \times \frac{7}{32\times 11} \approx 2^{132.8}$ 11-round Serpent encryptions to find the correct value of $(K_{1,18}, K_{1,22}, K_{1,24}, K_{1,25}, K_{12,1}, K_{12,8}, K_{12,11}, K_{12,13}, K_{12,18}, K_{12,23}, K_{12,28})$. Then for the correct $(K_{12,1}, K_{12,8}, K_{12,11}, K_{12,13}, K_{12,18}, K_{12,23}, K_{12,28})$, we can get the 192-bit key by performing an exhaustive search on the remaining 164 key bits, which takes 2^{164} 11-round Serpent encryptions.

We can reduce the time complexity using a different 9-round differential-linear distinguisher, which is obtained by changing the output mask of the $\mathbf{S}_2$ operation to $0x00001000000000007000010000100001$. Consequently, the output mask $\Gamma\delta$ becomes $0x000B0000B001030220B0200C00400010$, and it now concerns 11 S-boxes in the following round. Since the bias for the linear approximation remains invariant, the distinguisher has a bias $2^{-59.41}$. Similarly, the attack (with $\phi = 1$) requires $2^{109.5}$ structures (as described above) to have a success probability of about 99%, and has a time complexity

of $2^{125.5}+2\times 2^{124.5}\times 2^{16}\times\frac{4}{32\times 11}+2\times 2^{16}\times 2^{44}\times 2^{88}\times\frac{11}{32\times 11}\approx 2^{144}$ 11-round encryptions to find the correct value for ($K_{1,18}$, $K_{1,22}$, $K_{1,24}$, $K_{1,25}$, $K_{12,1}$, $K_{12,5}$, $K_{12,8}$, $K_{12,11}$, $K_{12,13}$, $K_{12,15}$, $K_{12,16}$, $K_{12,18}$, $K_{12,20}$, $K_{12,23}$, $K_{12,28}$). Given the correct ($K_{12,1}$, $K_{12,5}$, ..., $K_{12,28}$), an exhaustive search for the remaining 148 key bits takes 2^{148} 11-round encryptions. Therefore, the attack has a total time complexity of approximately $2^{144}+2^{148}\approx 2^{148.1}$ 11-round Serpent encryptions to find the 192-bit key.

We can also perform an attack without using the counters; it is a time-memory tradeoff to the above 11-round Serpent-192 attack. For every guess of ($K_{1,18}$, $K_{1,22}$, $K_{1,24}$, $K_{1,25}$, $K_{12,1}$, $K_{12,8}$, ..., $K_{12,28}$), we first get the ciphertext pairs as in Steps 2(b)–(d), and then partially decrypt every ciphertext pair to get the XOR of the pair of the 16 bits concerned by the output mask, and finally perform as in Steps 2(f)(ii), 2(f)(iii) and 3 (with $\phi=1$). The resulting attack has a total time complexity of approximately $2\times 2^{122.2}\times 2^{44}\times\frac{7}{32\times 11}+2^{164}\approx 2^{164.3}$ 11-round Serpent encryptions.

6.4.2 Note 2: Differential-linear attack on 10-round Serpent-128

The attack is basically the version of the 12-round Serpent attack when the first two rounds are removed. Let $\phi=1$; then with a success probability of about 99% the attack requires $2^{122.4}$ plaintext pairs with difference α, and has a time complexity of $2^{123.4}+2\times 2^{28}\times 2^{56}\times\frac{7}{32\times 10}\approx 2^{123.4}$ 10-round Serpent encryptions to find the correct value for ($K_{1,18}$, $K_{1,22}$, $K_{1,24}$, $K_{1,25}$). Given the recorded ($K_{1,18}$, $K_{1,22}$, $K_{1,24}$, $K_{1,25}$), we can get the 128-bit key by performing an exhaustive search on the remaining 112 key bits, which takes 2^{112} 10-round Serpent encryptions.

We also find some 8-round and several 9-round (different) differential-linear distinguishers that can be used to break 10-round Serpent-128, like the two to be described in the latter part of Note 3. Nevertheless, among them only one differential-linear distinguisher can be used to break 10-round Serpent-128 without requiring additional memory for similar counters, and it is 9-round. This 9-round differential-linear distinguisher consists of a 6.5-round linear approximation $\Gamma\gamma\rightarrow\Gamma\delta$ with bias 2^{-27} and all the 2.5-round differentials $\{\Delta\alpha\rightarrow\Delta\beta\}$ that meet $\beta\odot\gamma=0$ with $\Delta\alpha=0x00000000000000090000000000000\ 00000$. The 6.5-round linear approximation $\Gamma\gamma\rightarrow\Gamma\delta$ is $0x000B0000B0000300\ 00B0200E00000010\rightarrow 0x00400000000000000000000000040008$, which operates on 6.5 consecutive rounds from the **L** operation immediately before the $\mathbf{S}_3$ operation until the **L** operation immediately after the $\mathbf{S}_4$ operation in the decryption direction (It is quite similar to the corresponding 6.5 rounds from the 9-round linear approximation with bias 2^{-50} given in [15]). The 2.5-round differentials $\{\Delta\alpha\rightarrow\Delta\beta\}$ operate on 2.5 consecutive rounds in the decryption direction, from the $\mathbf{S}_5$ S-boxes of Round 13 until the $\mathbf{S}_3$ S-boxes of Round 11. We can similarly get that the distinguisher has a bias of $2^{-59.01}$. An important property of the distinguisher is that $\Gamma\delta$ concerns only three S_4 S-boxes of the preceding round. Likewise, with a success probability of 99%, the attack requires $2^{121.2}$ ciphertext pairs with difference α, and has a total time complexity of $2^{122.2}+2\times 2^{121.2}\times 2^{12}\times\frac{3}{32\times 10}+2^{112}\approx 2^{127.5}$ 10-round Serpent encryptions to find the 128-bit key, where we use $\phi=1$. We can also use the distinguisher to break 11-round Serpent-192 by appending one round after the round with $\Delta\alpha$, but anyway cannot use it to break 12-round Serpent with a 256-bit key, because there are a large number of required unknown subkey bits in the extra round.

It is worthy to note that in both the above 10-round Serpent-128 attacks we can reduce the data and time complexity by using a reasonably greater ϕ, while keeping the same success probability.

6.4.3 Note 3: A few 9-round differential-linear distinguishers

We have checked the biases of many 9-round differential-linear distinguishers, and a few of them can be used to attack 10-round Serpent-128 or 11-round Serpent-192, but almost all of them cannot be used to nontrivially break 12-round Serpent with a 256-bit key. For instance, we have computed the bias of the 9-round differential-linear distinguisher constructed by replacing the input difference of the 9-round differential-linear distinguisher described in Sect. 6.2 with $0x00000000000004000000000000000000$ and replacing the input mask with $0x00200000000000000000000000000002$ (the intermediate masks remain unchanged). This input mask is also used as the input mask of the 6-round linear approximation in Biham et al.'s 9-round differential-linear distinguisher. The reason for choosing the input difference is because it causes a minimal number of active bits for the following $\mathbf{S}_3$ operation. As a result, we have gotten that the probability of $\beta \odot \gamma = 0$ is approximately 0.49988768994808197. Besides, we have also checked the 9-round differential-linear distinguisher when the input difference is $0x00000000000000040000000000000000$ (and the input mask is $0x00200000000000000000000000000002$), and the probability of $\beta \odot \gamma = 0$ is approximately 0.50003069639205 93. Both distinguishers have a bias smaller than 2^{-64}, and are not useful.

Interestingly, we find such a 9-round differential-linear distinguisher that for every possible difference ω the probability of $\beta_\omega \odot \gamma = 0$ is equal or surprisingly close to $\frac{1}{2}$, which is obtained by replacing the input difference of the 9-round differential-linear distinguisher described in Sect. 6.2 with $0x00000000000004000000000000000000$ and replacing the input mask with $0x00E000000000000000000000000000E$ (the intermediate masks keep invariant). The reason for choosing the input mask is that it will make a 6-round linear approximation with bias 2^{-25}, thus improving a factor of 4 over the one used above. The input difference generates four possible output differences after the active S_2 S-box: $\{\omega | \omega = 0x6, 0xA, 0xB, 0xD\}$, each with probability 2^{-2}; the probability of $\beta_\omega \odot \gamma = 0$ for $\omega \in \{0x6, 0xA\}$ is 0.5, the probability of $\beta_\omega \odot \gamma = 0$ for $\omega = 0xB$ is 0.50000000018189894, and the probability of $\beta_\omega \odot \gamma = 0$ for $\omega = 0xD$ is 0.50000000000499334. Hence, the total probability of $\beta \odot \gamma = 0$ is approximately 0.50000000004672309, surprisingly close to $\frac{1}{2}$.

The second best 9-round differential-linear distinguisher which we have found might be potentially used to break 12-round Serpent with a 256-bit key is the one described in Sect. 6.2 with the input difference being replaced by $0x00000000000000000000005000000000$ (keeping the other parts unchanged). The probability of $\beta \odot \gamma = 0$ is approximately 0.49902951717376709, and thus the distinguisher has a bias of approximately $2 \times 2^{-9} \times (2^{-27})^2 = 2^{-62}$, larger than 2^{-64}. There are 5 active S-boxes in Round 1, and 28 active S-boxes in Round 0. Changing the input difference to $0x0A000000000000000000000000000000$, we obtain the third best 9-round differential-linear distinguisher that we have found might be potentially used to break 12-round Serpent with a 256-bit key. The probability of $\beta \odot \gamma = 0$ is approximately 0.49903964996337891, and the distinguisher has a bias of approximately $2 \times 2^{-9.03} \times (2^{-27})^2 = 2^{-62.03}$. There are 4 active S-boxes in Round 1, and 26 active S-boxes in Round 0. If they were used to attack 12-round Serpent with a 256-bit key, the resulting attacks would require almost the entire codebook to have an acceptable success probability.

7 Possible extensions of our methodology

In this section we briefly discuss several possible extensions of our methodology, although particulars should be noticed.

The first possible extension is to consider the case when using two different values for the output mask δ in Definition 3, say δ_1, δ_2; that is, we might consider the event $\mathbb{E}(P) \odot \delta_1 = \mathbb{E}(P \oplus \alpha) \odot \delta_2$ for a randomly chosen $P \in \{0, 1\}^n$. The resulting differential-linear distinguisher would have a bias of $2(2\widehat{p} - 1)\epsilon_1\epsilon_2$ for some ϵ_1 and ϵ_2 denoting the respective bias of the two linear approximations. From a theoretical point of view, there seems no need to use two different output masks, for we can always choose the output mask with the bigger bias, and a key-recovery attack based on a differential-linear distinguisher with two different output masks requires us to guess no less key bits than that based on a differential-linear distinguisher with one output mask; however, the case with two different output masks may depend on Assumption 2 to a lesser degree than the discussed case with one output mask, for the two linear approximations can be independent somewhat, instead of two identical linear approximations used in the case with one output mask, and thus it may potentially be particularly helpful when making a practicable attack in reality.

The second possible extension is to consider the case when applying our methodology in a related-key [3,27,30] attack scenario. The notion of the related-key differential-linear analysis appeared in [26], and later Kim [29] described an enhanced version based on Langford and Biham et al.'s methodology. Likewise, we can get a more reasonable and general version based on our new methodology.

Other possible extensions are to obtain new methodologies, in a way similar to the above new methodology for differential-linear cryptanalysis, for the high-order differential-linear attack, the differential-bilinear attack and the differential-bilinear-boomerang attack, which were proposed in [13]. At present, however, these attack techniques appear to be hard to apply to obtain good cryptanalytic results in practice.

8 Conclusions

In this paper we have given a new methodology for differential-linear cryptanalysis under only the two assumptions implicitly used in the very first published paper on this technique. The new methodology is more reasonable and more general than Langford and Biham et al.'s methodology, and it can lead to some better differential-linear cryptanalytic results for some block ciphers than the previously known methodologies.

Using the new methodology, we have presented differential-linear attacks on 10-round CTC2 with a 255-bit block size and key, 13-round DES, and 12-round Serpent. In terms of the numbers of attacked rounds: The 10-round CTC2 attack is the first published cryptanalytic attack on the version of CTC2; the 13-round DES attack is inferior to the best previously published cryptanalytic results for DES; and the 12-round Serpent attack matches the best previously published cryptanalytic result for Serpent (that was obtained using Langford and Biham et al.'s methodology). In addition, an important merit for these new differential-linear cryptanalytic results is that they are obtained under only two assumptions and thus are more reasonable than those obtained using Langford and Biham et al.'s methodology. Like most cryptanalytic results on block ciphers, most of these attacks are far less than practical at present, but they provide a comprehensive understanding of the security of the block ciphers.

The new methodology as well as its possible extensions is a general cryptanalysis technique and can be potentially used to cryptanalyse other block ciphers; and block cipher designers should pay attention to this new methodology when designing ciphers.

The new methodology still requires Assumptions 1 and 2. As a direction for future research on differential-linear cryptanalysis, it would be interesting to investigate how to further reduce

the number of assumptions used, making a more reasonable and more general methodology that could be used in practice.

Acknowledgments The author is very grateful to Orr Dunkelman and Nathan Keller for their discussions on the flaw about CTC2 and to several anonymous referees for their comments on earlier versions of the paper, particularly the one who pointed out that the second methodology appeared earlier in Langford's PhD thesis and that the flaw on CTC2 might be because a wrong linear operation was used and thus the 3.5-round linear approximation from reference [22] might be flawed, too.

References

1. Anderson R., Biham E., Knudsen L.R.: Serpent: a new block cipher proposal. In: FSE 1998. Lecture Notes in Computer Science, vol. 1372, pp. 222–238. Springer, Heidelberg (1998).
2. Anderson R., Biham E., Knudsen L.R.: Serpent: a proposal for the Advanced Encryption Standard, NISTY AES Proposal (1998).
3. Biham E.: New types of cryptanalytic attacks using related keys. J. Cryptol. **7**(4), 229–246 (1994).
4. Biham E., Biryukov A.: An improvement of Davies' attack on DES. J. Cryptol. **10**(3), 195–206 (1997).
5. Biham E., Shamir A.: Differential cryptanalysis of DES-like cryptosystems. In: CRYPTO 1990. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer, Heidelberg (1990).
6. Biham E., Shamir A.: Differential cryptanalysis of DES-like cryptosystems. J. Cryptol. **4**(1), 3–72 (1991).
7. Biham E., Shamir A.: Differential cryptanalysis of the full 16-round DES. In: CRYPTO 1992. Lecture Notes in Computer Science, vol. 740, pp. 487–496. Springer, Heidelberg (1993).
8. Biham E., Dunkelman O., Keller N.: The rectangle attack—rectangling the Serpent. In: EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, pp. 340–357. Springer, Heidelberg (2001).
9. Biham E., Dunkelman O., Keller N.: Linear cryptanalysis of reduced round Serpent. In: FSE 2001. Lecture Notes in Computer Science, vol. 2355, pp. 16–27. Springer, Heidelberg (2002).
10. Biham E., Dunkelman O., Keller N.: Enhancing differential-linear cryptanalysis. In: ASIACRYPT 2002. Lecture Notes in Computer Science, vol. 2501, pp. 254–266. Springer, Heidelberg (2002).
11. Biham E., Dunkelman O., Keller N.: New results on boomerang and rectangle attacks. In: FSE 2002. Lecture Notes in Computer Science, vol. 2365, pp. 1–16. Springer, Heidelberg (2002).
12. Biham E., Dunkelman O., Keller N.: Differential-linear cryptanalysis of Serpent. In: FSE 2003. Lecture Notes in Computer Science, vol. 2887, pp. 9–21. Springer, Heidelberg (2003).
13. Biham E., Dunkelman O., Keller N.: New combined attacks on block ciphers. In: FSE 2005. Lecture Notes in Computer Science, vol. 3557, pp. 126–144. Springer, Heidelberg (2005).
14. Collard B., Standaert F.-X., Quisquater J.-J.: Improved and multiple linear cryptanalysis of reduced round Serpent. In: Inscrypt 2007. Lecture Notes in Computer Science, vol. 4990, pp. 51–65. Springer, Heidelberg (2008).
15. Collard B., Standaert F.-X., Quisquater J.-J.: Improved and multiple linear cryptanalysis of reduced round Serpent—description of the linear approximations. http://perso.uclouvain.be/fstandae/PUBLIS/47b.
16. Courtois N.T.: CTC2 and fast algebraic attacks on block ciphers revisited. IACR ePrint report 2007/152 (2007).
17. Courtois N.T., Pieprzyk J.: Cryptanalysis of block ciphers with overdefined systems of equations. In: ASIACRYPT 2002. Lecture Notes in Computer Science, vol. 2501, pp. 267–287. Springer, Heidelberg (2002).
18. Daemen J., Rijmen V.: AES proposal: Rijndael. In: Proceedings of the First Advanced Encryption Standard Candidate Conference, NIST, Ventura, CA (1998).
19. Davies, D.: Investigation of a potential weakness in the DES algorithm (1987) (unpublished manuscript).
20. Davies D., Murphy S.: Pairs and triplets of DES S-boxes. J. Cryptol. **8**(1), 1–25 (1995).
21. Dunkelman O.: Techniques for cryptanalysis of block ciphers. Ph.D. thesis, Technion-Israel Institute of Technology, Israel (2006).
22. Dunkelman O., Keller N.: Cryptanalysis of CTC2. In: CT-RSA 2009. Lecture Notes in Computer Science, vol. 5473, pp. 226–239. Springer, Heidelberg (2009).
23. Dunkelman O., Indesteege S., Keller N.: A differential-linear attack on 12-round Serpent. In: INDOCRYPT 2008. Lecture Notes in Computer Science, vol. 5365, pp. 308–321. Springer, Heidelberg (2008).
24. GNU Project, http://www.gnupg.org/oids.html.
25. Handschuh H., Naccache D.: SHACAL. In: Proceedings of the First Open NESSIE Workshop (2000).

26. Hawkes P.: Differential-linear weak key classes of IDEA. In: EUROCRYPT 1998. Lecture Notes in Computer Science, vol. 1403, pp. 112–126. Springer, Heidelberg (1998).
27. Kelsey J., Schneier B., Wagner D.: Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In: CRYPTO 1996. Lecture Notes in Computer Science, vol. 1109, pp. 237–251. Springer, Heidelberg (1996).
28. Kelsey J., Kohno T., Schneier B.: Amplified boomerang attacks against reduced-round MARS and Serpent. In: FSE 2000. Lecture Notes in Computer Science, vol. 1978, pp. 75–93. Springer, Heidelberg (2001).
29. Kim J.: Combined differential, linear and related-key attacks on block ciphers and MAC algorithms. Ph.D. thesis, Katholieke Universiteit Leuven, Belgium (2006).
30. Knudsen L.R.: Cryptanalysis of LOKI91. In: ASIACRYPT 1992. Lecture Notes in Computer Science, vol. 718, pp. 196–208. Springer, Heidelberg (1993).
31. Knudsen L.R.: Trucated and higher order differentials. In: FSE 1994. Lecture Notes in Computer Science, vol. 1008, pp. 196–211. Springer, Heidelberg (1995).
32. Knudsen L.R., Mathiassen J.E.: A chosen-plaintext linear attack on DES. In: FSE 2000. Lecture Notes in Computer Science, vol. 1978, pp. 262–272. Springer, Heidelberg (2001).
33. Kohno T., Kelsey J., Schneier B.: Preliminary cryptanalysis of reduced-round Serpent. In: Proceedings of the Third AES Candidate Conference (2000).
34. Kunz-Jacques S., Muller F.: New improvements of Davies-Murphy cryptanalysis. In: ASIACRYPT 2005. Lecture Notes in Computer Science, vol. 3788, pp. 425–442. Springer, Heidelberg (2005).
35. Lai X., Massey J.L., Murphy S.: Markov ciphers and differential cryptanalysis. In: EUROCRYPT 1991. Lecture Notes in Computer Science, vol. 547, pp. 17–38. Springer, Heidelberg (1991).
36. Langford S.K.: Differential-linear cryptanalysis and threshold signatures. Ph.D. thesis, Stanford University, USA (1995).
37. Langford S.K., Hellman M.E.: Differential-linear cryptanalysis. In: CRYPTO 1994. Lecture Notes in Computer Science, vol. 839, pp. 17–25. Springer, Heidelberg (1994).
38. Lu J.: Cryptanalysis of block ciphers. Ph.D. thesis, University of London, UK (2008).
39. Lu J.: New methodologies for differential-linear cryptanalysis and its extensions. Cryptology ePrint Archive, Report 2010/025 (2010). http://eprint.iacr.org/2010/025.
40. Lu J.: A methodology for differential-linear cryptanalysis and its applications (extended abstract). In: FSE 2012. Lecture Notes in Computer Science, vol. 7549, pp. 69–89. Springer, Heidelberg (2012).
41. Matsui M.: Linear cryptanalysis method for DES cipher. In: EUROCRYPT 1993. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer, Heidelberg (1994).
42. Matsui M.: The first experimental cryptanalysis of the Data Encryption Standard. In: CRYPTO 1994. Lecture Notes in Computer Science, vol. 839, pp. 1–11. Springer, Heidelberg (1994).
43. Matsui M., Yamagishi A.: A new method for known plaintext attack of FEAL cipher. In: EUROCRYPT 1992. Lecture Notes in Computer Science, vol. 658, pp. 81–91. Springer, Heidelberg (1993).
44. National Bureau of Standards (NBS), Data Encryption Standard (DES), FIPS-46 (1977).
45. National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), FIPS-197 (2001).
46. Selçuk A.A.: On probability of success in linear and differential cryptanalysis. J. Cryptol. **21**(1), 131–147 (2008).
47. Vaudenay S.: Provable security for block ciphers by decorrelation. In: STACS 1998. Lecture Notes in Computer Science, vol. 1373, pp. 249–275. Springer, Heidelberg (1998).
48. Wagner D.: The boomerang attack. In: FSE 1999. Lecture Notes in Computer Science, vol. 1636, pp. 156–170. Springer, Heidelberg (1999).
49. Wang X.Y., Hui L.C.K., Chow K.P., Chong C.F., Tsang W.W., Chan H.W.: The differential cryptanalysis of an AES finalist - Serpent. Technical report TR-2000-04, Department of Computer Science and Information Systems, The University of Hong Kong, China (2000). http://www.cs.hku.hk/research/techreps/document/TR-2000-04.