A Differential-Linear Attack on 12-Round Serpent

Orr Dunkelman[1,⋆], Sebastiaan Indesteege[2,⋆⋆], and Nathan Keller[3,⋆⋆⋆]

[1] École Normale Supérieure
Département d'Informatique,
CNRS, INRIA
45 rue d'Ulm, 75230 Paris, France.
orr.dunkelman@ens.fr
[2] Katholieke Universiteit Leuven
Department of Electrical Engineering ESAT/SCD-COSIC
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
sebastiaan.indesteege@esat.kuleuven.be
[3]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il

Abstract. Serpent is an SP Network block cipher submitted to the AES competition and chosen as one of its five finalists. The security of Serpent is widely acknowledged, especially as the best known attack so far is a differential-linear attack on only 11 rounds out of the 32 rounds of the cipher.
In this paper we introduce a more accurate analysis of the differential-linear attack on 11-round Serpent. The analysis involves both theoretical aspects as well as experimental results which suggest that previous attacks had overestimated complexities. Following our findings we are able to suggest an improved 11-round attack with a lower data complexity. Using the new results, we are able to devise the first known attack on 12-round Serpent.

1 Introduction

Serpent [1] is one of the five block ciphers chosen as AES finalists. The cipher has an SP Network structure repeating 32 rounds consisting of 4-bit to 4-bit S-boxes and a linear transformation. The block size is 128 bits, and the supported key size is of any length between 0 and 256 bits.

Since its introduction, Serpent was the target of extensive cryptanalytic efforts [5–7, 9, 13]. Despite that, the best previously known attack is on 11-round

⋆ The first author was supported by the France Telecome Chaire. Some of the work presented in this paper was done while the first author was staying at K.U. Leuven.
⋆⋆ F.W.O. Research Assistant, Fund for Scientific Research – Flanders (Belgium). Also supported by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy).
⋆⋆⋆ The research presented in this paper was supported by the Adams fellowship.

Serpent. In [13] a 256-bit key variant of 9-round Serpent is attacked using the amplified boomerang attack. The attack requires 2^{110} chosen plaintexts and its time complexity is 2^{252} 9-round Serpent encryptions.

In [5] the rectangle attack is applied to 256-bit key 10-round Serpent. The attack uses $2^{126.8}$ chosen plaintexts and has a time complexity of 2^{217} memory accesses.[1] The 10-round rectangle attack is improved in [7] and the improved attack requires $2^{126.3}$ chosen plaintexts with time complexity of $2^{173.8}$ memory accesses. A similar boomerang attack which requires almost the entire code book is also presented in [7].

In [6] a linear attack on 11-round Serpent is presented. The attack exploits a 9-round linear approximation with bias of 2^{-58}. The attack requires data complexity of 2^{118} known plaintexts and time complexity of 2^{214} memory accesses.

The linear approximation presented in [6] is combined with a differential in [9] to construct a differential-linear attack on 11-round Serpent. The data complexity of this attack is $2^{125.3}$ chosen plaintexts and the time complexity is about $2^{139.2}$ 11-round Serpent encryptions. The first attack on 10-round Serpent with 128-bit keys is also presented in [9]. The 10-round attack requires $2^{107.2}$ chosen plaintexts and $2^{125.2}$ 10-round Serpent encryptions.

We note that Serpent is also very common example in research about the use of multiple linear approximations in linear cryptanalysis [11, 12]. This line of research actually shows that the use of multiple linear approximations can give a great advantage from the data complexity point of view, but not necessarily from the time complexity point of view.

In this paper we present a more accurate analysis of the 11-round attack from [9], showing that the attack requires less data than previously believed (namely, $2^{121.8}$ chosen plaintexts). This leads to an immediate reduction in the time complexity of the attack (to $2^{135.7}$ encryptions). We then switch the order of the differential and the linear parts in the differential-linear approximation. The new 9-round differential-linear approximation is used to construct a new 11-round attack that uses $2^{113.7}$ chosen ciphertexts and has a running time of $2^{137.7}$ memory accesses.

The reduced data and time complexities allow to extend the 11-round attack from [9] by one extra round, and obtain the first 12-round attack on Serpent. This attack requires $2^{123.5}$ chosen plaintexts and has a time complexity of $2^{249.4}$ encryptions.

Finally, we present a novel related-key attack applicable to a modified variant of Serpent in which the round constants are removed from the key schedule algorithm. We note that, while the removal of these constants changes the cipher into a more symmetric structure, the repeated core, i.e., 8-round Serpent, is still relatively secure. The (still) non-trivial key schedule and the strong keyed permutation make most related-key attacks are quite likely to fail.

We organize this paper as follows: In Section 2 we present a short description of Serpent. Section 3 describes the differential-linear technique. We present the

[1] In [5] a different number is quoted, but in [7] this mistake is identified, and the correct time complexity of the algorithm is presented.

differential-linear attacks of this paper (the improved 11-round attack and the new 12-round attack) in Section 4. Section 5 describes a related-key attack on a modified variant of Serpent. We summarize our results and compare them with previous results on Serpent in Section 6. The appendices contain the differentials and the linear approximation used in the attacks.

2 A Description of Serpent

In [1] Anderson, Biham and Knudsen presented Serpent. Serpent has a block size of 128 bits and it accepts 0–256 bit keys. Serpent is an SP Network block cipher with 32 rounds. Each round is composed of key mixing, a layer of S-boxes and a linear transformation. There is an equivalent bitsliced description which is more efficient and easier to describe.

In our description we adopt the notations of [1] in the bitsliced version. The intermediate value before round i is denoted by $\hat{B}_i$ (a 128-bit value), where the 32 rounds are numbered $0, 1, \ldots, 31$. Each $\hat{B}_i$ is composed of four 32-bit words X_0, X_1, X_2, X_3.

Serpent uses a set of eight 4-bit to 4-bit S-boxes. Each round function R_i uses a single S-box applied 32 times in parallel. For example, R_0 uses 32 copies of S_0 in parallel. The first copy of S_0 takes the least significant bits from X_0, X_1, X_2, X_3 and returns the output to these bits. The set of eight S-boxes is used four times. S_0 is used in round 0, S_1 is used in round 1, etc. After using S_7 in round 7, S_0 is used again in round 8, then S_1 in round 9, and so on. In the last round (round 31) the linear transformation is omitted and another key is XORed.

The cipher may be formally described by the following equations:

$$\begin{aligned} \hat{B}_0 &:= P \\ \hat{B}_{i+1} &:= R_i(\hat{B}_i) \qquad i = 0, \ldots, 31 \\ C &:= \hat{B}_{32} \end{aligned}$$

where

$$\begin{aligned} R_i(X) &= LT(\hat{\mathcal{S}}_i(X \oplus \hat{K}_i)) \qquad \text{For } i = 0, \ldots, 30 \\ R_i(X) &= \hat{\mathcal{S}}_i(X \oplus \hat{K}_i) \oplus \hat{K}_{32} \qquad \text{For } i = 31 \end{aligned}$$

where $\hat{\mathcal{S}}_i$ is the application of the S-box $S_{(i \bmod 8)}$ thirty two times in parallel, and LT is the linear transformation of Serpent.

As our attack do not use explictly the properties of the linear transformation or the key schedule algorithm, we omit their description and refer the interested reader to [1].

3 Differential-Linear Cryptanalysis

Differential cryptanalysis [2] analyzes ciphers by studying the development of differences through the encryption process. A differential attack is mostly concerned with an input difference Ω_P for which an output difference Ω_T holds with

high enough probability (even though there are variants which use the fact that the probability is zero).

Linear cryptanalysis [16] analyzes the cipher by approximating the encryption process in a linear manner. The attacker finds a linear approximation $\lambda_P \cdot P \oplus \lambda_T \cdot T$ which holds with probability $1/2 + q$ (q might be negative) and gathers many plaintexts and ciphertexts. By checking whether the approximation holds, one can deduce subkey information or distinguish the cipher from a random permutation.

In 1994, Langford and Hellman [15] showed that both kinds of analysis can be combined together in a technique called *differential-linear cryptanalysis*. The attack uses a differential that induces a linear relation between two intermediate encryption values with probability one. In [8, 14] this technique is extended to the cases where the probability of the differential part is smaller than one.

We use notations based on [2, 4] for differential and linear cryptanalysis, respectively. In our notations Ω_P, Ω_T are the input and output differences of the differential characteristic, and λ_T, λ_C are the input and output subsets (denoted by bit masks) of the linear approximation.

Let E be a block cipher described as a cascade of two sub-ciphers E_0 and E_1, i.e., $E = E_1 \circ E_0$. Langford and Hellman suggested to use a truncated differential $\Omega_P \rightarrow \Omega_T$ for E_0 with probability 1. To this differential they concatenate a linear approximation $\lambda_T \rightarrow \lambda_C$ for E_1 with probability $1/2+q$ (or bias q). Their attack requires that the bits masked in λ_T have a zero difference in Ω_T.

If we take a pair of plaintexts P_1 and P_2 that satisfy $P_1 \oplus P_2 = \Omega_P$, then after E_0, $\lambda_T \cdot E_0(P_1) = \lambda_T \cdot E_0(P_2)$. This follows from the fact that $E_0(P_1)$ and $E_0(P_2)$ have a zero difference in the masked bits according to the output difference of the differential.

Recall that the linear approximation predicts that $\lambda_T \cdot T = \lambda_C \cdot E_1(T)$ with probability $1/2+q$. Hence, $\lambda_T \cdot E_0(P_1) = \lambda_C \cdot E_1(E_0(P_1))$ with probability $1/2+q$, and $\lambda_T \cdot E_0(P_2) = \lambda_C \cdot E_1(E_0(P_2))$ with probability $1/2 + q$. As the differential predicts that $\lambda_T \cdot E_0(P_1) = \lambda_T \cdot E_0(P_2)$, then with probability $1/2 + 2q^2$, $\lambda_C \cdot C_1 = \lambda_C \cdot C_2$ where C_1 and C_2 are the ciphertexts corresponding to P_1 and P_2, respectively, i.e., $C_i = E_1(E_0(P_i))$.

This fact allows to construct differential-linear distinguishers based on encrypting many plaintext pairs and checking whether the ciphertexts agree on the parity of the output subset. The data complexity of the distinguishers is $O(q^{-4})$ chosen plaintexts. The exact number of plaintexts is a function of the desired success rate, and of the number of possible subkeys.

In [8] Biham, Dunkelman and Keller proposed a way to deal with differentials with probability $p < 1$. In case the differential is satisfied (probability p), the above analysis remains valid. The assumption for the remaining $1 - p$ of the pairs is that the input subset parities are distributed randomly. In that case, the probability that a pair with input difference Ω_P will satisfy $\lambda_C \cdot C_1 = \lambda_C \cdot C_2$ is $p(1/2 + 2q^2) + (1 - p) \cdot 1/2 = 1/2 + 2pq^2$.

Furthermore, in [8] it is shown that the attack can still be applicable if $\Omega_T \cdot \lambda_T = 1$, i.e., the differential predicts that there is a difference in approximated

bits. In this case, the analysis remains valid, but instead of looking for the instances for which $\lambda_T \cdot C_1 = \lambda_T \cdot C_2$, we look for the cases when $\lambda_T \cdot C_1 \neq \lambda_T \cdot C_2$. As the analysis remains the same given a pair of plaintexts with the input difference Ω_P, the probability that the pair disagrees on the output subset parity is $1/2 + 2pq^2$. Another interesting result is that the attack still applies even when $\Omega_T \cdot \lambda_T$ is unknown, as long as its value is fixed. The data complexity of the enhanced differential-linear attack is $O(p^{-2}q^{-4})$.

4 Differential-Linear Attacks on Serpent

We first recall the 11-round attack from [9] which we use as a starting point of our research. We then continue to improve the 11-round attack by reducing the data complexity by a factor of about 2^8. Our main results follow from a small change in the linear approximation, which takes into account the huge difference between the number of active S-boxes and the number of pairs. Finally, we extend the 11-round attack to 12 rounds.

4.1 The Previous Attack on 11-Round Serpent

The attack from [9] is a differential-linear attack using a 9-round differential-linear approximation for rounds 2–10 composed of a 3-round differential and a 6-round linear approximation. The input difference of the 3-round differential is Ω_P = 0000 0000 0000 0000 0000 0000 4005 0000_x which with probability 2^{-6} does not affect bits 1 and 117 at the entrance of round 5. The 6-round linear approximation starts with these bits, and the output mask is λ_C = 0000 1000 0000 0000 5000 0100 0010 0001_x. The bias of the approximation is 2^{-27}, and thus the total bias of the differential-linear approximation is $2pq^2 = 2 \cdot 2^{-6} \cdot (2^{-27})^2 = 2^{-59}$. We describe the differential and the approximation in Appendices A and B, respectively.

There are 5 active S-boxes in the round before the differential-linear approximation and 7 active S-boxes afterward. Thus, the attacker uses structures of 2^{20} chosen plaintexts each (covering the five active S-boxes), thus resulting in 2^{39} pairs (of which 2^{19} are expected to have difference Ω_P at the entrance to round 1). After generating sufficiently many such structures, the attacker uses the following algorithm: For each guess of the subkey in round 0 that enters the 5 active S-boxes, the attacker partially encrypts all the plaintexts, and finds all the plaintext pairs with input difference Ω_P. Then, for each of these pairs, and for each guess of the subkey in round 10, the attacker checks whether the partial decryption of the pair satisfies the approximation or not.

The last step is done in an optimized way using a table look-up. We note that for each pair only 7 S-boxes are decrypted. Thus, there are 28 bits from each of the two ciphertexts that are being decrypted (under a subkey guess of 28 bits). Thus, instead of repeatedly decrypting the same values under the same subkey guess, the attacker counts for each subkey guess of round 1 how many times each of the 56-bit ciphertext values (the 28 bits from each of the

two paired ciphertexts) appears. Then, by performing 2^{28} trial encryptions for each of these counters, the attacker is able to deduce how many pairs satisfy the approximation.

In [9] the above attack is applied using $2^{125.3}$ plaintexts (which compose $2^{124.3}$ pairs). The time complexity is mostly dominated by the division into pairs, i.e., the partial encryption of $2^{125.3}$ values under 2^{20} possible subkeys. Repeating the analysis done in [9] and using the success probability formula establishedin [17] we have found out that for $2^{122.3}$ pairs, the success probability of the attack is expected to be about 84%. Thus, the actual data and time complexity of the original 11-round attack is $2^{123.3}$ chosen plaintexts, and the time complexity is $2^{137.2}$ encryptions.

We have experimentally verified the differential-linear property with a 3-round differential and the first round of the linear approximation. While the expected bias for this shortened approximation is $2 \cdot 2^{-6}(\cdot 2^{-5})^2 = 2^{-15}$, we found out that the bias of the shortened differential-linear approximation is $2^{-13.75}$. We performed 100 experiments, where in each experiment 2^{36} pairs with input difference Ω_P were encrypted, and the intermediate encryption values were checked with respect to whether the parity of the output subset is the same or not. The standard deviation of the bias was $2^{-18.87}$.

The difference between the expected value and the actual value follows the fact that even when the differential is not satisfied, and a difference enters one of the approximated S-boxes, the output mask is still biased. This means that the assumption that for pairs which do not follow the differential, there is no bias from 1/2 with respect to whether the approximations hold simultaneously for the two intermediate values, does not hold.

By assuming the piling-up lemma [16] to hold for the remainder of the linear approximation, we expect that the actual bias of the 9-round differential-linear approximation is also $2^{1.25}$ times higher than 2^{-59}, i.e., the probability that two pairs with input difference Ω_P have the same parity in λ_C is $1/2+2^{-57.75}$. Taking this into account shows that the actual data complexity required for the original 11-round attack is $2^{121.8}$ chosen plaintexts, and that the actual time complexity is $2^{135.7}$ encryptions.

4.2 Further Improvements of the 11-Round Attack

We first note that the attack can be easily improved by using the optimization ideas performed in the original attack also in the differential side of the distinguishers ,i.e., in round 1. For each subkey guess, we can build a list of the pairs according to the value in the 20 bits which enter the five active S-boxes. Thus, let P_1 and P_2 be a pair under some subkey guess, and flip a bit which does not enter an active S-box in both plaintexts to obtain P_1' and P_2', respectively. It is obvious that P_1' and P_2' are actually a pair, without any need to partially encrypt them. Thus, it is possible to improve the attack from [9] to be $2^{20} \cdot 2^{121.8} = 2^{141.8}$ memory accesses rather than $2^{135.7}$ 11-round encryptions.

The second improvement is based on the observation that the attacker has to process $2^{121.8}$ plaintexts/ciphertexts, and thus, the time complexity of the

partial decryption at round 11 (which is about $2^{28} \cdot 2^{28}$ partial decryptions and 2^{84} memory accesses for a subkey guess of round 1) can be increased without affecting the time complexity of the attack. This can be achieved by inverting the order of the differential and the linear approximation. If we use a 3-round differential for rounds 11–13 (with probability 2^{-6}) and the linear approximation for rounds 5–10 as before (but in the decryption direction), then the linear approximation can be improved (increasing its bias by a factor of 2), thus reducing the data complexity of the attack, and as a consequence the time complexity as well. The change in the linear approximation is changing one of the approximations in round 5 to activate more S-boxes in the round before in exchange for a higher bias. The new 3-round differential for rounds 11–13 is presented in Appendix A.

We experimentally verified that the bias in the number of pairs with ciphertext difference Ω_C having the same parity in the input of the differential is 2^{-7}. When we decrypted one more round, and applied the last round of the linear approximation we expected a bias of $2pq^2 = 2 \cdot 2^{-6} \cdot (2^{-6})^2 = 2^{-17}$. However, for 100 different keys, we have observed a bias of 2^{-14} (we used 2^{36} pairs in each experiment, and the mean value was $2^{-13.93}$ with standard deviation of $2^{-18.92}$). Assuming that the remaining rounds behave independently, the expected bias of the entire 9-round differential-linear approximation in the inverse direction is 2^{-54}.

The difference between the expected and the computed values follows from the correlation between the differential and the linear approximation. It appears[2] that in about half of the pairs satisfying the differential, the input difference in at least one of the five active S-boxes in the first round of the linear approximation is zero. As a result, the bias of the differential-linear approximation for these pairs is much higher, and this causes the higher bias of the overall differential-linear approximation.

Thus, the improved 11-round attack is as follows:

1. Select $N = 2^{113.7}$ ciphertexts, consisting of $2^{89.7}$ structures, each is chosen by selecting:
   (a) A ciphertext C_0.
   (b) The ciphertexts $C_1,\ldots,C_{2^{24}-1}$ which differ from C_0 by all the $2^{24}-1$ possible (non-empty) subsets of the twenty four bits which enter the 6 active S-boxes in round 14.
2. Decrypt these ciphertexts under the unknown key K.
3. For each value of the 24 bits of K_{14} entering these 6 S-boxes:
   (a) Initialize an array of 2^{72} counters to zeroes.
   (b) Partially decrypt for each ciphertext the 6 active S-boxes in round 14 and find the pairs which satisfy the difference Ω_C after round 13.
   (c) Given those $2^{112.7}$ pairs, perform for each ciphertext pair: Let $extract_{36}$ be the function that extracts the 36 bits which enter the 9 active S-boxes in round 4, then for each pair P, P' increment the counter corresponding to $extract_{36}(P)||extract_{36}(P')$.

[2] We have verified this claim experimentally.

(d) For every 36-bit guess of the subkey entering these S-boxes, compute the parity of the corresponding partial decrypted pairs, and store the most biased guess for these 36 subkey bits (along with the guess of K_{14}).

4. Output the subkey combination with the largest deviation from $N/2$.

The data complexity of the attack is $2^{113.7}$ chosen plaintexts. The time complexity of the attack is dominated mainly by Step 3 which is repeated 2^{24} times. For each of these guesses the attacker first identifies the pairs (using tables) and has to perform $2^{113.7}$ memory accesses to compute $extract_{36}$ for all the pairs (Step 3(c)) and about 2^{108} memory accesses in Step 3(d), which means that the total time complexity of the attack is $2^{137.7}$ memory accesses. Again, using the success formula found in [17], for $2^{113.7}$ chosen plaintexts, the probability that the right key has the largest bias is about 93%.

The memory complexity of the attack is $2^{24} \cdot 2^{72} = 2^{96}$ counters. As the attack is repeated 2^{24} times (once for each guess of K_{14}), we can either store all the data, i.e., $2^{113.7}$ values, or store for each such guess the number of pairs. The second way is more efficient, as it allows analyzing each structure independently of others, and discarding it once the analysis is done. This approach has no impact on the data complexity or the time complexity, but it reduces the memory complexity to 2^{96} counters, each of up to 64 bits, or a total of 2^{99} bytes.

4.3 12-Round Differential-Linear Attack

We now present a differential-linear attack on 12-round Serpent. The attack is based on the original 11-round attack (in the forward direction) and uses the fact that a pair which satisfies the input difference of the differential has at most 28 active S-boxes in round 0. Thus, it is possible to change the attack algorithm a bit and obtain a 12-round attack against Serpent with 256-bit keys.

We have tried all the possible input differences to round 1 that lead to the difference $LT^{-1}(\Omega_P) = 2000\ 0000\ 0000\ 01A0\ \ 0E00\ 4000\ 0000\ 0000_x$. This difference is not affected by S-boxes 2, 3, 19, and 23, i.e., these S-boxes do not affect the active bits of $LT^{-1}(\Omega_P)$. Thus, we can construct structures of plaintexts which take this fact into consideration and obtain a 12-round attack on Serpent:

1. Select $N = 2^{123.5}$ plaintexts, consisting of $2^{11.5}$ structures, each is chosen by selecting:
   (a) Any plaintext P_0.
   (b) The plaintexts $P_1,\ldots,P_{2^{112}-1}$ which differ from P_0 by all the $2^{112} - 1$ possible (non-empty) subsets of the bits which enter all S-boxes besides 2, 3, 19, and 23 in round 0.
2. Request the ciphertexts of these plaintext structures (encrypted under the unknown key K).
3. For each value of the 112 bits of K_0 entering these 28 S-boxes, partially encrypt all the plaintexts the first round, and apply the original 11-round attack.

4. Each trial of the key gives us $112 + 20 + 28 = 160$ bits of the subkeys (112 bits in round 0, 20 bits in round 1 and 28 bits in round 11), along with a measure for correctness. The correct value of the 160 bits is expected to be the most frequently suggested value (with more than 84% success rate).
5. The rest of the key bits are then recovered by auxiliary techniques.

The data complexity of the attack is $2^{123.5}$ chosen plaintexts. The time complexity of the attack is $2^{123.5} \cdot 2^{112} \cdot \frac{28}{384} = 2^{231.7}$ encryptions for the partial encryption in Step 3, and $2^{112} \cdot 2^{137.4} = 2^{249.4}$ for the repeated trials of the 11-round attack.[3]

4.4 10-Round Differential-Linear Attack on Serpent with 128-bit Keys

We can use the three improvements suggested earlier to improve the 10-round attack on Serpent. We recall the three improvements:

- Better analysis of the bias of the differential-linear approximation,
- Better analysis of the success probability,
- Changing the output mask.

We shall start with changing the output mask of the approximation. In the 10-round attack in [9], the last round of the approximation is omitted, and the new 8-round differential-linear approximation has a bias of $2 \cdot 2^{-6} \cdot (2^{-22})^2 = 2^{-49}$. The last round of the approximation is optimized for reducing the number of active S-boxes in the last round (to 5 S-boxes). However, as before, we may activate a few more S-boxes, and almost have no effect on the time complexity of the attack (by increasing the counters).

By changing the output mask of the last round (where S_1 is used) from $\lambda'_C = 0010\ 0001\ 0000\ 1000\ \ 0100\ 0000\ 0000\ 0000$ to $\lambda'_C = 0010\ 0001\ 0000\ 1000\ \ 0300\ 0000\ 0000\ 0000$, we increase the bias of the linear approximation by a factor of 2, i.e., the differential-linear approximation has a bias of 2^{-47}.

Taking into consideration the better transition between the differential and the linear approximation, we obtain that the actual bias is $2^{-45.75}$. Using the formula from [17], and taking into consideration that there are 5 active S-boxes before the differential, and 9 active S-boxes after the linear approximation, we need $2^{96.2}$ pairs, i.e., $2^{97.2}$ chosen plaintexts to achieve a success rate of 84%.

The time complexity of the attack is $2^{111.2}$ 10-round encryptions (the time complexity required for partial encryptions and locating all the pairs) and 2^{128} memory accesses for handling the tables.

We note that if the approximation is not changed, the data complexity of the 10-round attack is $2^{101.2}$ chosen plaintexts, and the time complexity is $2^{115.2}$ encryptions.

[3] We note that the time complexity of the 11-round attack is $2^{135.7}$ encryptions. As the number of plaintexts is $2^{1.7}$ times larger in this attack, the time complexity of one iteration of the 11-round attack in this case is $2^{1.7}$ times larger.

5 A Related-Key Attack on a Modified Serpent

It is a well known fact that ciphers that iterate the exact same round function over and over are susceptible to slide attacks and related-key attacks [3, 10]. In Serpent the constants which modify the round function are found in two places: the different S-boxes (which are used in a cycle of 8 rounds), and the constants in the key schedule algorithm.

Removing the constants from the key schedule algorithm makes the cipher susceptible to related-key attacks which treat the cipher as an iteration of the same "round" function which is composed of 8 consecutive rounds. Even though there are several attacks on 8-round Serpent, it is highly unlikely to elevate them into attacks on the full Serpent, as 8-round Serpent is secure enough to prevent easy detection of the related-key plaintext pairs.

We present a related-key relation that holds with probability of 2^{-124}, and can be used to distinguish this simplified variant of Serpent from a random permutation (for 256-bit keys) with data complexity of about 2^{125} chosen plaintexts, and a negligible time complexity. We then use this relation to retrieve partial information about the keys.

Consider two related keys K and K' such that $K = (w_{-8}, \ldots, w_{-1})$ and $K' = (w_{-8} \lll 1, \ldots, w_{-1} \lll 1)$. For these two keys, all the corresponding subkeys k_i and k'_i respectively satisfy that $k_i = k'_i \lll 1$. Under these two keys we consider the plaintexts $P = (a, b, c, d)$ and $P' = (a \lll 1, b \lll 1, c \lll 1, d \lll 1)$. We denote such keys, plaintexts, or intermediate encryption values, i.e., two values such that the second is a rotate to the left by one bit of each 32-bit word independently, as satisfying the *rotation property*.

The rotation property is kept through the key addition, i.e., $P' \oplus K'_1$ is a rotate left by one bit word-wise of $P \oplus K$, and the S-boxes layer. The only problem is the linear transformation which contains cyclic rotations and shifts. The cyclic rotations do not affect the rotation property, so the only problem in extending the property is the shift operation. However, the property can bypass a shift with probability of 2^{-2}. Let X be a 32-bit word, and let $X' = X \lll 1$. Then, if the least significant bit of X is zero, and the least significant bit of $X' \ll m$ (the most significant bit of $X \ll m$) is 0 as well, then $X \ll m$ and $X' \ll m$ satisfy the rotation property. As in each linear transformation there are two such shifts, the probability that the rotation property is maintained after the linear transformation is 2^{-4}.

Serpent has 31 linear transformations, and thus, the probability that the rotation property remains from the plaintext till the ciphertext is 2^{-124}, while for two random permutations, one expects the probability of 2^{-128}. This property can be used to distinguish this variant of Serpent from a random permutation using about 2^{125} plaintexts. Given the pair that satisfies the rotation property it is also possible to deduce the equivalent of 4 bits of the key.

6 Summary

In this paper we studied differential-linear cryptanalysis of Serpent. We showed several improvements in the analysis of the previously best known results on 11-round Serpent, and suggested a new 11-round attack with a much lower data complexity. Combining experimental results and the improved analysis, we presented the first attack on 12-round Serpent. The attack uses $2^{123.5}$ chosen plaintexts, and has a time complexity of $2^{249.4}$ encryptions.

Finally, we explored a related-key attack on a modified Serpent where the round constants are removed from the key schedule, and showed that despite the strong repeated cipher (8-round of Serpent), there are high probability related-key properties that can be used both for distinguishing and key recovery.

We summarize our new attacks, and selected previously published attacks against Serpent in Table 1.

Rounds	Type of Attack	Key Size	Complexity Data	Time	Memory
10	Rectangle [7]	192 & 256	$2^{126.3}$ CP	$2^{173.8}$ MA	$2^{131.8}$ B
	Boomerang [7]	192 & 256	$2^{126.3}$ AC	$2^{173.8}$ MA	2^{89} B
	Differential-Linear [9]	all	$2^{105.2}$ CP	$2^{123.2}$ En	2^{40} B
	Differential-Linear (Sect. 4.4)	all	$2^{101.2}$ CP	$2^{115.2}$ En	2^{40} B
	Differential-Linear (Sect. 4.4)	all	$2^{97.2}$ CP	2^{128} MA	2^{72} B
11	Differential-Linear [9]	192 & 256	$2^{125.3}$CP	$2^{172.4}$ En	2^{30} B
	Differential-Linear [9]	192 & 256	$2^{125.3}$ CP	$2^{139.2}$ En	2^{60} B
	Differential-Linear (Sect. 4.1)	192 & 256	$2^{121.8}$ CP	$2^{135.7}$ MA	2^{76} B
	Differential-Linear (Sect. 4.2)	192 & 256	$2^{113.7}$ CC	$2^{137.7}$ MA	2^{99} B
12	Differential-Linear (Sect. 4.3)	256	$2^{123.5}$ CP	$2^{249.4}$ En	$2^{128.5}$ B

En — Encryptions, MA — Memory Accesses, B — bytes, CP — Chosen Plaintexts
CC — Chosen Ciphertexts, AC — Adaptive Chosen Plaintexts and Ciphertexts

Table 1. Summary of Attacks on Serpent with Reduced Number of Rounds

References

1. Ross Anderson, Eli Biham, Lars R. Knudsen, *Serpent: A Proposal for the Advanced Encryption Standard*, NIST AES Proposal, 1998.
2. Eli Biham, A Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer, 1993.
3. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, Springer, 1994.
4. Eli Biham, *On Matsui's Linear Cryptanalysis*, Advances in Cryptology, proceeding of EUROCRYPT 1994, Lecture Notes in Computer Science 950, pp. 341–355, Springer, 1994.

5. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, proceeding of EUROCRYPT 2001, Lecture Notes in Computer Science 2045, pp. 340–357, Springer, 2001.
6. Eli Biham, Orr Dunkelman, Nathan Keller, *Linear Cryptanalysis of Reduced Round Serpent*, proceedings of Fast Software Encryption 8, Lecture Notes in Computer Science 2355, pp. 16–27, Springer, 2002.
7. Eli Biham, Orr Dunkelman, Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceeding of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 1–16, Springer, 2002.
8. Eli Biham, Orr Dunkelman, Nathan Keller, *Enhancing Differential-Linear Cryptanalysis*, Advances in Cryptology, proceeding of ASIACRYPT 2002, Lecture Notes in Computer Science 2501, pp. 254–266, Springer, 2002.
9. Eli Biham, Orr Dunkelman, Nathan Keller, *Differential-Linear Cryptanalysis of Serpent*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 9–21, Springer, 2003.
10. Alex Biryukov, David Wagner, *Slide Attacks*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 245–259, Springer, 1999.
11. Baudoin Collard, François-Xavier Standaert, Jean-Jacques Quisquater, *Improved and Multiple Linear Cryptanalysis of Reduced Round Serpent*, proceedings of Inscrypt 2007, Lecture Notes in Computer Science 4817, pp. 77–88, Springer, 2007.
12. Baudoin Collard, François-Xavier Standaert, Jean-Jacques Quisquater, *Experiments on the Multiple Linear Cryptanalysis of Reduced Round Serpent*, proceedings of Fast Software Encryption 15, in Lecture Notes in Computer Science 5086, pp. 382–397, Springer, 2008.
13. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 75–93, Springer, 2001.
14. Susan K. Langford, *Differential-Linear Cryptanalysis and Threshold Signatures*, Ph.D. thesis, 1995.
15. Suzan K. Langford, Martin E. Hellman, *Differential-Linear Cryptanalysis*, Advances in Cryptology, proceedings of CRYPTO '94, Lecture Notes in Computer Science 839, pp. 17–25, Springer, 1994.
16. Mitsuru Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 386–397, Springer, 1994.
17. Ali Aydin Selçuk, *On Probability of Success in Linear and Differential Cryptanalysis*, Journal of Cryptology, vol. 21, no. 1, pp. 131–147, Springer, 2008.
18. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, Springer, 1999.

A The Differential Characteristic

A.1 The Original 3-Round Differential

The 3-round truncated differential used in the original 11-round attack is as follows. The first round of the differential is round 2 (or any other round that

uses S_2) with probability 2^{-5}:

$$\begin{array}{rll} \Omega_P = & 0000\ 0000\ 0000\ 0000\ \ 0000\ 0000\ 4005\ 0000 & \stackrel{S_2}{\rightarrow} \quad \Pr = 2^{-5} \\ & 0000\ 0000\ 0000\ 0000\ \ 0000\ 0000\ A004\ 0000 & \stackrel{LT}{\rightarrow} \\ & 0040\ 0000\ 0000\ 0000\ \ 0000\ 0000\ 0000\ 0000 & \stackrel{S_3}{\rightarrow} \quad \Pr = 2^{-1} \\ & 00X0\ 0000\ 0000\ 0000\ \ 0000\ 0000\ 0000\ 0000 & \end{array}$$

where $X \in \{2, 4, 6, 8, A_x, C_x, E_x\}$. After the linear transformation, we get the following truncated differential in S_4:

$$\begin{array}{rl} 0QT_30\ 0T_200\ 0T_100\ 0000\ \ 000Y_4\ 00Y_30\ W_2Y_20W_1\ Y_10Z0 & \stackrel{S_4}{\rightarrow} \\ 0??0\ 0?00\ 0?00\ 0000\ \ 000?\ 00?0\ ??0?\ ?0?0 & = \Omega_T, \end{array}$$

where ? is any possible difference and $Y_i \in \{0, 1\}, Z \in \{0, 2\}, W_i \in \{0, 8\}, T_i \in \{0, 4\}, Q \in \{0, 2, 4, 6\}$.

A.2 The 3-Round Differential in the Improved 11-Round Attack

The 3-round differential used in the improved 11-round attack is in the backward direction. The output difference is $\Omega_C = 0000\ 0000\ 0000\ 0090\ 0000\ 0000\ 0000\ 0000_x$ which with probability of about 2^{-6} does not affect the bits in $LT(\lambda_C)$. This follows from the main following differential characteristic:

$$\begin{array}{rll} \Omega_C = & 0000\ 0000\ 0000\ 0090\ \ 0000\ 0000\ 0000\ 0000 & \stackrel{S_5^{-1}}{\rightarrow} \quad \Pr = 2^{-2} \\ & 0000\ 0000\ 0000\ 0040\ \ 0000\ 0000\ 0000\ 0000 & \stackrel{LT^{-1}}{\rightarrow} \\ & 0000\ A004\ 0000\ 0000\ \ 0000\ 0000\ 0000\ 0000 & \stackrel{S_4^{-1}}{\rightarrow} \quad \Pr = 2^{-3} \\ & 0000\ ?009\ 0000\ 0000\ \ 0000\ 0000\ 0000\ 0000 & \stackrel{LT^{-1}}{\rightarrow} \\ & 0Z_300\ T_2YZ_2R\ 0T_14Z_1\ 2080\ \ 0X_200\ 10X_10\ 01Q0\ 0W00 & \stackrel{S_3^{-1}}{\rightarrow} \quad \Pr = 1 \\ & 0?00\ ????\ 0???\ ?0?0\ \ 0?00\ ?0?0\ 0??0\ 0?00 & = \Omega_T \end{array}$$

with probability 1, when $W \in \{0, 4\}, Q \in \{0, 2, 8, A_x\}, X_i \in \{0, 1\}, Z_i \in \{0, 8\}, T_i \in \{8, A_x\}, R \in \{8, C_x\}$.

We note that despite the fact that the probability of this differential is 2^{-5}, when counting all possible output differences, the probability that there is a difference in the bits covered by $LT(\lambda_C)$, the bias was found to be 0.007773, i.e., $1/128.7 \approx 2^{-7.007}$. This follows mostly from the cases where the differential does not follow the second round, i.e., the input difference to S-box 24 is not 9, as then there is an active S-box which affects the approximation with relatively high probability (with output difference 2). Thus, the "probability" of the differential can be assumed to be $p = 2^{-6}$.

B The Linear Approximation

The 6-round linear approximation used in the attack is as follows. It starts before the linear transformation of round 4 with $\lambda_T = 2006\ 0040\ 0000\ 0100\ \ 1000\ 0000$

0000 0000_x. In round 5 the following approximation[4] holds with bias -2^{-5}:

$$
\begin{array}{rlll}
LT(\lambda_T) = & 0020\ 0000\ 0000\ 0000 \ \ 0000\ 0000\ 0000\ 0002 & \xrightarrow{S_5} & \Pr = \frac{1}{2} - 2^{-5} \\
& 0040\ 0000\ 0000\ 0000 \ \ 0000\ 0000\ 0000\ 0008 & \xrightarrow{LT} & \\
& 0000\ 0000\ 0000\ 0000 \ \ 0000\ 0000\ 8000\ 0000 & \xrightarrow{S_6} & \Pr = \frac{1}{2} - 2^{-3} \\
& 0000\ 0000\ 0000\ 0000 \ \ 0000\ 0000\ 1000\ 0000 & \xrightarrow{LT} & \\
& 0000\ 00A0\ 0001\ 0000 \ \ 0000\ 0000\ 0000\ 0000 & \xrightarrow{S_7} & \Pr = \frac{1}{2} - 2^{-5} \\
& 0000\ 0010\ 0001\ 0000 \ \ 0000\ 0000\ 0000\ 0000 & \xrightarrow{LT} & \\
& 0000\ 0000\ 0000\ 0000 \ \ 0000\ 1000\ 0B00\ 00A0 & \xrightarrow{S_0} & \Pr = \frac{1}{2} + 2^{-6} \\
& 0000\ 0000\ 0000\ 0000 \ \ 0000\ 1000\ 0100\ 0010 & \xrightarrow{LT} & \\
& 0010\ 000B\ 0000\ B000 \ \ 0A00\ 0000\ 0000\ 0000 & \xrightarrow{S_1} & \Pr = \frac{1}{2} - 2^{-7} \\
& 0010\ 0001\ 0000\ 1000 \ \ 0100\ 0000\ 0000\ 0000 & \xrightarrow{LT} & \\
& 0000\ A000\ 0000\ 0000 \ \ 1000\ 0B00\ 00B0\ 000B & \xrightarrow{S_2} & \Pr = \frac{1}{2} - 2^{-6} \\
& 0000\ 1000\ 0000\ 0000 \ \ 5000\ 0100\ 0010\ 0001 & = \lambda_C. &
\end{array}
$$

After the linear transformation of round 11, $LT(\lambda_C) = 000B\ 0000\ B000\ 0300\ \ 00B0\ 200E\ 0000\ 0010$, i.e., there are seven active S-boxes: 1, 8, 11, 13, 18, 23 and 28.

[4] For the improved attack we change the input bias in S-box 29 to E_x and the bias in that case is 2^{-4}.