The Rectangle Attack – Rectangling the Serpent*

Eli Biham† Orr Dunkelman‡ Nathan Keller§

Abstract

Serpent is one of the 5 AES finalists. The best attack published so far analyzes up to 9 rounds. In this paper we present attacks on 7-round, 8-round, and 10-round variants of Serpent. We attack a 7-round variant with all key lengths, and 8- and 10-round variants with 256-bit keys. The 10-round attack on the 256-bit keys variants is the best published attack on the cipher. The attack enhances the amplified boomerang attack and uses better differentials. We also present the best 3-round, 4-round, 5-round and 6-round differential characteristics of Serpent.

1 Introduction

Serpent [1] is a block cipher which was suggested as a candidate for the Advanced Encryption Standard (AES) [8], and was selected to be among the five finalists.

In [4] a modified variant of Serpent in which the linear transformation was modified into a permutation was analyzed. The permutation allows one active S box to activate only one S box in the consecutive round, a property that cannot occur in Serpent. Thus, it is not surprising that this variant is much weaker than Serpent, and that it can be attacked with up to 35 rounds.

In [6] the 256-bit variant of Serpent up to 9 rounds is attacked using an amplified boomerang attack. The attack is based on building a 7-round distinguisher for Serpent, and using it for attacking up to 9 rounds. The distinguisher is built using the amplified boomerang technique. It uses a 4-round differential characteristic in rounds 1–4, and a 3-round characteristic in rounds 5–7.

*The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-1999-12324 and by the fund for the promotion of research at the Technion.

†Computer Science department, Technion – Israel Institute of Technology, Haifa 32000, Israel, biham@cs.technion.ac.il, http://www.cs.technion.ac.il/~biham/.

‡Computer Science department, Technion – Israel Institute of Technology, Haifa 32000, Israel, orrd@cs.technion.ac.il, http://vipe.technion.ac.il/~orrd/me/.

§Mathematics department, Technion – Israel Institute of Technology, Haifa 32000, Israel, nkeller@tx.technion.ac.il.

In this paper we enhance the amplified boomerang attack, and present the best 3-round, 4-round, 5-round and 6-round differential characteristics of Serpent published so far. We use these characteristic to devise an attack on 7-round Serpent with all key lengths, and an attack on 8-round Serpent with 256-bit keys. We also use these results to develop the best known distinguisher for 8-round Serpent by presenting a new cryptanalytic tool — the rectangle attack. This tool is then used to attack 10-round 256-bit key Serpent.

The paper is organized as follows: In Section 2 we give the description of Serpent. In Section 3 we present a differential attack on 7-round Serpent, and a differential attack on 8-round 256-bit key Serpent. In Section 4 we present the *Rectangle Attack*, and in Section 5 we describe the 8-round distinguisher and implement the attack on 10-round 256-bit key Serpent. Section 6 summarizes the paper. In the appendices we describe new 3-round, 4-round, 5-round and 6-round differential characteristics, which are the best known so far.

2 A Description of Serpent

Serpent [1] is a block cipher with block size of 128 bits and 0–256 bit keys. It is an SP-network, consisting of alternating layers of key mixing, S boxes and linear transformation. Serpent has an equivalent bitsliced description, which makes it very efficient.

The key scheduling algorithm of serpent accepts 256-bit keys. Shorter keys are padded by 1 followed by as many 0's needed to have a total length of 256 bits. The key is then used to derive 33 subkeys of 128 bits.

We use the notations of [1]. Each intermediate value of the round i is denoted by $\hat{B}_i$ (which is a 128-bit value). The rounds are numbered from 0 to 31. Each $\hat{B}_i$ is composed of four 32-bit words X_0, X_1, X_2, X_3.

Serpent has 32 rounds, and a set of eight 4-bit to 4-bit S boxes. Each round function R_i ($i \in \{0, \ldots, 31\}$) uses a single S box 32 times in parallel. For example, R_0 uses S_0, 32 copies of which are applied in parallel. Thus, the first copy of S_0 takes bits 0 from X_0, X_1, X_2, X_3 and returns the output to the same bits (0). This is implemented as a boolean expression of the 4 registers.

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 we use S_0 again in round 8, then S_1 in round 9, and so on. The last round is slightly different from the others: apply S_7 on $\hat{B}_{31} \oplus \hat{K}_{31}$, and XOR the result with $\hat{K}_{32}$ rather than applying the linear transformation.

The cipher may be formally described by the following equations:

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

where

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

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.

The linear transformation is as follows: The 32 bits in each of the output words are linearly mixed by

$$
\begin{aligned}
X_0, X_1, X_2, X_3 &:= \hat{\mathcal{S}}_i(\hat{B}_i \oplus \hat{K}_i) \\
X_0 &:= X_0 <<< 13 \\
X_2 &:= X_2 <<< 3 \\
X_1 &:= X_1 \oplus X_0 \oplus X_2 \\
X_3 &:= X_3 \oplus X_2 \oplus (X_0 << 3) \\
X_1 &:= X_1 <<< 1 \\
X_3 &:= X_3 <<< 7 \\
X_0 &:= X_0 \oplus X_1 \oplus X_3 \\
X_2 &:= X_2 \oplus X_3 \oplus (X_1 << 7) \\
X_0 &:= X_0 <<< 5 \\
X_2 &:= X_2 <<< 22 \\
\hat{B}_{i+1} &:= X_0, X_1, X_2, X_3
\end{aligned}
$$

where $<<<$ denotes rotation, and $<<$ denotes shift. In the last round, this linear transformation is replaced by an additional key mixing: $B_{32} := S_7(B_{31} \oplus K_{31}) \oplus K_{32}$.

3 Differential Attack on 7- and 8-Round Serpent

In this section we present attacks on 7-round and 8-round Serpent from round 4 to round 10 (or round 11 in the 8-round variant), i.e., encryption starts with S_4 and ends with S_2 (S_3 for the 8-round variant)[1]. In Appendix D a 6-round differential characteristic between round 4 and round 9 with probability 2^{-93} is presented. In the rest of this paper we keep the round numbers as in the corresponding rounds of Serpent, i.e., from round 4 to round 10, rather than from round 0 to round 6.

We adopt the representation of the differential characteristics using figures as in [5], but add more data to the figures. The figures describe data blocks by

[1] Attacks starting from other rounds do not necessarily have the same complexities since the S boxes used in the various rounds are different.

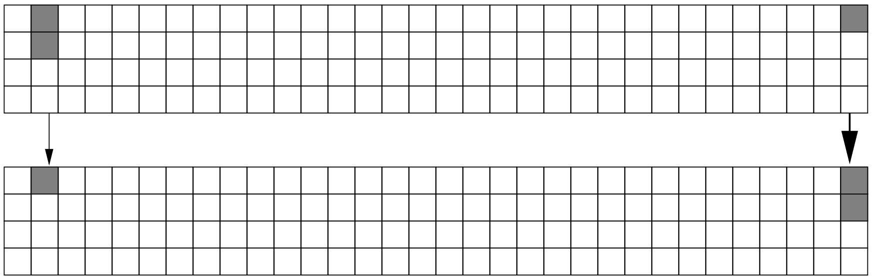


Figure 1: Difference Representation Example

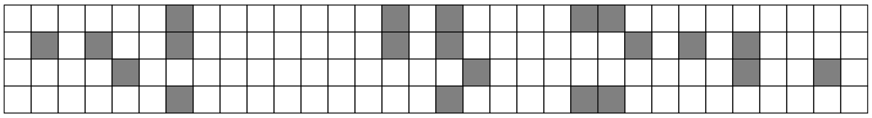


Figure 2: The Input Difference of the 6-Round Differential Characteristic

rectangles of 4 rows and 32 columns. The rows are the bitsliced 32-bit words, and each column is the input to a different S box. The upper line represents X_0, the lower line represents X_3, and the rightmost column represents the least significant bits of the words. A thin arrow represents a probability of 1/8 for the specific S box (given the input difference, the output difference is achieved with probability 1/8), and a fat arrow stands for probability 1/4. If there is a difference in a bit, the box related to it is filled. Example for our notation can be found in Figure 1, in which in the first S box (S box 0; related to bits 0) the input difference 1 causes an output difference 3 with probability 1/4, and in S box 30 input difference 3 causes an output difference 1 with probability 1/8.

The attack uses 2^{14} characteristics with different input differences but the same output difference. The 2^{14} characteristics differ only in the first round, in which they have the same active S boxes with different input differences. All the characteristics have the same differences after the first round, and all have the same probability 2^{-93}. The input difference for one of the 6-round characteristics is presented in Figure 2, and the common output is presented in Figure 3 (the full characteristic is presented in Appendix D).

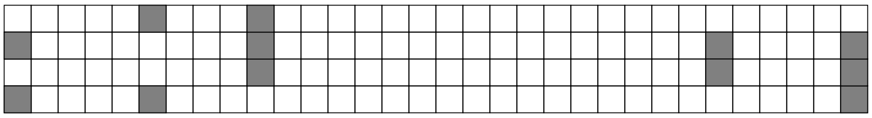


Figure 3: The Output Difference of the 6-Round Differential Characteristic

The attack requires 2^{32} structures of 2^{52} chosen plaintexts each. In each structure all the inputs to the 19 inactive S boxes in the first round are fixed to some (random) value, while the 52 bits of input to the 13 active S boxes receive all the 2^{52} possible values. In these data structures there are $2^{32} \cdot 2^{51} = 2^{83}$ pairs for each possible characteristic. Each characteristic has probability 2^{-93}, therefore, we expect that about $2^{83} \cdot 2^{14} \cdot 2^{-93} = 2^4$ pairs satisfy one of the characteristics. We call these pairs *right pairs*. The number of possible pairs in each structure is $(2^{52})^2/2 = 2^{103}$, thus we have $2^{103} \cdot 2^{32} = 2^{135}$ pairs to consider in total.

Each pair satisfying one of the characteristics has 19 active S boxes in round 10, thus any pair with non-zero output difference in one of the remaining 13 S boxes can be automatically discarded. Thus, about $2^{103-52} = 2^{51}$ candidates for right pairs remain from each structure.

Moreover, in 3 S boxes only 4 output differences are possible if one of the characteristics is satisfied; in 6 S boxes only 6 output differences are possible; in 9 S boxes only 7 output differences are possible, and in the remaining S box eight output differences are possible. Discarding any pair with a wrong output difference using the above filter should keep only a fraction of $\frac{4}{16}^3 \cdot \frac{6}{16}^6 \cdot \frac{7}{16}^9 \cdot \frac{8}{16} \approx 2^{-26.22}$ of the pairs. Thus, only about $2^{51} \cdot 2^{-26.22} = 2^{24.78}$ pairs remain for each structure.

For each structure, we check whether the remaining pairs satisfy one of the 2^{14} possible plaintext differences (corresponding to the 2^{14} characteristics). As there are about 2^{52} possible input differences, only a fraction of about $2^{-52} \cdot 2^{14} = 2^{-38}$ of the pairs remain at this stage. Thus, the expected number of remaining pairs in all the 2^{32} structures is $2^{24.78} \cdot 2^{-38} \cdot 2^{32} = 2^{18.78}$.

For each remaining pair we compute a list of possible whitening subkeys of the 19 active S boxes in the last round. For each active S box, each pair suggests at most 4 values for the subkey of that S box. Thus, counting on m S boxes results in at most $2^{18.78} \cdot 4^m = 2^{18.78+2m}$ hits. The average number of hits (for a wrong value) is $2^{18.78+2m}/2^{4m}$, which is smaller than 1 for $m \geq 10$. On the other hand, the correct subkey is counted for each right pair, i.e., about 16 times, and thus it can be easily identified when $m \geq 10$. Then, we count on the remaining 9 S boxes and take the only value suggested more than two or three times. Note that even if we got more than one possible subkey after counting on 10 S boxes, only one of them is expected to remain after this stage. In total we retrieve 76 subkey bits using at most $2^{38.78}$ one round encryptions and 2^{40} 4-bit counters. We can retrieve 52 additional bits by analyzing the first round as well.

After we retrieve 128 bits of subkey material we can easily find a 128-bit key using linear equations. For 192- and 256-bit keys we can take another set of characteristics. The new set includes the original characteristics used in the attack rotated one bit to the left, i.e., if we have a difference in the least significant bit of X_0 in the original characteristics, we have a difference in the second bit (bit 1) of X_0 in the new set. There is an additional set, in which the rotation is by two bits. (Note that rotation by 3 bits does not make good characteristics). This way we obtain additional 36 subkey bits from round 4 (as out of the 52 bits in the input to the 13 active S boxes there are 16 common bits). This phase of the attack is much simpler, as we already know the common 16 subkey bits, and can easily discard wrong pairs. We also get 32 additional bits from round 10, thus obtaining additional 68 bits (36 from round 4, and 32 from round 10). For 192-bit keys, this information is sufficient to recover the key. For 256-bit keys we can use other differentials with probability 2^{-94} (which are just equivalent to the differential we have used with slight modifications in the last round of the characteristic) using similar techniques and retrieve the remaining unknown bits.

We conclude that the attack requires $2^{52} \cdot 2^{32} = 2^{84}$ plaintexts for 128-bit keys, and twice as much for 192-bit and 256-bit keys. The time complexity of the attack is 2^{85} memory accesses. The memory requirements are 2^{40} 4-bit counters and 2^{52} cells for a hash table.

In order to reduce the time of analysis we perform the algorithm in the following way:

1. For each structure:
   (a) Insert all the ciphertexts into a hash table according the 52 ciphertext's bits of the inactive S boxes in the last round.
   (b) For each entry with collision (a pair of ciphertext with equal 52-bit values) check whether the plaintexts' difference (in round 4) is one of the 2^{14} characteristics' input difference.
   (c) If a pair passes the above test, check whether the difference (in the 76 bits) can be caused by the output difference of the characteristics.
   (d) If a pair passes also the above test, we add 1 to the counter related to the 40 bits of the subkey (as there are $4m$ subkey bits, and for $m = 10$ we get the best results).
2. Collect all the (few) subkeys whose counter has at least 10 hits. With a high probability the correct subkey is in this list (and it is the only one in it).
3. For each pair suggesting a value in the list, we complete the subkey of the other 9 S boxes in round 10, and the 13 S boxes from round 4. As we should have only right pairs (with very few additional wrong pairs), and as the right pairs agree on the rest of the subkey, we can identify the right subkey by intersecting the sets proposed by the various pairs.

.

For each structure 2^{52} memory accesses are performed for the hashing. In the hash table about $1/e$ of the entries are empty, and $1/e$ of the entries contain only one plaintext (and no pairs need to be analyzed). Counting on all the possibilities for the number of plaintexts in each entry of the hash table we conclude that 2^{51} pairs from each structure need to be analyzed. Most of them are discarded by the first filter, and about 2^{13} pairs remain for the second filtering and counting. Therefore, we can estimate the work for each structure as the work needed to hash all plaintexts and then to look at the hash table afterwards, and to perform the search whenever there are more than two plaintexts in one hash entry. The number of pairs we expect to check is 2^{51} and most of them can be discarded almost immediately. We perform about 2^{53} memory accesses for each structure, and the amount of work needed for the whole attack is equivalent to about $2^{33} \cdot 2^{52} = 2^{85}$ memory accesses.

3.1 8-Round 256-Bit Key Serpent

One can easily extend our attack to 8 rounds for the 256-bit key variant by guessing the subkey of round 11. For each possible value of the subkey of round 11 we decrypt the last round and use the attack from the previous subsection. This way, there is no need to make the extra work of completing the key by retrieving other subkeys. The data complexity remains the same 2^{84}, and the time complexity is $2^{128} \cdot 2^{85} = 2^{213}$ memory accesses with 2^{40} counters.

4 The Rectangle Attack

4.1 Amplified Boomerang Attack

The main idea of the amplified boomerang attack [6] is to use two short differential characteristics instead of one long characteristic. Therefore, this technique is very useful when we have good short differential characteristics and very bad long ones.

Let a cipher $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ be composed of two encryption functions E_0 and E_1. Thus, $E = E_1 \circ E_0$. We assume that a good differential is not known for E, but for E_0 we have a differential characteristic $\alpha \to \beta$ with probability p, and for E_1 we have a differential characteristic $\gamma \to \delta$ with probability q, where $pq \gg 2^{-n/2}$.

The basic attack is based on building quartets of plaintexts (x, y, z, w) which satisfy several differential conditions. Assume that $x \oplus y = \alpha$ and $z \oplus w = \alpha$. Each pair has probability p to satisfy the characteristic $\alpha \to \beta$ in E_0. We denote by x', y', z', w' the encrypted values of x, y, z, w under E_0, respectively ($x' = E_0(x), \ldots, w' = E_0(w)$). We are interested in the cases where $x' \oplus y' = \beta$, $z \oplus w' = \beta$ and $x' \oplus z' = \gamma$, as in these cases $y' \oplus w' = (x' \oplus \beta) \oplus (z' \oplus \beta) = \gamma$ as well. We receive two pairs for E_1 each with input difference γ. When encrypting those x', y', z', w' by E_1, in some of the cases the input difference γ becomes δ, and we look for the cases where both differences become $x'' \oplus z'' = \delta$ and

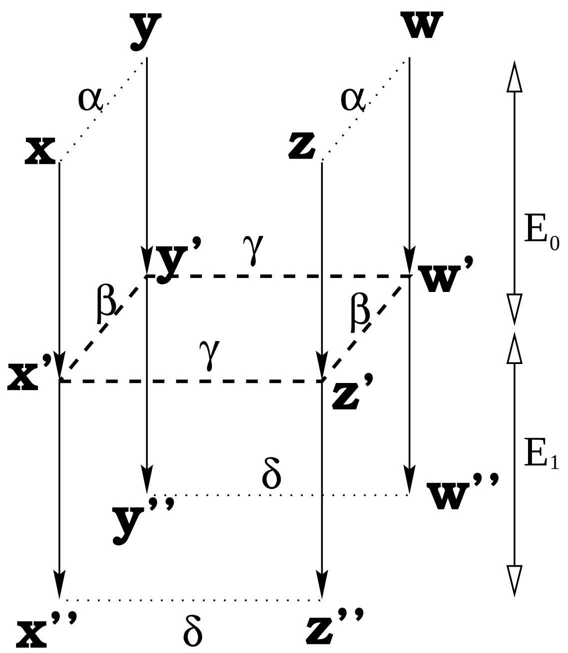


Figure 4: Right Quartet for the Rectangle Attack

$y'' \oplus z'' = \delta$ after E_1. A quartet satisfying all these differential requirements is called a *right quartet*. An outline of such a quartet is shown in Figure 4.

The question which rises is what is the fraction of the right quartets among all the quartets. If we have m pairs with difference α, a fraction of about p of them satisfies the characteristic for E_0. Thus, we have about mp pairs with output difference β in the input to E_1, giving about $(mp)^2/2$ quartets consisting of two such pairs. Assuming that the intermediate encryption values distribute uniformly over all possible values, then with probability 2^{-n} we get x' and z' such that $x' \oplus z' = \gamma$, but once this occurs we automatically get another pair with input difference γ (the pairs are (x', z') and (y', w')). Note that x' and w' have also a probability 2^{-n} to have a difference $x' \oplus z' = \gamma$, thus, given two pairs (x', y') and (z', w') we have two ways to use them as a quartet, with probability 2^{-n+1}. Therefore, we have $(mp)^2/2 \cdot 2^{-n+1}$ quartets which might satisfy our requirements. Each of the pairs satisfies the second characteristic for E_1 with probability q. Thus, starting with m pairs (x, y),(z, w), the expected number of right quartets is:

$$\binom{mp}{2} \cdot 2^{-n+1} \cdot q^2 \cdot 2^{-n} = m^2 \cdot 2^{-n} \cdot (pq)^2.$$

Therefore, the distinguisher counts quartets (x, y, z, w) of plaintexts which satisfy that $x'' \oplus z'' = y'' \oplus w'' = \delta$.

For a random permutation (or for a random value of α and δ) the expected number of quartets is $m^2 \cdot 2^{-2n}$, as there are m^2 possible quartets (there are $m^2/2$ pairs of pairs, and each pair of pairs can create two quartets e.g., $(x, y), (z, w)$ and $(x, y), (w, z)$). For each pair (x, z) or (y, w) the probability of having a specific difference in the output is 2^{-n}. Therefore, if $pq > 2^{-n/2}$, we would count more quartets than random noise. This way when m is sufficiently large we can have a distinguisher which distinguishes between E and a random cipher.

4.2 Rectangling the Boomerang

The first improvement was suggested in [6], in which it was observed that instead of requiring a specific γ, we can count on all possible γ' values for which $\gamma' \to \delta$ by E_1.

Therefore, the probability $\Pr^2(\gamma \to \delta) = q^2$ for the pairs (x', z') and (y', w') to have output difference δ is replaced by the probability $\sum_{any\ \gamma'} \Pr^2(\gamma' \to \delta)$, and we have about

$$\binom{m \cdot \Pr(\alpha \to \beta)}{2} \cdot 2^{-n+1} \cdot \sum_{any\ \gamma'} \Pr{}^2(\gamma' \to \delta)$$

quartets satisfying the rectangle conditions. As a result, we might prefer the difference $x'' \oplus z''$ to be some value δ which has many lower probability characteristics instead of an optimal δ with one characteristic with the highest probability.

Our second improvement is quite similar. Instead of discarding pairs with wrong β value, we sort the pairs into piles according to the output difference (β) of E_0. For each possible pile we perform the original attack. For each pile we have probability $\sum_{any\ \gamma'} \Pr^2(\gamma' \to \delta)$ to have a quartet at the end. The number of pairs in each pile β' is

$$\binom{m \cdot \Pr(\alpha \to \beta')}{2}$$

Thus, we have about

$$\sum_{any\ \beta'} \binom{m \cdot \Pr(\alpha \to \beta')}{2} \cdot 2^{-n+1} \cdot \sum_{any\ \gamma'} \Pr{}^2(\gamma' \to \delta) =$$

$$m^2 \cdot 2^{-n} \cdot \sum_{any\ \beta'} \Pr{}^2(\alpha \to \beta') \cdot \sum_{any\ \gamma'} \Pr{}^2(\gamma' \to \delta)$$

quartets for the second step of the attack.

Our third improvement is based on the first two. We can take into consideration more quartets. Assume that for the first pair the difference α causes

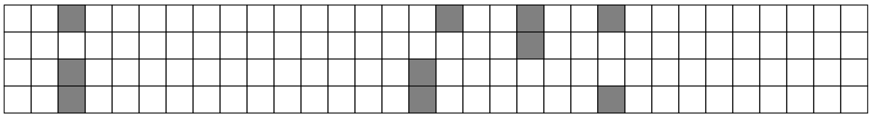


Figure 5: The Input Difference α of the Rectangle Attack

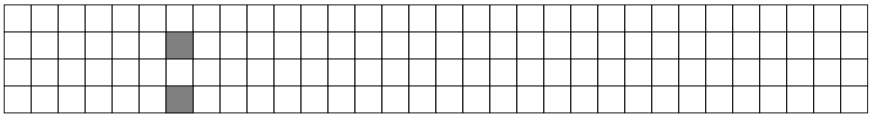


Figure 6: The Output Difference δ of the Rectangle Attack

some difference a, and for the second pair $\alpha \to b$. Then, we can count also characteristics for which $\gamma \to \delta$ and $\gamma \oplus a \oplus b \to \delta$. This way the number of quartets is

$$m^2 \cdot 2^{-n} \cdot \sum_{a,b} \left[\Pr(\alpha \to a) \Pr(\beta \to b) \cdot \sum_{\gamma} \Pr(\gamma \to \delta) \Pr(\gamma \oplus a \oplus b \to \delta) \right]$$

Note that this improvement counts all the quartets with plaintext difference α and ciphertext difference δ. However, it is very hard to do the exact calculation.

5 Attacking 10-Round Serpent

In Section 4 we presented a method to build a distinguisher for a function $E = E_1 \circ E_0$. We now present a method to use the distinguisher to find subkey material.

We attack a 10-round 256-bit key Serpent (round 0 to round 9) using an 8-round rectangle distinguisher. In this distinguishing attack E_0 is rounds 1–4 of Serpent, and E_1 is rounds 5–8. The basic differential characteristic ($\alpha \to \beta$) used in rounds 1–4 is also the best known 4-round differential characteristic of Serpent. This characteristic and the basic differential characteristic used in rounds 5–8 are presented in Appendix B. α and δ are presented in Figure 5 and Figure 6, respectively.

The first differential has probability of 2^{-29}. Using the second improvement and counting on all possible output differences of round 4, we receive $\sum_{\beta'} \Pr^2[\alpha \to \beta'] = 2^{-50.8}$. The second differential has probability of 2^{-47}. Using the first improvement and counting on a large set of characteristics (all are very similar to the basic one, and have the same last two rounds) we receive $\sum_{\gamma'} \Pr^2[\gamma' \to \delta] = 2^{-69.8}$. These probabilities were computed by a computer program which scanned characteristics similar to those presented in Appendix B.

For $m = 2^{125.8}$ pairs with the difference α of Figure 5 about $2 \cdot (2^{125.8} \cdot 2^{-25.4})^2/2 = 2^{200.8}$ quartets can be formed after the fourth round. The probability to get a specific γ is about 2^{-128}, thus the expected number of quartets with a given γ is about $2^{72.8}$ quartets. As $\sum_{\gamma'} \Pr^2[\gamma' \to \delta] = 2^{-69.8}$, the number of right quartets is 8.

To attack 10-round Serpent (rounds 0–9) we use a similar technique to the one used in [6]. We request $2^{62.8}$ structures of 2^{64} plaintexts each. The structures are chosen so that each structure varies over all the possible inputs to the active S boxes in round 1, while the input of the rest of the S boxes is kept fixed (this is done by checking which S boxes in round 0 affect the active bits in round 1, and trying all the inputs to these S boxes). Using this procedure for choosing the plaintexts we get $2^{125.8}$ pairs with difference α after round 0.

We keep all the plaintexts and their corresponding ciphertexts in a large table (whose size is $2^{126.8} \cdot 2 \cdot 16 = 2^{131.8}$ bytes of memory), and keep 2^{84} 4-bit counters, where each counter corresponds to one of the possible values of the 84 bits of the subkeys we search for (64 bits entering 16 S boxes in the first round, and 20 bits entering 5 S boxes in the last round). In order to count the number of quartets with the given α and δ we perform the following algorithm:

1. Initialize the counter's array with 0's.
2. For each 64-bit subkey value in round 0, for each 20-bit value subkey value in round 9, and for each plaintext x:
   - Partially encrypt x through round 0 in the 16 S boxes and denote the value we get by x_1.
   - Calculate $x_1 \oplus \alpha$ and denote this value by y_1.
   - Partially decrypt y_1 through round 0 in the 16 S boxes, and find the corresponding plaintext, which we denote by y (this plaintext y exists in our data, due to the way we choose the structures). The value of the plaintext bits of y related to the other 16 S boxes is the same as of x.
   - Let x'' and y'' be the corresponding ciphertexts of x and y respectively. then,
   - Partially decrypt through the 5 active S boxes x'' and y'', denote the value you get by x''_9 and y''_9, respectively.
   - Partially encrypt $x''_9 \oplus \delta$ and $y''_9 \oplus \delta$ and check whether the corresponding ciphertexts exist in our data. If these ciphertexts exist, we check their corresponding plaintexts, whether under the guessed 64-bit subkey of round 0 we get a difference α. If so, we increase the corresponding counter by 1.
3. Run over all counters, and print the corresponding indices whose counter is greater than or equal to 7.

The inner loop is performed at most $2^{84} \cdot 2^{126.8}$ times, and includes at most 4 times encrypting 16 S boxes (equivalent to two rounds of Serpent) and 4 times decrypting 5 S boxes (equivalent to 5/8 rounds of Serpent). Thus, the time complexity of the attack is at most $2^{84} \cdot 2^{125.8} \cdot 2\frac{5}{8}/10 \approx 2^{208.4}$ 10-round Serpent encryptions. The time complexity can be reduced by half by building in advance an equivalent table in which each entry i contains $S_1(S_1^{-1}(i) \oplus \delta)$ and use it in the last round, and similarly computing a table with $S_0^{-1}(S_0(i) \oplus \alpha)$ for the first round.

5.1 Reducing Time Requirements

One can also use the technique of hash tables presented in [5] to reduce the time complexity to 2^{205} memory accesses, in exchange for increasing the memory complexity to 2^{196} bytes of RAM.

6 Summary

In this paper we presented the best published attack on 10-round 256-bit key Serpent. The attack requires $2^{126.8}$ chosen plaintexts, $2^{207.4}$ time and $2^{131.8}$ bytes of RAM. A variant of the attack requires 2^{205} time but 2^{196} bytes of RAM.

We presented a differential attack on 7-round Serpent, which works for all key sizes, with data complexity of 2^{84} chosen plaintexts, time complexity of 2^{85} memory accesses and 2^{52} memory (blocks of 128-bit). We presented an attack on 8-round 256-bit key Serpent requiring 2^{84} chosen plaintexts, 2^{213} time and 2^{84} memory (blocks of 128-bit). We summarize these results in Table 1.

We also presented the best known 3-round, 4-round, 5-round and 6-round differential characteristics of Serpent, whose probabilities are 2^{-15}, 2^{-29}, 2^{-60} and 2^{-93}, respectively. In Table 2 we summarize these characteristics and the best previously published characteristics.

7 Acknowledgment

The authors would like to thank Prof. R. Adler of the Industrial Engineering Department at the Technion for the fruitful discussions related to probability theory. The authors would also like to thank R. Anderson for his remarks concerning this paper.

References

[1] R. Anderson, E. Biham, L.R. Knudsen, *Serpent: A Proposal for the Advanced Encryption Standard*, NIST AES Proposal, 1998.

Rounds	Key Size	Complexity			Source
		Data	Time	Memory	
6	all	2^{83}	2^{90}	2^{40}	[5] - Section 3.2
	all	2^{71}	2^{103}	2^{75}	[5] - Section 3.3
	192 & 256	2^{41}	2^{163}	2^{45}	[5] - Section 3.4
7	256	2^{122}	2^{248}	2^{126}	[5] - Section 3.5
	all	2^{84}	2^{85} MA	2^{52}	This paper
8	192 & 256	2^{128}	2^{163}	2^{133}	[5] - Section 4.2
	192 & 256	2^{110}	2^{175}	2^{115}	[5] - Section 5.3
	256	2^{84}	2^{213} MA	2^{84}	This paper
9	256	2^{110}	2^{252}	2^{212} bytes	[5] - Section 5.4
10	256	$2^{126.8}$	$2^{207.4}$	$2^{131.8}$ bytes	This paper
	256	$2^{126.8}$	2^{205}	2^{196} bytes	This paper

MA - Memory Accesses
Memory unit is one block, unless written otherwise

Table 1: Summary of Differential Attacks on Serpent with Reduced Numbers of Rounds

Number of Rounds	Paper	Starting from	Number of Active S boxes	Probability
3	[5]	S_5	7	2^{-16}
	This paper*	S_2	7	2^{-15}
4	[5]	S_1	14	2^{-31}
	[10]	S_6	14	2^{-34}
	This paper	S_1	13	2^{-29}
5	[5]	S_1	38	2^{-80}
	[10]	S_5	24	2^{-61}
	This paper	S_5	25	2^{-60}
6	[10]	S_1	41	2^{-97}
	This paper	S_4	38	2^{-93}

* This is also the upper bound presented in this paper.

Table 2: Summary of the Differential Characteristics of Serpent

[2] E. Biham, *A Note on Comparing the AES Candidates*, Second AES Candidate Conference, 1999.

[3] E. Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[4] O. Dunkelman, *An Analysis of Serpent-p and Serpent-p-ns*, presented at the rump session of the Second AES Candidate Conference, 1999. Available on-line at *http://vipe.technion.ac.il/~orrd/crypt/*.

[5] T. Kohno, J. Kelsey, B. Schneier, *Preliminary Cryptanalysis of Reduced-Round Serpent*, Third AES Candidate Conference, 2000.

[6] J. Kelsey, T. Kohno, B. Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 2000, to appear.

[7] L.R. Knudsen, *Truncated and Higher Order Differentials*, proceedings of Fast Software Encryption 2, Springer-Verlag, LNCS 1008, pp. 196–211, 1995.

[8] NIST, *A Request for Candidate Algorithm Nominations for the AES*, available on-line at *http://www.nist.gov/aes/*.

[9] D. Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 1999, Springer Verlag, LNCS 1636, pp. 156–170, 1999.

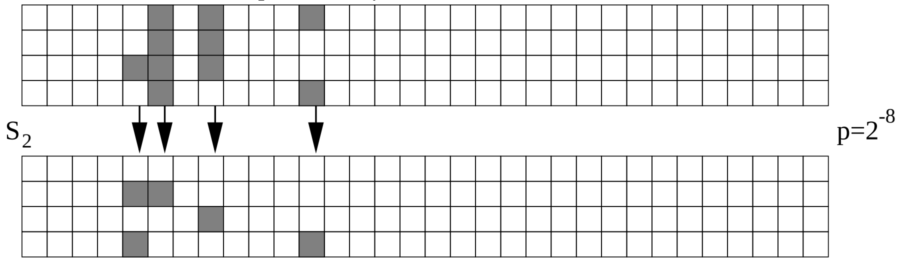
[10] X.Y. Wang, L.C.K. Hui, C.F. Chong, W.W. Tsang, H.W. Chan, *The Differential Cryptanalysis of an AES Finalist - Serpent*, Technical Report TR-2000-04. Available on-line at: *http://www.csis.hku.hk/research/techreps/*.

A A 3-Round Differential Characteristic

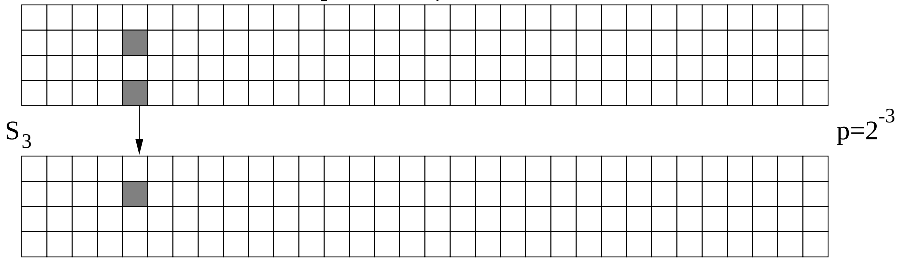
Our 3-round differential characteristic is based on the one found in [5], where a 3-round differential characteristic with 7 active S boxes and probability 2^{-16} is presented. The characteristic is based on 4 active S boxes in the first round, 1 in the second round and 2 in the last round. The problem in finding characteristics is not finding the first round's input and the last round's output of the S boxes, as they can be chosen to have maximal probability. The problem is to have a minimal number of active S boxes, which is related to the output of the first round (which passes the linear transformation), the second round, and the input for the last round (as this determines the number of active S boxes in the last round).

We start by selecting the differences of the second round in a similar way to [5]. We observe that if we use the second round of the characteristic having S_3 instead of S_6 and having probability 1/8 in S_3, we can ensure that all active S boxes in rounds 2 and 4 of the cipher (which are the first and third rounds of the characteristic, respectively) have probability 1/4, thus having a total probability of 2^{-15}.

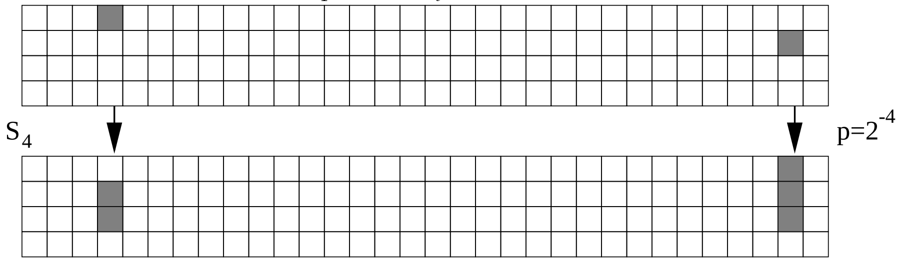
The 3-round differential characteristic with probability 2^{-15} that we get is as follows: In round 2 (or 10 or 18 or any other round having S_2) the following characteristic holds with probability 2^{-8}:



After the linear transformation and the application of S_3 we get the following differential characteristic with probability 2^{-3}:



After the linear transformation and the application of S_4 we get the following differential characteristic with probability 2^{-4}:



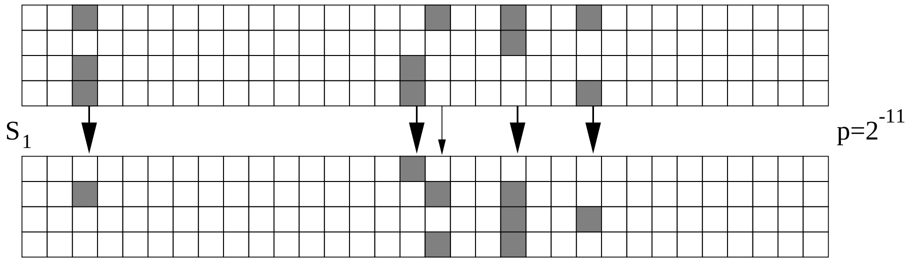
During the search for the best characteristic, we exhaustively checked all possible 3-round characteristics with 7 active S boxes and found this to be the best possible characteristic. As all 3-round characteristics have at least 7 active S boxes, and with 8 active S boxes the probability of the characteristic is at most 2^{-16}, this proves that this is the best 3-round differential characteristic of Serpent.

B A 4-Round Differential Characteristics

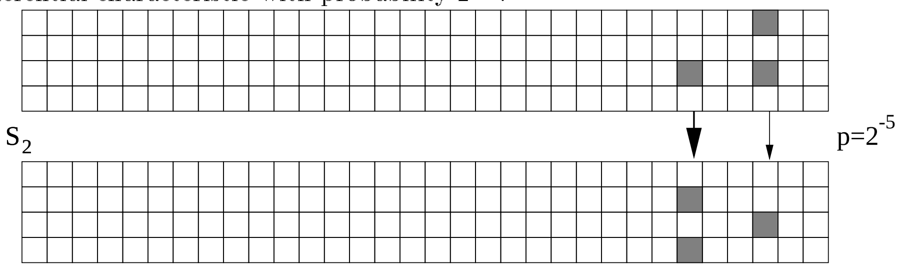
B.1 A 4-Round Characteristic for Rounds 1–4

One option for achieving a minimal number of S boxes (13 according to [1]) is to have in the second round's S box S_2 $5 \to 4$ and $4 \to A_x$, and in the third round to have an active S box S_3 with $4 \to A_x$. Of course we would like to maximize the probabilities of these entries.

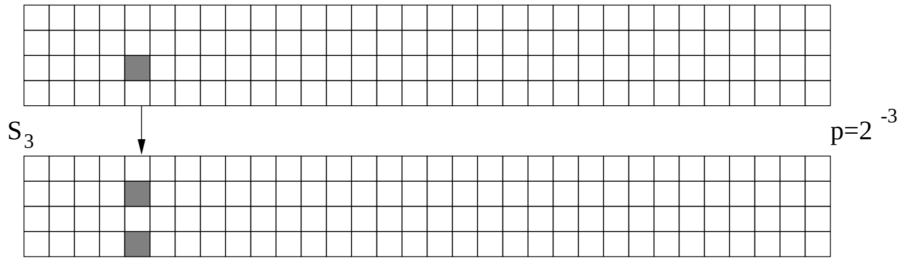
Checking the S boxes for such instances we found out that the best characteristic is when the first round of the 4-round characteristic is set at rounds using S_1. We receive the following 4-round differential characteristic with probability 2^{-29}:
In round 1 (or any other round having S_1) the following characteristic holds with probability 2^{-11}:



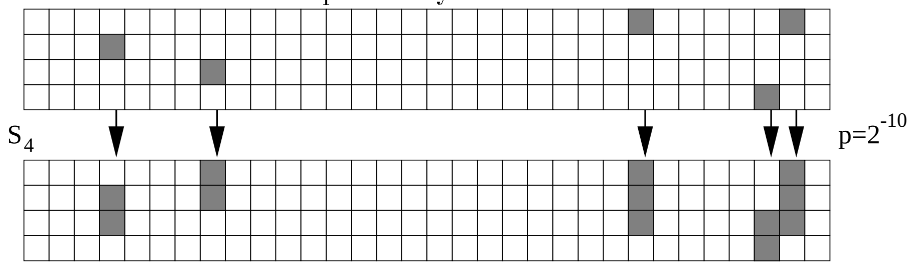
After the linear transformation and the application of S_2 we get the following differential characteristic with probability 2^{-5}:



After the linear transformation and the application of S_3 we get the following differential characteristic with probability 2^{-3}:

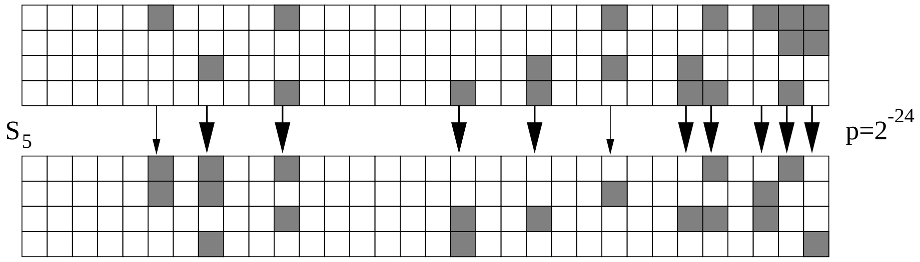


After the linear transformation and the application of S_4 we get the following differential characteristic with probability 2^{-10}:

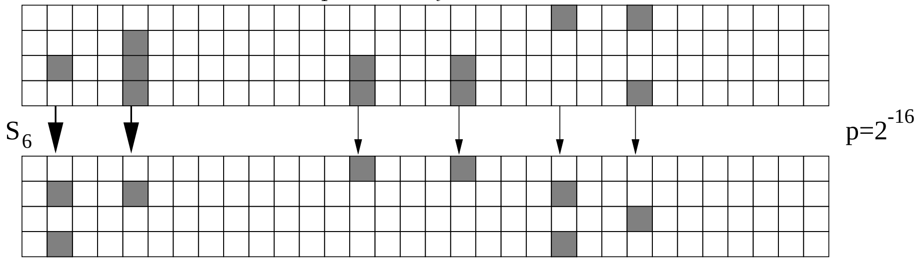


B.2 A 4-Round Characteristics for Rounds 5–8

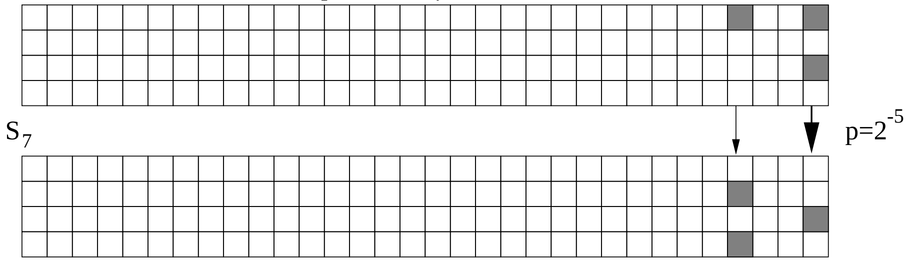
This second 4-round differential characteristic is used along with the previous one in the attack of Section 5, and has probability of 2^{-47}. We use the basic characteristic described in [10], where a 5-round differential characteristic with probability 2^{-61} is described. As we need a characteristic of round 5–8, we remove the last round and get a 4-round characteristic with probability 2^{-48}. As part of our efforts to find higher probability differential characteristics for the amplified boomerang attack, we try a technique found very useful in previous attempts: we add another active S box in the first round. This might seem a bad thing (as this reduces the probability) but we found out that in exchange we get 3 more entries with probability 1/4 instead of 1/8. Thus, our characteristic has probability of 2^{-47}.
In round 5 (or any other round having S_5) the following characteristic holds with probability 2^{-24}:



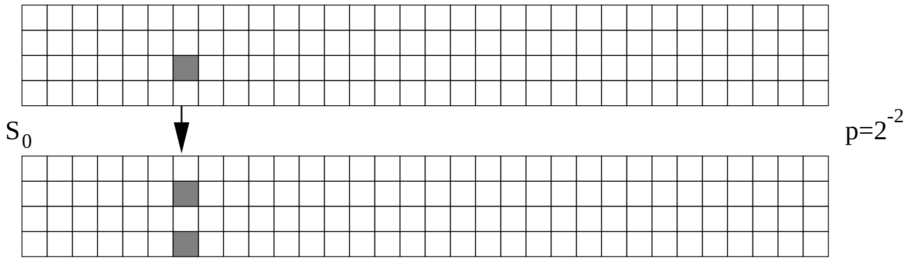
After the linear transformation and the application of S_6 we get the following differential characteristic with probability 2^{-16}:



After the linear transformation and the application of S_7 we get the following differential characteristic with probability 2^{-5}:



After the linear transformation and the application of S_0 we get the following differential characteristic with probability 2^{-2}:

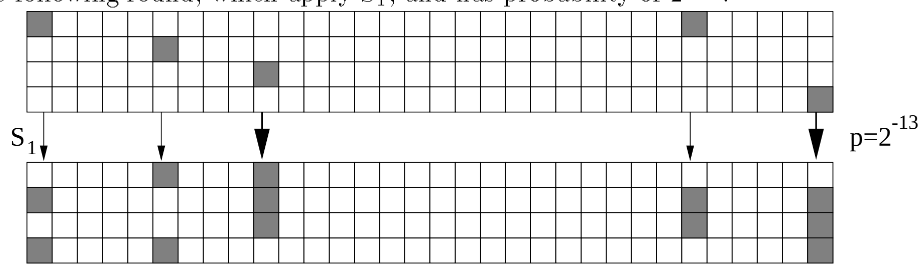


Note that the last two rounds are the same as in [10].

C A 5-Round Differential Characteristic

As stated in Appendix B, we took a 5-round characteristic from [10], truncated it and improved it to have 4-round characteristic. By adding the last round from [10] back to the characteristic we get a 5-round characteristic with probability 2^{-60}.

Thus, we add after the 4th round of the characteristic from Appendix B.2 the following round, which apply S_1, and has probability of 2^{-13}:



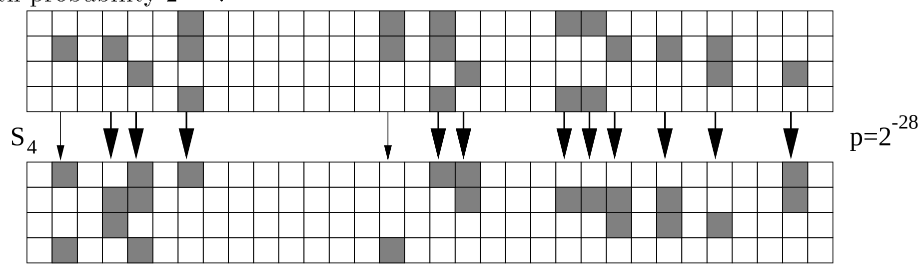
We have found another 5-round differential characteristic with probability 2^{-60}, and three more characteristics with probability 2^{-61} which are quite similar to the this one.

D A 6-Round Differential Characteristic

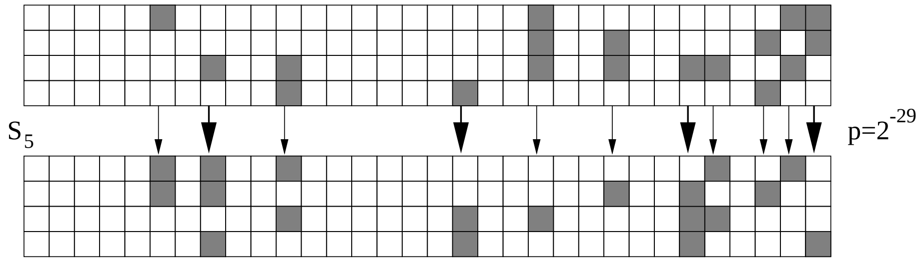
In order to get the best 6-round characteristic we can, we add a round before the 5-round characteristic from Appendix C and alter the first two rounds of it.

Thus, the 6-round characteristic starts in a round using S_4, and has probability 2^{-93}.

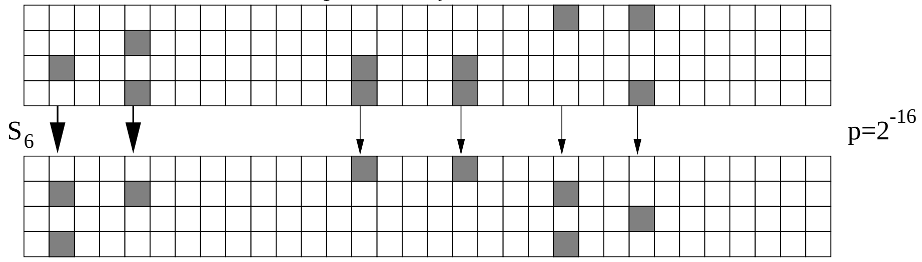
In round 4 (or any other round having S_4) the following characteristic holds with probability 2^{-28}:



After the linear transformation and the application of S_5 we get the following differential characteristic with probability 2^{-29}:



After the linear transformation and the application of S_6 we get the following differential characteristic with probability 2^{-16}:



After this round the characteristic is the same as described for S_7, S_0 and S_1 in Appendices B.2 and C.

We observe that there are 2^{14} 6-round characteristics with the same last 5-rounds (only the input difference of the first round changes). This follows from the fact that in 2 S boxes in the first round we have 8 possible input differences with the same probability for the given output differences and in 8 S boxes we have two possibilities.

We also observed that by rotating all the characteristics one bit to the left (or two) the characteristics remain valid with the same probability (rotation by three or more bits does not work).