Truncated Differentials of SAFER

Lars R. Knudsen*[1]
Thomas A. Berson[2]

[1] Laboratoire d'Informatique, École Normale Supérieure, Paris, France
[2] Anagram Laboratories, Palo Alto, CA 94301, USA

Abstract. In this paper we do differential cryptanalysis of SAFER. We consider "truncated differentials" and apply them in an attack on 5-round SAFER, which finds the secret key in time much faster than by exhaustive search.

1 Introduction

In [6] a new encryption algorithm, SAFER K-64, hereafter denoted SAFER, was proposed. Both the block and the key size is 64. The algorithm is an iterated cipher, such that encryption is done by iteratively applying the same function to the plaintext in a number of rounds. The suggested number of rounds is minimum 6 and maximum 10 [6, 7]. Finally an output transformation is applied to produce the ciphertext. Strong evidence has been given that the scheme is secure against differential cryptanalysis after 5 rounds [7] and against linear cryptanalysis after 2 rounds [2]. In [9] it was shown that by replacing the S-boxes in SAFER by random permutations, about 6% of the resulting ciphers can be broken faster than by exhaustive search. In [4] a weakness in the key schedule of SAFER was exploited to establish a key-related chosen plaintext attack faster than an exhaustive search for the key. Furthermore, it was shown that for SAFER with 6 rounds used in the standard hashing modes collisions can be found much faster than by a brute force attack. Also, in [8] it was shown that there exists a projection on the plaintext and ciphertext spaces that is independent on one quarter of the key. As a consequence of all this, Massey decided recently to adopt the proposed stronger key schedule suggested by Knudsen in [4] and to recommend that 8 rounds is used for SAFER with a 64 bit key. The new cipher has been named SAFER SK-64. Massey also proposed 128 bit key variants of both versions, namely SAFER K-128 and SAFER SK-128, respectively.

In this paper we consider "truncated differentials" and apply them in an attack on 5-round SAFER, the original version, which finds the secret key much faster than by exhaustive search. The attack uses a 5-round truncated differential of probability 2^{-70}, which can be obtained using only about 2^{39} chosen plaintexts. The attack uses several of these differentials, needs totally about 2^{45} chosen plaintexts and runs in time similar to 2^{46} encryptions of 5-round SAFER. Another version of the attack needs totally about 2^{46} chosen plaintexts and runs

* Postdoctoral researcher sponsored by the Danish Technical Research Council.

in time similar to 2^{35} encryptions of 5-round SAFER. This should be compared to the analysis made in [7], where a differential attack using conventional differentials on SAFER with 5 rounds was estimated to require more computations than a brute force exhaustive attack.

Our attack is independent of the S-boxes used in SAFER and furthermore it needs only a small amount of chosen plaintext compared to conventional differential attacks [1] and illustrates the importance of truncated differentials.

2 Description of SAFER

SAFER is an r round iterated cipher with both block and key size of 64 bits and with all operations done on bytes. The key is expanded to $2r+1$ round keys each of 64 bits, described later. The designer's recommendation for r is 6 [6]. Each round takes 8 bytes of text input and two round keys each of 8 bytes. The input and the round keys are divided into 8 bytes and the first round key is xor'ed, respectively added modulo 256, according to Fig. 1. The bytes are then processed using 2 permutations or S-boxes, $X(a) = (45^a \bmod 257) \bmod 256$, and the inverse of X, $L(a) = log_{45}(a) \bmod 257$ for $a \neq 0$ and where $L(0) = 128$. After the S-boxes each byte of the second round key is added modulo 256, respectively xor'ed, and finally the so-called *Pseudo-Hadamard Transformation (PHT)* is applied to produce the output of the round. PHT is defined by three layers of the 2-PHT, which is defined by

$$2\text{-}PHT(x, y) = (2 * x + y, x + y)$$

where each coordinate is taken modulo 256. Between two layers of 2-PHT's a permutation of the bytes is done, see Fig. 1. After the last round an output transformation, OT, is applied, which consists of xor'ing, respectively adding modulo 256, the last-round key. Let $o_1, ..., o_8$ be the eight bytes of the output after r rounds, and let $k_1, ..., k_8$ be the eight bytes of the last-round key. The ciphertext is defined

$$OT(o_1, ..., o_8, k_1, ..., k_8) =$$
$$(o_1 \oplus k_1, o_2 + k_2, o_3 + k_3, o_4 \oplus k_4, o_5 \oplus k_5, o_6 + k_6, o_7 + k_7, o_8 \oplus k_8).$$

The key of 64 bits is expanded to $2r+1$ round keys each of 64 bits in the following way. Let $K = (k_{1,1}, ..., k_{1,8})$ be an 8 byte key. The round key byte j in round i is denoted $K_{i,j}$. The round key bytes are derived as follows: $K_{1,j} = k_{1,j}$ for $j = 1, ..., 8$ and

$$k_{i,j} = k_{i-1,j} << 3$$
$$K_{i,j} = k_{i,j} + bias[i, j] \bmod 256$$

for $i = 2, ..., 2r+1$ and $j = 1, ..., 8$. '<< 3' is a bitwise rotation 3 positions to the left and $bias[i, j] = X(X(9i + j))$, where X is the exponentiation function described above.

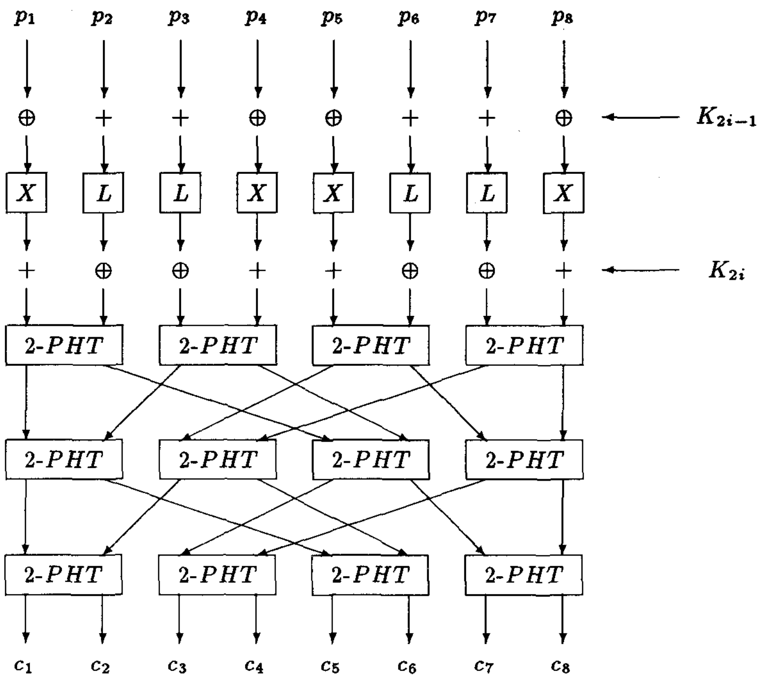

Fig. 1. One round of SAFER.

The newly suggested SAFER SK-64 varies from SAFER in the suggested number of rounds, which is 8, and in the key schedule. Let $K = (k_{1,1}, ..., k_{1,8})$ be an 8 byte key. Define $k_{1,9} = \bigoplus_{i=1}^{8} k_{1,i}$. The round keys $K_{i,j}$, are defined as follows.

For $j = 1, ..., 8: \;\; K_{1,j} = k_{1,j}$

For $i = 2, ..., 2r + 1,$

For $j = 1, ..., 9: \;\; k_{i,j} = k_{i-1,j} << 3$

For $j = 1, ..., 8: \;\; K_{i,j} = k_{i,(i+j-2 \bmod 9)+1} + bias[i, j] \bmod 256$

The 128 bit versions of SAFER differ from SAFER in the suggested number of rounds which is 10 and in the key schedule. The key schedule is essentially two key schedules of the respective 64 bit version, such that the odd no. round keys are taken from the first key schedule and the even no. round keys from the second key schedule. A 128 bit version is compatible with its 64 version, if the two 64 bit key halves input to the key schedule are equal.

3 Differentials of SAFER

In [7] strong evidence was given that SAFER is secure against differential cryptanalysis. It was argued that a 5-round differential for SAFER will have a probability of much less than 2^{-57}, and that a differential attack will require more computations than a brute force search for the key.

In this section we consider other types of differentials than the ones given in [7]. We will use the notation of "expanded views" from [7] and denote a one round differential by three tuples of each 8 entries. The first tuple indicates the difference in the 8 bytes of the inputs to the round, the second tuple indicates the difference of the bytes before the PHT-transformation and the third tuple indicates the difference of the bytes after the PHT-transformation, i.e. the difference of the outputs of the round. For convenience, when considering s-round differentials for $s > 1$, we will omit the third tuple in all but the last round, since the output difference of one round equals the input difference to the following round. Throughout this paper, a *difference* of two bytes (a, b) is defined as

$$a - b \bmod 256.$$

A differential that predicts only parts of an n bit value is called a *truncated differential* [5]. More formally, let (a, b) be an i-round differential. If a' is a subsequence of a and b' is a subsequence of b, then (a', b') is called an i round truncated differential.

In [7] ten tables of "PHT correspondences" are given. The truncated differentials we have found follows from these properties of the PHT transformation. As an example, consider the following one-round differential with the expanded view

$$[a, b, c, d, 0, 0, 0, 0], [e, f, -e, -f, 0, 0, 0, 0], [2g, g, 2h, h, 0, 0, 0, 0], \qquad (1)$$

where $g = 2e + f$ and $h = e + f$. This truncated differential has probability 2^{-16} on average for all values of a, b, c, d. Consider the first and second tuples of (1). A difference a in the first byte and a difference c in the third byte will yield differences e and $-e$, respectively, with an average probability of 2^{-8}. More formally, let $Pr(a \to e)$ denote that an input difference a to an S-box yields an output difference e, then

$$2^{-16} \times \sum_a \sum_c \sum_e Pr(a \to e) \times Pr(c \to -e) =$$
$$2^{-16} \times \sum_c \sum_e Pr(c \to -e) \times \sum_a Pr(a \to e) =$$
$$2^{-16} \times \sum_c (\sum_e Pr(c \to -e)) =$$
$$2^{-16} \times \sum_c 1 = 2^{-8}$$

Since the round key bytes are independent, the probability of the differential can be calculated by multiplying the probabilities for the differentials for every single S-box. Similarly, a difference b in the second byte and a difference d in the third byte will yield differences f and $-f$, respectively, with an average probability of 2^{-8}. The PHT transformation takes the second tuple into the third tuple which is easily verified. As another example, consider the following one-round differential with the expanded view

$$[0, 0, 0, 0, 0, 0, a, b], [0, 0, 0, 0, 0, 0, 0, 0, e, -e], [e, e, 0, 0, e, e, 0, 0]. \tag{2}$$

This truncated differential has probability 2^{-8} on average for all values of a, b. In the above examples, we did not state any specific values of the non-zero bytes. We do not intend to predict the actual values of the non-zero bytes, merely predict the bytes which are zero. There are many one-round differentials like (1) and (2) above. To save space, we introduce a new notation. We will denote a differential by the indices of the bytes which are non-zero. We will write $1234 \rightarrow 1234$ for the differential (1) and, similarly, $78 \rightarrow 1256$ for the differential (2). In the appendix, Tables 2 and 3, many such differentials are listed. E.g. the differential (1) can be found in Table 3 as Input: 1234, Output: 1234, Prob. 16.

As we will show now, one can concatenate the one-round differentials of Tables 2 and 3. Consider the following three-round truncated differential

1. $[a, b, c, d, 0, 0, 0, 0], [e, f, -e, -f, 0, 0, 0, 0]$,
2. $[2g, g, 2h, h, 0, 0, 0, 0], [i, j, -i, -j, 0, 0, 0, 0]$,
3. $[2k, l, 2k, l, 0, 0, 0, 0], [m, n, -m, -n, 0, 0, 0, 0], [2p, p, 2q, q, 0, 0, 0, 0]$,

where $g = 2e + f$ and $h = e + f$ etc. In the other notation, the differential is $1234 \rightarrow 1234 \rightarrow 1234 \rightarrow 1234$. The probability in the first round is 2^{-16}, as we saw earlier. The probabilities in the second round and in the third round will both be approximated by 2^{-16}, although the input differences are dependent. The overall probability for the three-round differential is approximated by the product of the probabilities of the three one-round differentials, in this case 2^{-48}. Since the round keys are dependent this is not a correct way to calculate the probability. Despite this, and the fact that the input differences to pairs of two bytes in both the second and third rounds are dependent, computer experiments have shown that the probability is well approximated this way which is illustrated later. Consider now the following three-round differential.

1. $[a, b, c, d, 0, 0, 0, 0], [e, -e, f, -f, 0, 0, 0, 0]$,
2. $[2g, g, 0, 0, 2h, h, 0, 0], [i, j, 0, 0, -i, -j, 0, 0]$,
3. $[2k, 0, 2l, 0, k, 0, l, 0], [m, 0, -m, 0, n, 0, -n, 0], [2p, 2q, p, q, 0, 0, 0, 0]$.

or similarly, $1234 \rightarrow 1256 \rightarrow 1357 \rightarrow 1234$. This differential has also a probability of 2^{-48}. Now since the two above differentials have the same input difference and the same output difference, that is, the outputs differ in the same bytes, a truncated differential with input difference $[a, b, c, d, 0, 0, 0, 0]$ and output difference $[x, y, z, w, 0, 0, 0, 0]$ will contain both the above differentials. There are totally 8 differentials each of probability 2^{-48} covered by this truncated differential, which therefore will have a probability of about $8 \times 2^{-48} = 2^{-45}$.

4 Differential attacks on SAFER

In this section we consider differential attacks using truncated differentials for SAFER. Consider 3-round SAFER and the 3-round truncated differential with input difference $[a, b, c, d, 0, 0, 0, 0]$ and output difference $[x, y, z, w, 0, 0, 0, 0]$. The probability of the differential is approximately 2^{-45}. In a conventional differential attack with a differential of probability p one needs about $2/p$ chosen plaintexts to get a right pair [1]. Using the above truncated differential for SAFER we can choose n different plaintexts, all of them with the four rightmost bytes of equal values. From these n plaintexts one can form about $(n \times (n-1))/2 \approx \frac{n^2}{2}$ pairs of plaintexts with an input difference zero in the four rightmost bytes. As an example, by choosing 2^{23} plaintext this way, one obtains about 2^{45} pairs with the desired difference and thus with a high probability one right pair. How does one identify a right pair? Pairs with non-zero difference in the four rightmost bytes of the outputs after three rounds can be discarded. A wrong pair has a zero difference in these bytes with probability 2^{-32}. This filtering of pairs leaves only 2^{13} pairs. Note that the output transformation of SAFER is applied after the third round of encryption. Therefore we cannot determine wrong pairs by looking at the difference in the four leftmost bytes directly. Each of the 2^{13} pairs will suggest about 2^{16} values of the four leftmost key bytes in the first round. The remaining key bytes can be found by exhaustive search. In this case the complexity of the attack is about $1/2 \times 2^{61} = 2^{60}$. Additional filtering is possible and would decrease the complexity dramatically, but we omit the details here. The attack in general goes as follows

1. Get the encryptions of the n chosen plaintexts.
2. Check for wrong pairs.
3. Get the key candidates for all non-discarded pairs.
4. Do an exhaustive search for all remaining key bits.

The storage of plaintexts is of great importance. In the general attack one needs to store about n 64 bit quantities. Sorting the ciphertexts will take time about $n \log n$ simple operations. This can be reduced to n operations by using a hash table and a hash function to store values equal in the four rightmost bytes in the same entry, which is illustrated later. We will assume that the time to store (and sort) the ciphertexts is small and unimportant compared to the time to get the n encryptions.

The above estimation is only an approximation, since, first, the round keys of SAFER are not independent as assumed, second, the many pairs we get are not independent. To justify the above method of estimating the probabilities, we did some tests on a mini-version of SAFER. Instead of working on bytes we let SAFER work on nibbles (4 bits), i.e. a 32-bit block cipher with a 32-bit key, called SAFER(32). We define $X_4(a) = (3^a \bmod 17) \bmod 16$, and the inverse of X_4, $L_4(a) = log_3(a) \bmod 17$ for $a \neq 0$ and where $L(0) = 8$. Since 17 is a prime number, exponentiation with the primitive element, 3, is a permutation. All xor operations are on nibbles and additions are calculated modulo 16. We considered the 5-round truncated differential $1234 \rightarrow 5678$ in SAFER(32). There are 824 different differentials in this truncated differential, each of probability 2^{-40}, and the overall probability of the truncated differential is about $2^{-30.3}$. We used structures consisting of 2^{16} plaintexts, all different in the four leftmost bytes and equal in the four rightmost bytes. From each structure we obtain about 2^{31} pairs, of which the expected number of right pairs is 1.6 and about $2^{31}/2^{16} = 2^{15} = 32768$ pairs will have zero difference in the four leftmost bytes, but are wrong pairs. In ten structures of each 2^{16} plaintexts and each with a different key we found 17 right pairs and 327781 wrong pairs, thus confirming our theory. In the following section we show how to attack 5-round SAFER, 64 bits, using truncated differentials.

4.1 A differential attack on 5-round SAFER

Consider the following 4-round truncated differential with input difference

$$[a, 0, 0, b, c, 0, 0, d]$$

and output difference $[0, 0, 0, 128, 0, 0, 0, 0]$ There are four differentials in this truncated differential, each of probability $2^{-71.7}$. They are

$$1458 \rightarrow 1357 \rightarrow 1357 \rightarrow 13 \rightarrow 4 \quad (3)$$

$$1458 \rightarrow 2468 \rightarrow 1357 \rightarrow 13 \rightarrow 4 \quad (4)$$

$$1458 \rightarrow 1357 \rightarrow 2468 \rightarrow 13 \rightarrow 4 \quad (5)$$

$$1458 \rightarrow 2468 \rightarrow 2468 \rightarrow 13 \rightarrow 4 \quad (6)$$

The probabilities in the first two rounds are of each 2^{-16} and the probability in the third round is 2^{-24}, according to Tables 2 and 3. The expanded view of this four-round truncated differential in the fourth round is

4. $[2v, 0, v, 0, 0, 0, 0, 0], [128, 0, 128, 0, 0, 0, 0, 0], [0, 0, 0, 128, 0, 0, 0, 0]$

This round has probability $2^{-15.7}$, which has been found by a direct computation. We concatenate the four-round truncated differential with the following one-round differential with the expanded view

5. $[0, 0, 0, 128, 0, 0, 0, 0], [0, 0, 0, x, 0, 0, 0, 0], [2x, x, 2x, x, 2x, x, 2x, x]$,

where the value of x is odd. This differential has probability 1, since an input difference 128 to the exponentiation permutation always yields an odd output difference [7]. Therefore we obtain a 5-round truncated differential with input difference $[a, 0, 0, b, c, 0, 0, d]$ and output difference $[2x, x, 2x, x, 2x, x, 2x, x]$ for odd x and with a probability of $2^{-69.7}$.

We can use structures of each 2^{32} plaintexts yielding 2^{63} pairs with the desired difference in the inputs. We need about 2^{70} pairs to get one right pair and therefore about 128 structures, a total of 2^{39} plaintexts. We can perform our analysis on each structure and thus the memory requirements are 2^{32} 64 bit quantities. In the following we will do the analysis for all 2^{70} pairs simultaneously.

In SAFER an output transformation is applied to the outputs of the last round to obtain the ciphertexts. This transformation consist of byte wise xor'ing and adding modulo 256 the last-round key. Therefore, right pairs for the above truncated differential will have the following form

$$[z_1,\ x,\ 2x,\ z_2,\ z_3,\ x,\ 2x,\ z_4], \tag{7}$$

where the z_i's are values we cannot predict exactly. The following lemma is easily proved.

Lemma 1. *Let $\tilde{z}$ and $\hat{z}$ be two bytes and let k be a key byte. The least significant bit of $z = \tilde{z} - \hat{z} \bmod 256$ and of $z' = (\tilde{z} \oplus k) - (\hat{z} \oplus k) \bmod 256$ are equal.*

Since it is known that x is odd, it follows from Lemma 1 that z_1 and z_3 must be even, and z_2 and z_4 must be odd.

The filtering of wrong pairs goes as follows. For every pair, let x' be the value of the difference of the second byte of the ciphertexts. Check if x' is odd, and if so, check if the difference in bytes 3, 6 and 7 have values $2x', x', 2x'$, respectively. This first filtering process discards all but one out of 2^{25} pairs. For all remaining 2^{45} pairs, check if the z_i's have the right parity. This second filtering process discards all but one out of 16 pairs, thus we are left with 2^{41} pairs. We know that the difference before the output transformation must be $[2x, x, 2x, x, 2x, x, 2x, x]$ for a pair to be a right pair. On average each of the remaining pairs will suggest two values of each of the bytes 1,4,5 and 8 of the last-round key, i.e. 16 values of a 32 bit subkey. For every pair and for all these 16 key values, one checks if the difference in the plaintexts yields a correct difference in the outputs after the first round. Since there are two possible sets of four bytes with non-zero values after the first round, every pair will suggest $16 \times 2^{-15} = 2^{-11}$ values on average of the four key bytes 1,4,5, and 8. Here we used the fact that the round key byte i, $1 \leq i \leq 8$, in each round is derived from the same key byte. Totally, the 2^{41} pairs will suggest 2^{30} values of four bytes of the key. Thus, an exhaustive search at this point for the key can be done in time about $1/2 \times 2^{30} \times 2^{32} = 2^{61}$.

The time and space requirements of the filtering processes above can be made small. One method is the following, proposed by an anonymous referee. Let the ciphertexts be denoted $(c_1, ..., c_8)$. Hash each ciphertext to $(c_3 - 2 * c_2, c_6 - c_2, c_7 - 2 * c_2)$. The ciphertexts with the same such hash value will be candidates for a right pair after the first filtering process. The second filtering process can be done at the same time.

By repeating the attack several times the complexity can be decreased considerably. The basic attack described above suggests 2^{30} values of 32 bits of the key. The differential we use has probability about 2^{-70}, so by generating 2^{70} pairs one gets one right pair with probability 0.63. Thus the right key value is suggested with probability 0.63 and a wrong key value is suggested with probability $2^{30}/2^{32} = 0.25$. We keep a counter for every possible value of the 32-bit key and increment the respective counter for every suggested value of the key. Let T be the number of times we repeat the above basic attack. Let $X(T)$ be a random variable counting the number of times the right key is suggested and let

$Y(T)$ be a random variable counting the number of times any other value of the key is suggested in T basic attacks. From the above $E(X(T)) = T \times 0.63$ and $E(Y(T)) = T \times 0.25$. By assuming that the $X(T)$ and $Y(T)$ are independent and that the suggested wrong values of the key are uniformly distributed, one can approximate the probability that $Y(T)$ takes on a greater value than $X(T)$ after T basic attacks, i.e. $Pr(X(T) < Y(T))$. By the Central Limit Theorem [3], $Pr(X(64) < Y(64)) \simeq 2^{-19}$ and $Pr(X(128) < Y(128)) < 2^{-32}$. Thus, by repeating the attack 64 times using totally 2^{45} plaintexts, the right key value will be among the $2^{32} \times 2^{-19} = 2^{13}$ most suggested values with a high probability. To increase the probability of success, we choose the 2^{14} most suggested values of the key and do an exhaustive search for the remaining 32 key bits for every one of these values using a few of the obtained plaintext- ciphertext pairs, thus totally one needs to do about 2^{46} encryptions. Every counter can be implemented as one byte, thus the storage needed for the counters is only 1/8 one the storage needed for the plaintexts. Another possibility is to repeat the attack 128 times using totally 2^{46} plaintexts. The right key value will be among the first few most suggested values with a high probability. Taking the 8 most suggested values and searching exhaustively for the remaining 32 bits, the time complexity of the attack is about 2^{35}. We summarize the complexities of our attacks for SAFER with 5 rounds in Table 1.

Rounds	Time	Plaintexts	Storage
5	2^{61}	2^{39}	2^{32}
5	2^{46}	2^{45}	2^{32}
5	2^{35}	2^{46}	2^{32}

Table 1. Complexities of the differential attack on SAFER with 5 rounds. Time units are encryptions with SAFER. Storage units are 64 bits.

In the above attack we used the four round truncated differential 1458 $\rightarrow$ 4 with probability $2^{-69.7}$. There are many other differentials that can be used in variants of the above attacks, which the reader can verify by taking a closer look at Tables 2 and 3.

4.2 6-round SAFER

For SAFER with 6 rounds there is a similar truncated differential as the one above for SAFER with 4 and 5 rounds. It has input difference $[a, 0, 0, b, c, 0, 0, d]$ and output difference $[2x, x, 2x, x, 2x, x, 2x, x]$ after 6 rounds with a probability of $2^{-81.8}$. To get a right pair, one needs about $2^{50.8}$ chosen plaintexts. However, we have not been able to find a method to filter out enough wrong pairs in order to do a successful attack on SAFER with 6 rounds. Also, there are truncated differentials predicting the exact values of four bytes after 6 rounds with similar probabilities. As an example, the 6-round truncated differential with input difference $[a, b, c, d, 0, 0, 0, 0]$ and output difference $[x, y, z, w, 0, 0, 0, 0]$ has a probability of $2^{-83.8}$. This truncated differential contains more than 4000 differentials. To get a right pair, one needs about $2^{52.8}$ chosen plaintexts. However, the number of wrong pairs is too high to do a successful differential attack.

4.3 SAFER K-128, SAFER SK-64, and SAFER SK-128

The above attack for SAFER with 5 rounds is applicable to SAFER K-128 also. The filtering of wrong pairs and the procedure of getting 16 suggested key values in the last round are the same. The suggested key values in the first round will give us candidates only for the bytes in the first round key, since the addition modulo 256 of the second round key will be invariant because of the notion of difference used. But since the first and the last round keys depend only on the same 64 bits of the original key, we will find 64 bits of the 128 bit key by the above attack.

The truncated differential used above in our attack on SAFER with 5 rounds was chosen to minimize the number of counters for key candidates of a 32 bit subkey. For SAFER SK-64 (and SAFER SK-128) the four key bytes in positions 1, 4, 5 and 8 in the round keys will depend on different bytes of the key from round to round. Therefore the above analysis is not directly applicable to SAFER SK-64. However, it is clear that the first part of the attack with time complexity 2^{61} is applicable. The 2^{41} non-discarded pairs will suggest 16 values of round key bytes in positions 1, 4, 5 and 8 in the last round. These bytes correspond to bytes no. 2, 5, 6 and 9 in the original key, where byte 9 is the parity byte [4]. For every one of these 16 values, the check in the first round of the differentials will give us about 2^9 values of the key bytes 1, 4, 5, and 8 of the original key. Thus, we get suggested values of key bytes 1, 2, 4, 5, 6, 8 and 9, and totally about $2^{41} \times 16 \times 2^9 = 2^{54}$ possible values for the 56 bit key. The remaining 8 bits can be found exhaustively.

It is infeasible to keep a counter for each 56 bit key and repeat this attack, as we did for SAFER. But simply trying all possible candidates is possible and an exhaustive search for the key at this point would require about $1/2 \times 2^{62} = 2^{61}$ operations. We leave it so far as an open problem to find other differentials to improve our attack on SAFER versions with the new key schedule of [4]. One idea is to use several differentials in parallel attacks, for example using the following, $1357 \rightarrow 4$, $2468 \rightarrow 4$ and $2367 \rightarrow 4$, all three with probability $2^{-69.7}$.

5 Concluding remarks

We considered truncated differentials for 5-round SAFER and established a differential attack, which finds the secret key in time much faster than exhaustive search. The attack is independent of the S-boxes used in SAFER and needs only a small amount of chosen plaintext compared to conventional differential attacks which illustrates the importance of truncated differentials.

The recent change in the key schedule of SAFER will complicate our attacks, but not prevent them in a significant way. The main property that makes our truncated differentials possible is the PHT transformation, not the key schedule. However, for SAFER with more than 5 rounds our method of filtering out wrong pairs is not efficient enough to do a successful differential attack. We encourage the reader to improve our methods. Though it might be possible to improve our methods to attack SAFER versions with 6 rounds, we strongly believe that SAFER versions with 8 rounds, as now recommended, or more rounds are invulnerable to our new attack.

6 Acknowledgments

We would like to thank Serge Vaudenay for helpful discussions and for the LaTeX of the SAFER-graph, Figure 1, and Eli Biham and an anonymous referee for valuable comments.

References

1. E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer Verlag, 1993.
2. C. Harpes, G.G. Kramer, and J.L. Massey. A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In L. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology - Eurocrypt'95, LNCS 921*, pages 24–38. Springer Verlag, 1995.
3. Hoel, Port, and Stone. *Introduction to Probability Theory.* Houghton Mifflin Company, 1979.
4. L.R. Knudsen. A key-schedule weakness in SAFER K-64. In D. Coppersmith, editor, *Advances in Cryptology - CRYPTO'95, LNCS 963*, pages 274–286. Springer Verlag, 1995.
5. L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption, LNCS 1008*, pages 196–211, Springer Verlag, 1995.
6. J.L. Massey. SAFER K-64: A byte-oriented block-ciphering algorithm. In *Fast Software Encryption - Proc. Cambridge Security Workshop, Cambridge, U.K., LNCS 809*, pages 1–17. Springer Verlag, 1994.
7. J.L. Massey. SAFER K-64: One year later. In B. Preneel, editor, *Fast Software Encryption, LNCS 1008*, pages 212–241, Springer Verlag, 1995.
8. S. Murphy. An analysis of SAFER. Private communication, 1995.
9. S. Vaudenay. On the need for multipermutations: Cryptanalysis of MD4 and SAFER. In B. Preneel, editor, *Fast Software Encryption, LNCS 1008*, pages 286–297, Springer Verlag, 1995.

A One round differentials of SAFER

Input	Output	Prob.	Input	Output	Prob.	Input	Output	Prob.	Input	Output	Prob.
2	68	8	3	48	8	4	2468	8	5	78	8
6	5678	8	7	3478	8	12	6	16	12	256	16
12	1256	8	12	3478	8	13	234	16	13	4	16
13	1234	8	13	5678	8	14	246	16	14	1278	8
14	1278	8	15	7	16	15	357	16	15	1357	8
15	2468	8	16	567	16	16	1458	8	17	347	16
17	1368	8	23	46	16	23	3456	8	24	24	16
24	1234	8	24	5678	8	25	67	16	25	2367	8
26	57	16	26	1357	8	26	2468	8	27	3467	16
28	1368	8	34	26	16	34	1256	8	34	3478	8
35	47	16	35	2457	8	36	4567	16	37	37	16
37	1357	8	37	2468	8	38	1458	8	46	2457	8
47	2367	8	48	1357	8	48	2468	8	56	56	16
56	1256	8	56	3478	8	57	34	16	57	1234	8
57	5678	8	58	1278	8	67	3456	8	68	1234	8
68	5678	8	78	1256	8	78	3478	8	123	78	24
123	3456	16	124	5678	16	125	48	24	127	38	24
134	3478	16	135	68	24	136	58	24	145	28	24
234	1278	16	234	28	24	246	68	24	256	58	24
347	48	24	357	38	24	567	78	24			

Table 2. One-round truncated differentials for SAFER with inputs different in less than four bytes. Probabilities are $(-\log_2)$.

Input	Output	Prob.	Input	Output	Prob.	Input	Output	Prob.	Input	Output	Prob.
1234	2	32	1234	12	24	1234	34	24	1234	56	24
1234	78	24	1234	1234	16	1234	1256	16	1234	3478	16
1234	5678	16	1256	5	32	1256	15	24	1256	26	24
1256	37	24	1256	48	24	1256	1256	16	1256	1357	16
1256	2468	16	1256	3478	16	1278	16	24	1278	25	24
1278	38	24	1278	47	24	1278	1256	16	1278	1368	16
1278	3478	16	1357	3	32	1357	13	24	1357	24	24
1357	57	24	1357	68	24	1357	1234	16	1357	1357	16
1357	2468	16	1357	5678	16	1368	14	24	1368	23	24
1368	58	24	1368	67	24	1368	1234	16	1368	1458	16
1368	5678	16	1458	17	24	1458	28	24	1458	35	24
1458	46	24	1458	1278	16	1458	1357	16	1458	2468	16
2367	17	24	2367	28	24	2367	35	24	2367	46	24
2367	1357	16	2367	2468	16	2367	3456	16	2457	14	24
2457	23	24	2457	58	24	2457	67	24	2457	1234	16
2457	2367	16	2457	5678	16	2468	13	24	2468	24	24
2468	57	24	2468	68	24	2468	1234	16	2468	1357	16
2468	2468	16	2468	5678	16	3456	16	24	3456	25	24
3456	38	24	3456	47	24	3456	1256	16	3456	2457	16
3456	3478	16	3478	15	24	3478	26	24	3478	37	24
3478	48	24	3478	1256	16	3478	1357	16	3478	2468	16
3478	3478	16	5678	12	24	5678	34	24	5678	56	24
5678	78	24	5678	1256	16	5678	3478	16	5678	1234	16
5678	5678	16									

Table 3. One-round truncated differentials for SAFER with inputs different in four bytes. Probabilities are $(-\log_2)$.