ECOLE NORMALE SUPERIEURE

Why SAFER K Changed Its Name

L. R. KNUDSEN

LIENS - 96 - 13

Département de Mathématiques et Informatique

CNRS URA 1327

Why SAFER K Changed Its Name

LIENS - 96 - 13

April 1996

Laboratoire d'Informatique de l'Ecole Normale Supérieure
45 rue d'Ulm 75230 PARIS Cedex 05

Tel : (33)(1) 44 32 00 00

Adresse électronique : ... @dmi.ens.fr

Why SAFER K Changed Its Name

Lars R. Knudsen
Laboratoire d'Informatique
École Normale Supérieure
Paris, France

Contents

Why SAFER K Changed Its Name

Abstract

In this paper we analyse the block cipher SAFER K. First we show a weakness in the key schedule, that has the effect that for almost every key K, there exists on the average 3 and half other keys K^* keys such that the encryptions of plaintexts different in one of 8 bytes yield ciphertexts also different in only one byte. Moreover, the difference in the keys, plaintexts, and ciphertexts are in the same byte. This enables us to do a related-key chosen plaintext attacks on SAFER K, which finds 8 bits of the key. Also, the security of SAFER K when used for in standard hashing modes, is greatly reduced, which is illustrated. Second, we propose a new key schedule for SAFER K avoiding these problems. Finally, we do differential cryptanalysis of SAFER K. We consider "truncated differentials" and apply them in an attack on 5-round SAFER K, which finds the secret key in time much faster than by an exhaustive search.

1 Introduction

In [11] a new encryption algorithm, SAFER K-64, hereafter denoted SAFER K, 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. Finally an output transformation is applied to produce the ciphertext. The suggested number of rounds is minimum 6 and maximum 10 [11, 12]. Also, Massey proposed a 128 bit key version called SAFER K-128 [12]. Strong evidence has been given that the scheme is secure against differential cryptanalysis after 5 rounds [12] and against linear cryptanalysis after 2 rounds [3]. In [17] Vaudenay showed that by replacing the S-boxes in SAFER K by random permutations, about 6% of the resulting ciphers can be broken faster than by exhaustive search.

In this paper we show a weakness in the key schedule of SAFER K and use our observations to establish related-key chosen plaintext attacks, which using from 2^{36} to 2^{39} chosen plaintexts finds 8 bits of the secret key with probabilities from 1 to 2^{-59} depending on certain circumstances of the attacks.

Furthermore, we show that for SAFER K with 6 rounds used in the standard hashing modes collisions can be found much faster than by a brute force attack. We found collisions of such hash functions in estimated time about 2^{23} encryptions when SAFER K is used as the underlying block cipher. This should be compared with a brute force collision attack, which requires about 2^{32} operations.

To avoid these problems we suggest a new key schedule for SAFER K making only small changes to the original one.

Also, in [14] Murhpy showed that there exists a projection on the input and output spaces of the round function in SAFER K which is independent on one quarter of the key. However, it is still unclear how to exploit Murphy's observations in an actual attack on SAFER K.

As a consequence of all this, Massey decided to adopt our stronger key schedule and to recommend that at least 8 rounds is used for SAFER K with a 64 bit key. The new cipher has been named SAFER SK-64. Massey also proposed a 128 bit key variant this version, namely SAFER SK-128.

Finally, we consider "truncated differentials" and apply them in an attack on 5-round SAFER K, 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 K. Another version of the attack needs totally about 2^{46} chosen plaintexts and runs in time similar to 2^{35} encryptions of 5-round SAFER K. This should be compared to the analysis made in [12], where a differential attack using conventional differentials on SAFER K 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 K and furthermore it needs only a small amount of chosen plaintext compared to conventional differential attacks [2] and illustrates the importance of truncated differentials.

This paper is organised as follows. First we give a short description of SAFER K and SAFER SK. In Sect. 3 we describe the weakness in the key schedule of SAFER K and show how to exploit this in a related-key chosen plaintext attack. In Sect. 4 we describe attacks on hash modes using SAFER K and give examples of collisions. In Sect. 5 we give different methods of how to improve SAFER K to avoid the problems described in the preceding sections and discuss the new key schedule used in the modified version of the algorithm, SAFER SK.

In Sect. 6 we consider truncated differentials of SAFER K and in Sect. 7 apply them in differential attacks. We give our concluding remarks in Sect. 8.

2 Description of SAFER K

SAFER K 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 [11]. 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

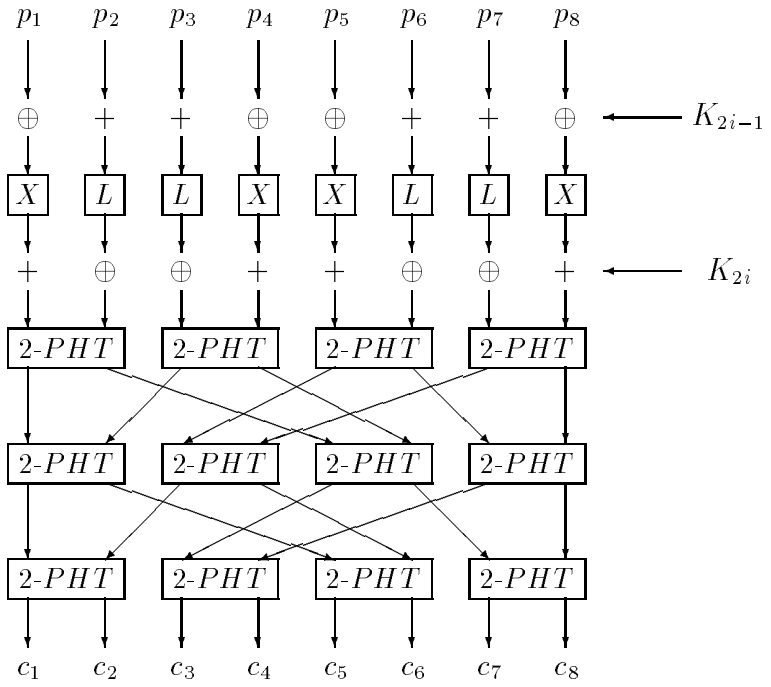

Figure 1: One round of SAFER K.

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, which using the cycle notation is $(1), (8), (253), (467)$, see also 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

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

2.1 The key schedule of SAFER K

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

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

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.

2.2 The key schedule of SAFER SK

The newly suggested SAFER SK varies from SAFER K 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.

$$\begin{aligned} K_{1,j} &= k_{1,j} \quad \text{for } j = 1, ..., 8 \\ k_{i,j} &= k_{i-1,j} <<< 3 \quad \text{for } j = 1, ..., 9 \\ K_{i,j} &= k_{i,(i+j-2 \bmod 9)+1} + bias[i, j] \bmod 256 \quad \text{for } j = 1, ..., 8 \end{aligned}$$

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

2.3 The 128 bit key schedules

The 128 bit key versions differ 64 bit version 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.

2.4 Some Properties of SAFER K

The following lemmas are used in this paper.

Lemma 1 *Let X be the exponentiation function of SAFER K and let a be any byte value. Then it holds that*

$$X(a) + X(a + 128) = 1 \bmod 256$$

Proof: The statement is proved as follows.

$$\begin{aligned} X(a) + X(a + 128) \bmod 256 &= (45^a + 45^{a+128} \bmod 257) \bmod 256 \\ &= (45^a \times (1 + 45^{128}) \bmod 257) \bmod 256 \\ &= (0 \bmod 257) \bmod 256 \end{aligned}$$

since $45^{128} = -1 \bmod 257$. And since both $X(a)$ and $X(a + 128)$ are in the range $[0, 256]$ and their sum is not zero, the statement follows. □

The mixed use of addition modulo 256 and exclusive-or operations in SAFER K was introduced to give the cipher *confusion* [11]. There is a simple and useful connection between the two operations when used on bytes, namely

Lemma 2 *Let a be a byte value. Then $a \oplus 128 = a + 128 \bmod 256$.*

Proof: Follows easily from the fact that the only possible carry bit of $a + 128$ disappears. □

A result similar to Lemma 2 is shown in [12].

3 Weakness in the Key Schedule

From the previous section it is seen that key byte j affects only S-box j directly in every round. Let $K = (k_1, ..., k_8)$ be an 8 byte key. Consider the first byte in the first round. A key byte is first xor'ed to the plaintext byte, the result is exponentiated and another key byte is added modulo 256, the ciphertext byte after one round is $X(y \oplus K_{1,1}) + K_{2,1}$, where $K_{1,1}, K_{2,1}$ are derived from k_1. While it is true that this is a permutation of the plaintext byte to the ciphertext byte for a fixed key, it is not a permutation of the key byte to the ciphertext byte for a fixed plaintext. That is, there exist keys $K^*_{1,1}, K^*_{2,1}$ derived from k^*_1, such that

$$X(y \oplus K_{1,1}) + K_{2,1} = X(y \oplus K^*_{1,1}) + K^*_{2,1} \tag{1}$$

for some inputs y.

Let $K^* = (k^*_1, ..., k^*_8)$ be an 8 byte key different from K in only one byte, say byte no. 1. Then if k_1 and k^*_1 encrypt some of the 256 possible inputs to S-box 1 in every round the same way, obviously K and K^* encrypt some 64 bit plaintexts over 6 rounds the same way.

If, say, n inputs to an S-box in the s'th round are encrypted the same way by two such keys we will say that the keys encrypt equally with probability $p_s = \frac{n}{256}$. Also we will call two such keys *related*. Consider S-box 1, K and K^* again. If a byte y is evaluated the same way with the two keys in S-box 1, i.e. such that (1) holds, then so is the byte $\tilde{y} = y \oplus K_{1,1} \oplus K^*_{1,1} \oplus 128$. This follows from Lemma 1 and 2. Since L is the inverse of X, a similar property holds for the logarithmic S-boxes. Therefore n is always a multiple of 2. The probability that a 64 bit plaintext encrypts into the same ciphertext using two such keys is

$$\prod_{s=1}^{6} p_s \;\geq\; 2^6/2^{48} = 2^{-42}, \tag{2}$$

and the number of plaintexts is $Pl = 2^{64} \times \prod_{s=1}^{6} p_s \geq 2^{22}$. Here we have tacitly assumed that the p_i's are independent. This is not the case, however our experimental results have shown that the product (2) of the round probabilities is a good approximation for SAFER K with 6 rounds. Since this phenomenon is isolated to one S-box we can easily do an exhaustive

Plaintext	Keys	Ciphertexts
8a 2c 62 a2 a2 81 c1 8c	e0 81 01 85 eb 3b 48 76	ca dd fc f6 30 ac 71 38
8a 2c 62 a2 a2 81 c1 8c	e0 81 01 85 eb 3b 48 bc	ca dd fc f6 30 ac 71 5c
50 1c 7a 44 39 63 f7 8c	e0 81 01 85 eb 3b 48 76	6a 7d db 51 44 89 5a f7
50 1c 7a 44 39 63 f7 8c	e0 81 01 85 eb 3b 48 bc	6a 7d db 51 44 89 5a 93

Table 1: Pseudo key-collisions for SAFER K (hex notation).

search for all such pairs of keys. We found that for two keys different only in the third byte with the values 132 and 173 respectively, $\prod_{s=1}^{6} p_s = \frac{6912}{2^{48}} \simeq 2^{-35}$ and $Pl \simeq 1.7 \times 2^{28}$. Note that since the only requirement we make is that the two keys have certain values in the third bytes, $Pl \simeq 1.7 \times 2^{28}$ for 2^{56} pair of keys. For another 3×2^{56} pairs of keys $Pl \simeq 1.13 \times 2^{28}$. How do we determine for how many keys there exist another key which encrypts from 2^{22} to about 2^{28} plaintexts the same way? Take a key K. Consider all $2^8 - 1$ keys K^* different from K only in byte 1. If none of them are related to K, choose keys K^* different from K only in byte 2 and so on. Again we can do an exhaustive search for all S-boxes isolated. The total number of keys for which there are no such other keys different in only one byte is about 2^{40}. For many keys K there exists more than one related key, on average about 2 related keys, and in some cases there are as many as 9 keys related to K.

In the search for the plaintext/ciphertext pairs that coincide for two keys it is not necessary to do two full 6 rounds of encryptions. One can start the encryptions in the second round with the inputs to this round such that the ciphertexts after the first two rounds of encryption are the same. This can be done easily by pre-computing two small tables. Assume that the two keys differ in the first byte only. For the 256 possible values of the text output of the first S-box in the first round, store in a table the values for which the two keys decrypt to equal plaintexts. For the 256 possible values of the text input to the first S-box in the second round, store in a table the values for which the two keys encrypt to equal values. By pairing the values in the two tables and determining which PHT inputs whose first byte equals the first byte of a pair give a PHT output whose first byte equals the second byte of this pair, one can compute all the 64 bit inputs to the second round, such that the two keys encrypt equally in both the first and the second round.

After every round of encryption one checks whether the encryptions are equal. In most trials only 1 round of encryption is needed for every plaintext in a pair. Therefore one needs only to do about $\frac{1}{6} \times 2/\prod_{i=3}^{6} p_i$ encryptions, which is 2^{22} in the optimal cases. Again we note that the output transformation, which consists of xor'ing, respectively adding modulo 256, the key K_{2r+1} makes the above ciphertexts differ in one byte, exactly the byte for which the keys differ. As illustrations we list in Fig. 1 two such examples. The first such *pseudo-collision* was found in time 2^{22}, the second in time $2^{22.1}$. We summarize our results.

Theorem 1 *For all but 2^{40} keys K in SAFER K, there exists at least one and on average two keys, K^*, different from K in one byte, say byte b_k, such that K and K^* encrypt from 2^{22} to about 2^{29} plaintexts the same way in 6 rounds. The output transformation of SAFER K makes the ciphertexts differ in one byte, byte b_k. The related keys can be found*

easily by exhaustive search over a single 8 bit S-box in 6 rounds. Given two related keys one such plaintext (and the two ciphertexts) can be found in time from about 2^{22} to 2^{28} encryptions.

From the above discussion also the following result follows.

Theorem 2 *For all but 2^{17} keys K in SAFER K, there exists at least one and on average 3.5 keys, K^*, different from K in one byte, say byte b_k, such that K and K^* encrypt from 2^{29} to about 2^{35} pairs of plaintexts, P, P^*, different in only byte b_k the same way in 6 rounds. The output transformation of SAFER K makes the ciphertexts differ in one byte, byte b_k.*

To find such "collisions", one can use the same method as described above for the result of Theorem 1, but this time start the search in the third rounds, such that the encryption in the second and third rounds are equal. Once two ciphertexts different in only byte b_k are found, the ciphertexts after one round are decrypted into two plaintexts different in only byte b_k. Examples of collisions from Theorem 2 are given in the section about collisions of hash functions. We can use Theorem 2 to establish a related-key attack on SAFER K.

3.1 A Related-key Chosen Plaintext Attack

In [5, 6, 1] new attacks based on related keys were introduced. In this section we apply the principles of these attacks and introduce a chosen plaintext attack on SAFER K. Assume we have access to two oracles, one encrypting plaintexts with a key K, the other encrypting plaintexts with a key K^*, such that K and K^* are related, i.e. encrypt a non-negligible fraction of all plaintexts the same way. Assume without loss of generality that the keys differ only in byte b_1. Consider the following attack

- Choose the values of the bytes b_2 to b_8 at random.
- Get the 256 encryptions $\{C_i\}$ of the plaintexts $b_1, b_2, ..., b_8$ for all values of b_1 encrypted under the first key.
- Get the 256 encryptions $\{C_j\}$ of the plaintexts $b_1, b_2, ..., b_8$ for all values of b_1 encrypted under the second key.
- Sort the ciphertexts just received and check, if any ciphertext in $\{C_i\}$ differs from any ciphertext in $\{C_j\}$ only in byte b_1. If a match is found the two ciphertexts are output.

If ciphertexts are output in the last step of the above attack, we search exhaustively for two 8 bit keys k and $k*$ for which the encryptions of the bytes b_1 for the two corresponding plaintexts yields equal outputs after one round. For these key bytes we check if the xor of the byte b_1 for the two ciphertexts is the value of the xor of the last-round key bytes induced by k and $k*$. If this is the case we have found 8 bits of the secret key with a high

# Plaintexts	Probability	Conditions
2^{36}	0.63×1	Two related keys
2^{36}	$0.63 \times 1/73$	The two keys differ in one known byte position.
2^{39}	$0.63 \times 1/73$	The two keys differ in one unknown byte position.
2^{39}	0.63×2^{-59}	The two keys are randomly chosen.

Table 2: Related-key chosen plaintext attacks on SAFER K finding one byte of the key. (Worst case considerations.)

probability. It could happen by accident that two ciphertext blocks are different only in one byte without the property that the encryptions after each of the 6 rounds are equal. But clearly that would happen only with negligible probability.

The attack is repeated until the last step of the algorithm outputs two ciphertexts. Note that since we choose all 256 plaintexts different in one byte, we can consider 2^{16} pairs of plaintexts, consisting of one plaintext encrypted under one key and another plaintext encrypted under the second key. It follows that there are 256 pairs of plaintexts encrypted the same way in the first round. According to Theorem 2 two related keys encrypt from 2^{29} to 2^{35} pairs of plaintexts equally in 6 rounds, and therefore the above algorithm needs to be repeated at most about 2^{27} times, in the optimal cases only 2^{21} times. The number of chosen plaintexts needed in the worst cases is about $2 \times 2^8 \times 2^{27} = 2^{36}$. The probability of success is about 0.63. The attack can be extended to the case where the attacker has no knowledge of the byte for which the keys differ. The above attack is simply repeated for all 8 bytes requiring a total of 2^{39} chosen plaintexts. If the two keys are chosen at random different in only one byte, the attack succeeds with a probability of $\frac{3.5}{256}$, according to Theorem 2. Two randomly chosen 8 byte keys will be different in only one byte with probability $8 \times \frac{255}{256} \times 2^{-56} \simeq 2^{-53}$. Therefore, if all of the 8 bytes of the two keys are chosen at random, the attack succeeds with a probability of $2^{-53} \times \frac{3.5}{256} \simeq 2^{-59}$. We summarize our results in Table 2 for SAFER K with the recommended 6 rounds. We note that the complexities given are worst case considerations. The factor 0.63 in the probabilities can be increased by using more chosen plaintexts. In Table 3 we give the complexities for similar related-key attacks on SAFER K with 4 and 8 rounds. Our attacks may seem unrealistic. But imagine Alice and Bob are sending many messages to each other every day. Alice and Bob have been acting in many cryptographic papers, so they know that the key should be changed often. So, they change the key every day, but to save computations only in one byte, so that all the bytes in the key are changed after eight days. Nowhere in the literature have they found evidence that this should be dangerous. Using SAFER K it will be. Eve hasn't appeared in as many papers as Alice and Bob, but is smart enough to trick one of the parties into encrypting many chosen plaintexts every day. Eve finds 8 bits of the secret key with probability $\frac{3.5}{256}$ every day, except the first day, using at most 2^{39} chosen plaintexts. We assume here that the time to sort and compare ciphertexts is

4 rounds		8 rounds		
Pl.texts	Prob.	Pl.texts	Prob.	Conditions
2^{22}	0.63×1	2^{50}	0.63×1	Two related keys.
2^{22}	$0.63 \times 1/14$	2^{50}	$0.63 \times 1/256$	The two keys differ in one known byte position.
2^{25}	$0.63 \times 1/14$	2^{53}	$0.63 \times 1/256$	The two keys differ in one unknown byte position.
2^{25}	0.63×2^{-57}	2^{53}	0.63×2^{-61}	The two keys are randomly chosen.

Table 3: Related-key chosen plaintext attacks on SAFER K with four and eight rounds finding one byte of the key. (Worst case considerations.)

negligible compared to the time of getting the many encryptions. After 73 days Eve has asked for about 2^{45} chosen plaintexts and with a probability 0.63 found at least 8 key bits. The number of chosen plaintexts can be reduced to 2^{42}, if Eve can predict which byte of the secret key is changed from day to day. Similar attacks on SAFER K with a reduced number of rounds will have much lower complexities.

Recently, Wagner [18] improved our related key attacks.

3.2 The Rotations and Bias Additions

In this section we consider the rotations and bias additions used in the key schedule of SAFER K. In [11] it is argued that the bias additions prevent weak keys. Moreover, by letting out the key biases, for any key K there exists another key K^*, such that the first 5 rounds of the encryption function induced by K are the same as the last 5 rounds of the encryption function induced by K^*. This is not a desirable property as illustrated in [5, 6, 1]. We have found a reason to have byte rotations as well.

Lemma 3 *PHT has 256 fixed points.*

This result can be found by using Gauss-eliminations on the 8×8 matrix of PHT. In each fixed point every byte value is a multiple of 64. There are 16 fixed points where every byte value is either 0 or 128. They are given in Appendix A, Table 7. If one leaves out the key rotations, but keeps the addition of the biases then these 16 fixed points for PHT are "linear structures" for SAFER K with any number of rounds in the following way. Let $a_1, ..., a_{16}$ be the fixed points from Table 7. Let $E(K, P) = C$ be the encrypted value of plaintext P using key K, then

$$E(K, P) = C \;\Rightarrow\; E(K + a_i, P + a_i) = C.$$

where '+' is byte-wise addition modulo 256. Thus, an exhaustive search for the key could be reduced by a factor of 16 using 16 chosen plaintexts. The 16 fixed points are the only such linear structures. Fixed points with entries of values 64 or 192 are affected/destroyed by the group operation changes exclusive-or/addition mod 256, but the values 0 and 128

are not, which follows from Lemma 2. The above illustrates that SAFER K needs both key rotations and bias additions in the key schedule.

4 Collision of Hash Functions

Often a block cipher is used as building block in hash functions. A hash function for which the hash code is of the same size as the block cipher is called a *single block length hash function*. In these hash functions the message blocks are hashed in a number of rounds, each round requiring one encryption of the underlying block cipher. There are essentially 12 secure single block length hash functions, which by a linear transformation of the inputs to one round of the hash function can be transformed into only 2 different schemes [15, 16]:

$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1} \quad (3)$$

$$H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1} \oplus M_i \quad (4)$$

The first scheme is known as the Davies-Meyer scheme. These schemes are believed to be secure, in the sense that, if the underlying block cipher has no weaknesses, free-start pre-image attacks and free-start collision attacks have time complexities 2^m and $2^{m/2}$ encryptions, respectively, of the underlying m-bit block cipher [10, 15]. In a free-start attack the attacker is free to choose the initial values. Using SAFER K as the underlying block cipher it is possible to find both free-start and fixed-start collisions with a complexity of much less than the brute force method of 2^{32} operations.

Also, we note that the attacks to follow will be applicable to many double block length hash functions based on a block cipher, since in free-start attacks it is possible to attack the two blocks independently. In the next section we show how to find free-start collisions for the schemes (3) and (4).

4.1 Free-start Collisions

In this section we exploit the phenomenon of Theorem 2. In the attacks to follow we choose two plaintexts different only in the byte for which both the keys differ. We hope in this way to obtain plain- and ciphertexts and keys, such that

$$\begin{aligned} E_{K_1}(P_1) \oplus P_1 &= E_{K_2}(P_2) \oplus P_2 \quad \text{and/or} \\ E_{K_1}(P_1) \oplus P_1 \oplus K_1 &= E_{K_2}(P_2) \oplus P_2 \oplus K_2 \end{aligned}$$

depending on the type of hash function we are attacking.

We can speed up this search by choosing the inputs of SAFER K to the third round, such that the keys encrypt equally in the second and third rounds. For (3), when we find two ciphertexts different in only one byte, we calculate the plaintexts and check for a collision. In the optimal cases these collisions can be found in estimated time about $2^{22.8}$ encryptions of SAFER K. In Table 4 we give examples of such collisions for hash functions

Initial value (pl. text)	Message (key)	Hash code
6e 32 68 46 c8 fd f1 a9	6f 2d 73 46 e1 2f 62 45	e5 12 8b 4d 3d 58 c2 18
6e 32 68 46 c8 fd f1 9c	6f 2d 73 46 e1 2f 62 f7	e5 12 8b 4d 3d 58 c2 18
f4 b1 a3 27 0b ed 78 a9	57 f5 9b 4e 49 77 0a 45	54 43 57 c4 be f9 88 c9
f4 b1 a3 27 0b ed 78 9c	57 f5 9b 4e 49 77 0a f7	54 43 57 c4 be f9 88 c9

Table 4: Free-start collisions for hash functions of type (3) with SAFER K.

Initial value (pl. text)	Message (key)	Hash code
ff 4e 79 3f c3 4f 52 5b	6d e6 02 f2 54 f0 59 a8	a7 a9 3e 8c 23 30 c3 b4
ff 4e 79 3f c3 4f 52 5b	e5 e6 02 f2 54 f0 59 a8	a7 a9 3e 8c 23 30 c3 b4
ff 9d e5 f5 c1 bc eb 71	6d 9b 13 2f 4d f5 7a b5	11 47 f9 f4 53 c8 e3 17
ff 9d e5 f5 c1 bc eb 71	e5 9b 13 2f 4d f5 7a b5	11 47 f9 f4 53 c8 e3 17

Table 5: Fixed-start collisions for hash functions of type (4) with SAFER K.

of type (3). The first collision was found in time $2^{20.6}$ encryptions, the second collision in time $2^{19.3}$ encryptions.

Similarly, it is possible to find free-start collisions for hash functions of type (4). We found such collisions in time about 2^{22}. In the next section we give examples of collisions for hash functions of type (4) with a fixed start.

4.2 Fixed-start Collisions

Although the collisions found in the last section are considered hard to find, if the underlying block cipher has no weaknesses, it is interesting to find collisions also for a fixed start, i.e. where the plaintexts of SAFER K are fixed. Using our observations about SAFER K this cannot be done for the hash round function (3), since if the plaintexts are equal for two related keys the hash value of (3) will always be different. However, it is possible to find collisions if we consider two rounds of the hash function. Assume H_0 is a fixed initial value. Using the related key properties described earlier in this paper one finds M_1 and M_1', such that $H_1 = E_{M_1}(H_0) \oplus H_0$ and $H_1' = E_{M_1'}(H_0) \oplus H_0$ differ in one byte. Then use the related key properties once again in the second round and find M_2 and M_2', such that $H_2 = E_{M_2}(H_1) \oplus H_1$ equals $H_2' = E_{M_2'}(H_1') \oplus H_1'$. We did not implement this attack. For the hash functions (4) it is possible to find fixed start collisions for the hash round function. For our pseudo-collisions for SAFER K, see Table 1, the ciphertexts and keys differ in the same byte. Therefore when both the plaintexts and the keys are fed forward in the hash mode, we can obtain collisions. The difference in the ciphertexts of Table 1 is equal to the difference in the last-round keys, which is not necessarily the difference in the keys themselves. Therefore for this attack to work we must use pairs of keys for which the byte differences in the keys are equal to the byte differences in the last-round keys of the keys. An exhaustive search reveals many pairs of keys with this property. Two keys different only in the fifth byte with values 9 and 129 respectively, encrypt about 2^{28} plaintexts in

the same way after 6 rounds. By using similar techniques as for free-start collisions one can show that a collision can be found in expected time about 2^{22} encryptions. In Table 5 we list such collisions. The first collision was found in time $2^{22.3}$ encryptions, the second collision in time $2^{20.0}$ encryptions. Many of our collision implementations ran faster than expected, which may be due to the fact that probabilities in (2) are not independent as assumed.

5 Improvements of SAFER K

In this section we suggest modifications of SAFER K, such that the above attacks cannot be effected. An obvious and immediate way is to increase the number of rounds.

5.1 An Increased Number of Rounds

In SAFER K with 8 rounds there are still many pairs of keys encrypting some plaintexts the same way. In the optimal case a pair of keys encrypt 1.5×2^{14} plaintexts into the same ciphertexts after 8 rounds of encryption using our method. Also, related keys encrypt pairs of plaintexts into equal ciphertexts with non-negligible probability. Therefore, a key-related attack is possible for SAFER K with 8 rounds, the complexity is given in Table 3. Collisions for hash modes using SAFER K with 8 rounds cannot be found faster than the time of 2^{32} encryptions.

In the optimal case for SAFER K with 10 rounds a pair of keys encrypt equally for all 10 rounds with probability of only 2^{-66} using our method. Since there are only 2^{64} different plaintexts there are no keys with the above phenomenon, thus the key-related attack is not possible and collisions based on related keys cannot be found faster than by brute force attack.

5.2 New Key Schedule for SAFER K

Another and in our taste better solution is to change the key schedule. The discoveries in this paper come from the fact that a key is applied to the text input before and just after the S-box, thus enabling collisions considering one byte isolated in every round. One way to hinder this is, paradoxically, to remove the second xor/addition of the key in every round or just in one of the middle rounds. To find collisions similar to the ones we've found would now require an incorporation of the PHT-transformation. That seems very unlikely to succeed. But, the fact that a one byte key is connected to the same S-box in every round seems dangerous and unnecessary.

We next discuss the modified key schedule for SAFER K already described in Sect. 2.2. As can be seen, there is a circular shift of the nine key bytes. In that way the 8 user-selected key bytes $k_1, ..., k_8$ are connected to different S-boxes from round to round. The parity byte is introduced to provide an avalanche effect in the key schedule. The new key schedule ensures that the round keys of two different keys are always different in two bytes

in some rounds and in one byte in the remaining rounds. For instance, in SAFER K with 6 rounds, two keys will be different in two bytes in 9 out of the 13 round keys. In SAFER K with 8 rounds, this will be the case in 13 out of the 17 round keys. Thus, our method of finding key-collisions is no longer applicable. Also, note that if the key is chosen uniformly at random, any round key is uniformly random.

6 Differentials of SAFER K

In [12] strong evidence was given that SAFER K is secure against differential cryptanalysis. It was argued that a 5-round differential for SAFER K 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 [12]. We will use the notation of "expanded views" from [12] 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. To cope with the mixed use of addition modulo 256 and the exclusive-or, Massey introduced *quasi-differentials*, where the notions of difference are different in input and output [12].

In our attacks, we can avoid doing that and throughout the rest of this paper, a *difference* of two bytes (a, b) is defined as

$$a - b \bmod 256.$$

Definition 1 *([8]) A differential that predicts only parts of an n bit value is called a* truncated differential. *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 [12] 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

$$[a, b, c, d, 0, 0, 0, 0], [e, f, -e, -f, 0, 0, 0, 0], [2g, g, 2h, h, 0, 0, 0, 0], \tag{5}$$

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 (5). 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}, the probability taken over all 2^{16} possible keys. More

formally, let $Pr(a \rightarrow e)$ denote that an input difference a to an S-box yields an output difference e etc., then

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

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

$$[0,0,0,0,0,0,a,b],[0,0,0,0,0,0,0,0,0,e,-e],[e,e,0,0,e,e,0,0]. \tag{6}$$

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 (5) and (6) 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 (5) and, similarly, $78 \rightarrow 1256$ for the differential (6). In Appendix B Tables 8 and 9, many such differentials are listed. E.g. the differential (5) can be found in Table 9 as Input: 1234, Output: 1234, Prob. 16.

As we will show now, one can concatenate the one-round differentials of Tables 8 and 9. 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 $8 \times 2^{-48} = 2^{-45}$.

7 Differential attacks on SAFER K

In this section we consider differential attacks using truncated differentials for SAFER K. Consider 3-round SAFER K 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 $1/p$ chosen plaintext pairs to get one right pair [2]. Using the above truncated differential for SAFER K 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 a 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 in this example the output transformation of SAFER K should be 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 K 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 K. Instead of working on bytes we let SAFER K work on nibbles (4 bits), i.e. a 32-bit block cipher with a 32-bit key, called SAFER K(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 K(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 K, 64 bits, using truncated differentials.

7.1 A differential attack on 5-round SAFER K

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

$$1458 \rightarrow 1357 \rightarrow 1357 \rightarrow 13 \rightarrow 4 \tag{7}$$

$$1458 \rightarrow 2468 \rightarrow 1357 \rightarrow 13 \rightarrow 4 \tag{8}$$

$$1458 \rightarrow 1357 \rightarrow 2468 \rightarrow 13 \rightarrow 4 \tag{9}$$

$$1458 \rightarrow 2468 \rightarrow 2468 \rightarrow 13 \rightarrow 4 \tag{10}$$

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 8 and 9. 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 [12]. 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 K 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

$$[z_1,\ x,\ 2x,\ z_2,\ z_3,\ x,\ 2x,\ z_4], \tag{11}$$

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

Lemma 4 *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 4 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 $1/2 \times 2^{30} \times 2^{32} = 2^{61}$.

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 6: Complexities of the differential attack on SAFER K with 5 rounds. Time units are encryptions with SAFER K. Storage units are 64 bits.

The time and space requirements of the filtering processes above can be made small. One method is the following, proposed by an anonymous referee of [9]. 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 [4], $Pr(X(64) < Y(64)) \simeq 2^{-19}$ 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 K with 5 rounds in Table 6.

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 8 and 9.

7.2 Attacks on 6-round SAFER K

For SAFER K with 6 rounds there is a similar truncated differential as the one above for SAFER K 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 K 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.

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

The above attack for SAFER K 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 K 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 [7]. 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 K. 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 K versions with the new key schedule of [7]. 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 $2^{-69.7}$.

8 Conclusion

In this paper we analysed the 6 round block cipher SAFER K. We discovered a weakness in the key schedule and exploited it in related-key attacks and to find collisions for SAFER K in the standard hashing modes much faster than by brute-force. Our analysis together with Murphy's [14] analyses led Massey to adopt our proposed strengthened key schedule for SAFER K, yielding the new block cipher SAFER SK with a recommended minimum of 8 rounds.

We considered truncated differentials for 5-round SAFER K 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 K and needs only a small amount of chosen plaintext compared to conventional differential attacks which illustrates the importance of truncated differentials.

Our attacks are not directly applicable for SAFER SK, but they are not prevented in a significant way. The main property that makes our truncated differential attacks possible is the PHT transformation, not so much the key schedule. However, for SAFER K 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 K versions with 6 rounds, we strongly believe that SAFER SK with 8 rounds, as now recommended, or more rounds are invulnerable to all our attacks.

9 Acknowledgments

We would like to thank Jim Massey, Serge Vaudenay, and for helpful discussions Tom Berson, co-author of [9]. Also thank you Eli Biham, Carlo Harpes, Xuejia Lai, Torben Pedersen, and David Wagner for valuable comments and Serge also for the LaTeX of the SAFER-graph.

References

[1] E. Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.

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

[3] 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.

[4] Hoel, Port, and Stone. *Introduction to Probability Theory.* Houghton Mifflin Company, 1979.

[5] L.R. Knudsen. Cryptanalysis of LOKI'91. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology, AusCrypt 92, LNCS 718*, pages 196–208. Springer Verlag, 1993.

[6] L.R. Knudsen. *Block Ciphers – Analysis, Design and Applications.* PhD thesis, Aarhus University, Denmark, 1994.

[7] L.R. Knudsen. A key-schedule weakness in SAFER K-64. In *Advances in Cryptology - Proc. Crypto'95, LNCS 963*, pages 274–286. Springer Verlag, 1995.

[8] L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 196–211. Springer Verlag, 1995.

[9] L.R. Knudsen and T. Berson. Truncated differentials of SAFER. In Gollmann D., editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, pages 15–26. Springer Verlag, 1995.

[10] X. Lai. *On the Design and Security of Block Ciphers.* PhD thesis, ETH, Zürich, Switzerland, 1992.

[11] 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.

[12] J.L. Massey. SAFER K-64: One year later. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 212–241. Springer Verlag, 1995.

[13] J.L. Massey. Announcement of a STRENGTHENED KEY SCHEDULE for the cipher SAFER. Posted on the Usenet newsgroups "sci.crypt" and "sci.crypt.research", Sept. 11, 1995.

[14] S. Murphy. An analysis of SAFER. Private communication, 1995.

[15] B. Preneel. *Analysis and Design of Cryptographic Hash Functions.* PhD thesis, Katholieke Universiteit Leuven, January 1993.

[16] B. Preneel. Hash functions based on block ciphers: A synthetic approach. In D.R. Stinson, editor, *Advances in Cryptology - Proc. Crypto'93, LNCS 773*, pages 368–378. Springer Verlag, 1993.

[17] S. Vaudenay. On the need for multipermutations: Cryptanalysis of MD4 and SAFER. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 286–297. Springer Verlag, 1995.

[18] D. Wagner. Private communications, 1995.

A Some fixed points of the PHT

(	0	0	0	0	0	0	0	0)
(	0	0	0	0	128	128	0	0)
(	0	0	128	0	0	0	128	0)
(	0	0	128	0	128	128	128	0)
(	0	128	0	128	0	0	0	0)
(	0	128	0	128	128	128	0	0)
(	0	128	128	128	0	0	128	0)
(	0	128	128	128	128	128	128	0)
(	128	0	0	128	0	128	128	128)
(	128	0	0	128	128	0	128	128)
(	128	0	128	128	0	128	0	128)
(	128	0	128	128	128	0	0	128)
(	128	128	0	0	0	128	128	128)
(	128	128	0	0	128	0	128	128)
(	128	128	128	0	0	128	0	128)
(	128	128	128	0	128	0	0	128)

Table 7: The 16 fixed points for the PHT with only entries 0 and 128.

B One round differentials of SAFER

In	Out	Prob.	In	Out	Prob.	In	Out	Prob.	In	Out	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 8: One-round truncated differentials for SAFER K with inputs different in less than four bytes. Probabilities are $(-\log_2)$.

In	Out	Prob.	In	Out	Prob.	In	Out	Prob.	In	Out	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 9: One-round truncated differentials for SAFER K with inputs different in four bytes. Probabilities are $(-\log_2)$.