J. Cryptology (2000) 13: 417–436
DOI: 10.1007/s001450010004

Journal of
CRYPTOLOGY

A Detailed Analysis of SAFER K

Lars R. Knudsen
Department of Informatics,
University of Bergen,
Bergen, Norway

Communicated by Don Coppersmith

Received 21 December 1997 and revised 29 May 1998
Online publication 21 March 2000

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

Key words. Cryptanalysis, Block cipher, SAFER, Truncated differentials, Collisions.

1. Introduction

In [11] Massey proposed a new encryption algorithm, SAFER K-64, hereafter denoted SAFER K. Both the block size and the key size are 64. The algorithm is an iterated cipher where 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 a minimum of six and a maximum of ten [11], [12]. Also, Massey proposed a 128-bit key version called SAFER K-128 [12]. Strong evidence has been given that SAFER K is secure against differential cryptanalysis after five rounds [12] and against linear cryptanalysis after two rounds [3]. In [16] it was shown that by replacing the S-boxes in SAFER K by random permutations, about 6% of the resulting ciphers can be broken faster than by an exhaustive search.

In this paper we show a weakness in the key schedule of SAFER K and use our observations to establish a related-key chosen plaintext attack, which using 2^{36} chosen plaintexts finds eight bits of the secret key. The attack can be repeated to find more key bits. Furthermore, we show that for SAFER K with six rounds used in the standard hashing

modes collisions can be found much faster than by a brute-force attack. Collisions of such hash functions can be found in time about 2^{22} 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. These results appeared in [7]. Also, in [14] Murphy showed that there exists a projection on the input and output spaces of the round function in SAFER K which is independent of one-quarter of the key. As a consequence of all this, Massey decided to adopt our stronger key schedule and to recommend the use of eight rounds [13]. The modified cipher was named SAFER SK-64. Massey also proposed a 128-bit key variant of this version, namely, SAFER SK-128.

We consider *truncated differentials* and apply them in an attack on five-round SAFER K, the original version, which finds the secret key much faster than by exhaustive search. The attack uses a five-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 a total of about 2^{45} chosen plaintexts, and runs in time similar to 2^{37} encryptions of five-round SAFER K. This should be compared with the analysis made in [12], where a differential attack using conventional differentials on SAFER K with five rounds was estimated to require more computations than an exhaustive key attack and this illustrates the importance of truncated differentials. These results appeared in [9]. Finally, we show that there exists a structure for SAFER similar to a higher-order differential, which holds with probability 1 after two rounds of encryption. The weaknesses of SAFER K reported in this paper are not due to bad intrinsic properties of the S-boxes used in SAFER K and attacks similar to the ones described here would be possible for most S-boxes.

This paper is organized as follows. First we give a short description of SAFER K and SAFER SK. In Section 3 a weakness in the key schedule of SAFER K is described and how to exploit this in a related-key chosen plaintext attack is shown. The same weakness is used in Section 4 to find collisions for hash functions using SAFER K. In Section 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 Section 6 we consider truncated differentials of SAFER K and apply them in attacks, and give concluding remarks in Section 7.

2. Description of SAFER K

SAFER K is an r-round iterated cipher with block size and key size both of 64 bits and with only byte-operations. 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 each divided into 8 bytes and the first round key is exclusive-ored (exored), respectively added modulo 256, according to Fig. 1. The bytes are then processed using two permutations or S-boxes, $X(a) = 45^a \bmod 257$ if $a \neq 128$, with $X(128) = 0$, and the inverse of X, $L(a) = \log_{45}(a) \bmod 257$ for $a \neq 0$ and with $L(0) = 128$. The outputs of the S-boxes

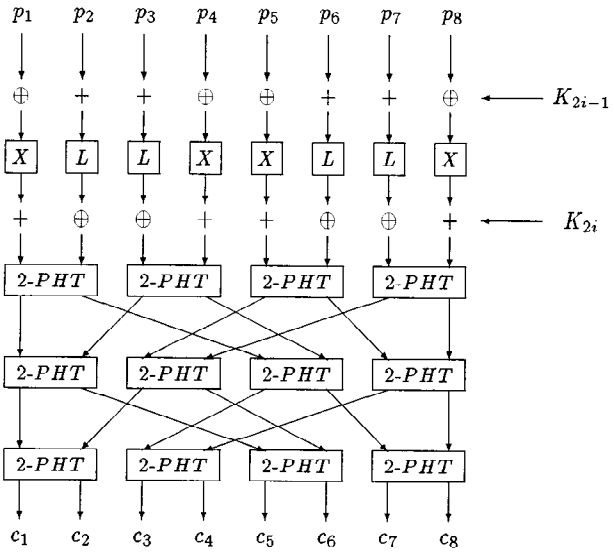

Fig. 1. One round of SAFER K.

are added modulo 256, respectively exored, to the second round key and finally the so-called *Pseudo-Hadamard Transformation* (PHT) is applied to produce the output of the round. The 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 computed modulo 256. Between two layers of 2-PHTs the bytes are permuted, a permutation which using cycle notation is (1)(8)(253)(467), see also Fig. 1. After the last round an output transformation, OT, is applied, which consists of exoring, respectively adding modulo 256, the last-round key. Let $\mathbf{o} = o_1, \ldots, o_8$ be the 8 bytes of the output after r rounds, and let $\mathbf{k} = k_1, \ldots, k_8$ be the 8 bytes of the last-round key. The ciphertext is defined as

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

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 $K = (k_{1,1}, \ldots, k_{1,8})$ 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 $j = 1, \ldots, 8$ 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 $i = 2, \ldots, 2r + 1$ and $j = 1, \ldots, 8$. "$<<< 3$" is a bitwise rotation three positions to the left and $bias[i, j] = X(X(9i + j))$, where X is the exponentiation permutation

described above. Let the input to the PHT be the 8-byte vector $X = [x_1, \ldots, x_8]$. The output vector y of the PHT can then be defined as $y = Mx$, where

$$M = \begin{bmatrix} 8 & 4 & 4 & 2 & 4 & 2 & 2 & 1 \\ 4 & 2 & 2 & 1 & 4 & 2 & 2 & 1 \\ 4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 & 2 & 1 & 1 \\ 4 & 2 & 4 & 2 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{1}$$

2.2. *The Key Schedule of SAFER SK*

SAFER SK varies from SAFER K in the suggested number of rounds, which is eight, and in the key schedule. Let $K = (k_{1,1}, \ldots, k_{1,8})$ 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:

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

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

2.3. *The* 128*-Bit Key Schedules*

The 128-bit key versions differ from the 64-bit versions 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-numbered round keys are taken from the first key schedule and the even-numbered round keys from the second key schedule. A 128-bit version is compatible with its 64-bit version, if the two 64-bit key halves input to the key schedule are set equal.

2.4. *Some Properties of SAFER K*

In this section we show two lemmas which 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. Consider first the case $a \neq 0, 128$:

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

since $45^{128} = -1 \bmod 257$. Since both $X(a)$ and $X(a+128)$ are in the range $1, \ldots, 255$ for $a \neq 0, 128$, it follows that $X(a) + X(a+128) = 257$. If $a = 0$ it follows that $X(a) + X(a+128) = X(0) + X(128) = 1$ and similarly for $a = 128$. □

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

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

3. Weakness in the Key Schedule of SAFER K

In SAFER K a key byte j affects only S-box j directly in every round. Let $K = (k_1, \ldots, k_8)$ be an 8-byte key and consider the first byte in the first round. First a key byte is exored to the plaintext byte, the result is exponentiated and another key byte is added modulo 256, that 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 from the plaintext byte to the ciphertext byte for a fixed key, it is not a permutation from the key byte to the ciphertext byte for a fixed plaintext. In other words, there exist keys $K^*_{1,1}, K^*_{2,1}$ derived from k^*_1, such that

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

for some inputs y. Let $K^* = (k^*_1, \ldots, k^*_8)$ be an 8-byte key different from K in only, say, the first byte. Then if k_1 and k^*_1 encrypt some of the 256 possible inputs y to (2) for S-box X in every round the same way, there is a good chance that K and K^* encrypt some 64-bit plaintexts over six rounds the same way. If, say, $n > 0$ inputs in the sth round are encrypted the same way we say that such *related* keys encrypt equally with probability $p_s = n/256$. Consider K and K^* again. If (2) holds for byte y, then it holds also for the byte $\tilde{y} = y \oplus K_{1,1} \oplus K^*_{1,1} \oplus 128$, which follows from Lemmas 1 and 2. Note that if $K_{1,1} \oplus K^*_{1,1} = 128$, $y = \tilde{y}$, but for such keys, (2) is never satisfied. To see this, note that in this case $X(y \oplus K_{1,1}) \oplus X(y \oplus K^*_{1,1})$ is odd, which follows from Lemma 1, but $K^*_{2,1} \oplus K_{2,1}$ is even, which is a consequence of the key schedule. Since L is the inverse of X, a similar property holds for the logarithmic S-boxes. Therefore n is always even. Since the round keys of SAFER are dependent, one cannot multiply the probabilities, p_s, for two consecutive rounds to get the total probability that two keys encrypt equally over two rounds. However, our experimental results have shown that this method is a good approximation to the real probability. Thus, the probability that a 64-bit plaintext encrypts into the same ciphertext after six rounds of encryption can be approximated as follows:

$$\prod_{s=1}^{6} p_s \;\geq\; \frac{2^6}{2^{48}} = 2^{-42}, \tag{3}$$

and the number of such plaintexts for a given pair of related keys is $Pl = 2^{64} \times \prod_{s=1}^{6} p_s \geq 2^{22}$. Since this phenomenon is isolated to one S-box it is easy to do an exhaustive search

for all such pairs of keys. As an example, for two keys different only in the third bytes with the values 132 and 173, respectively, $\prod_{s=1}^{6} p_s = 6912/2^{48} \simeq 2^{-35}$ and $Pl \simeq 2^{29}$. Note that since it is only required that the two keys have certain values in the third bytes, $Pl \simeq 2^{29}$ for 2^{56} pairs of keys. For another 3×2^{56} pairs of keys $Pl \simeq 2^{28}$. How does one determine which and how many keys are related? Take a key K. Consider all $2^8 - 1$ keys K^* different from K in only the first byte. If none of them is related to K, choose keys K^* different from K only in byte 2 and so on. Again an exhaustive search for all S-boxes can be done separately. The total number of keys for which there are no related keys different in only one byte is about 2^{40}. For many keys K there exists more than one related key, on the average about two and in some cases there are as many as nine 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 six rounds of encryptions. One can start the encryptions in the second round choosing the inputs such that the outputs of the first two rounds of encryption are the same. This can be done easily by precomputing 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 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 each round of encryption one checks whether the encryptions are equal. In most cases only one round of encryption is needed for every plaintext in a pair. Therefore one need only do about $\frac{1}{6} \times 2/\prod_{i=3}^{6} p_i$ encryptions, which is 2^{22} in the optimal cases. The output transformation, which consists of exoring, 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 an illustration Table 1 lists two such examples. The first such *pseudocollision* was found in time 2^{22}, the second in time $2^{22.1}$. We summarize our results.

Fact 1. *For all but 2^{40} keys K in SAFER K, there exists at least one and on the 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 six 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 eight-bit S-box in six 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.*

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

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

From the above discussion the following result also follows:

Fact 2. *For all but 2^{17} keys K in SAFER K, there exists at least one and on the 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 six 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 Lemma 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 Lemma 2 are given in the section about collisions of hash functions. In the following the result of Lemma 2 is used to establish a related-key attack on SAFER K.

3.1. *A Related-Key Chosen Plaintext Attack*

In [1] new attacks based on related keys were introduced. In this section we apply a variant of these attacks to SAFER K. Assume an attacker has 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, that is, encrypt a nonnegligible 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:

1. Choose the values of the bytes b_2 to b_8 at random.
2. Get the 256 encryptions $\{C_i\}$ of the plaintexts $b_1, b_2, \ldots, b_8$ for all values of b_1 encrypted under the first key.
3. Get the 256 encryptions $\{C_j\}$ of the plaintexts $b_1, b_2, \ldots, b_8$ for all values of b_1 encrypted under the second key.
4. Sort the ciphertexts just received and check if a ciphertext in $\{C_i\}$ differs from a 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 key bytes k and $k*$ for which (2) holds. For these key bytes it is checked if the exor of the byte b_1 of the two ciphertexts is the value of the exor 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 probability. It is possible that two ciphertext blocks are different only in one byte without the property that the encryptions after each of the six rounds are equal. However, this would happen only with small probability. The attack is repeated until the last step of the algorithm outputs two ciphertexts. Table 2 lists the complexities for the related-key attack on SAFER K with six and eight rounds. In the outlined attack it is assumed that an attacker is able to get encryptions of chosen plaintexts for two related keys. Although the attack may seem unrealistic, for most block ciphers such attacks are not applicable and moreover it is possible to modify SAFER K to thwart the attacks. Finally, we note that Wagner has improved the related-key attack, as outlined in [6].

Table 2. Related-key chosen plaintext attacks on SAFER K with six and eight rounds finding one byte of the key.

Rounds	# Plaintexts
6	2^{36}
8	2^{50}

3.2. *The Rotations and Bias Additions*

In this section we consider the rotations and biases used in the key schedule of SAFER K. In [11] it is argued that the addition of biases prevent weak keys. There is a reason to have byte rotations as well.

Lemma 3. *The PHT has* 256 *fixed points*.

This result can be found by using Gauss-eliminations on the 8×8 matrix of the PHT. Each byte value in a fixed point is a multiple of 64. There are 16 fixed points where the byte values are either 0 or 128. They are given in Table 6 of Appendix A. If one leaves out the key rotations, but keeps the addition of the biases, then these 16 fixed points for the PHT are "linear structures" for SAFER K with any number of rounds in the following way. Let $a_1, \ldots, a_{16}$ be the fixed points from Table 6. Let $E(K, P) = C$ be the encrypted value of plaintext P using key K, then $E(K + a_i, P + a_i) = C$, where "+" is bytewise 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 mix of group operation exclusive-or and 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

Methods of how to use a block cipher as a building block in hash functions are standardized [5]. In this section we show how to find collisions for hash functions using SAFER as the underlying block cipher. There are essentially two secure single block length hash functions, which use only one encryption per round [15]:

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

$$H_i = h(M_i, H_{i-1}) = E_{M_i}(H_{i-1}) \oplus H_{i-1} \oplus M_i, \tag{5}$$

where $H_0 = IV$ is an initial value. Often h is called the compression function. These schemes are believed to be secure, in the sense that, if the underlying block cipher has no weaknesses, preimage attacks and collision attacks on the compression functions have time complexities 2^m, respectively $2^{m/2}$, encryptions of the underlying m-bit block

Table 3. Collisions for compression functions of type (4) with SAFER K.

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

cipher. Using SAFER K as the underlying block cipher it is possible to find collisions with a complexity of much less than the brute-force method of 2^{32} operations.

We exploit the phenomenon of Lemma 2 to find collisions for the schemes (4) and (5). Consider two plaintexts and two keys, both pairs with different values only in the same single byte, such that $E_{K_1}(P_1) \oplus P_1 = E_{K_2}(P_2) \oplus P_2$ or $E_{K_1}(P_1) \oplus P_1 \oplus K_1 = E_{K_2}(P_2) \oplus P_2 \oplus K_2$ depending on the type of hash function which is considered. One can speed up the search for such quantities by choosing the inputs of SAFER K to the third round, such that the keys encrypt equally in the second and third rounds. For (4), when two ciphertexts different in only one byte are found, one calculates the plaintexts and checks for a collision. In the optimal cases these collisions can be found in estimated time about $2^{22.8}$ encryptions of SAFER K. Table 3 lists examples of such collisions for hash functions of type (4). The first collision was found in time $2^{20.6}$ encryptions, the second collision in time $2^{19.3}$ encryptions. Similarly, collisions for compression functions of type (5) were found in time about 2^{22}.

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 when the initial value is given and fixed. Using the results of Lemma 2 this cannot be done directly for (4), since if the plaintexts are equal for two related keys the hash value of (4) 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 the related key properties can be used once again in the second round to find M_2 and M_2', such that $H_2 = E_{M_2}(H_1) \oplus H_1$ equals $H_2' = E_{M_2'}(H_1') \oplus H_1'$. This attack was not implemented. For the hash functions (5) it is possible to find collisions with a fixed IV. For the pseudocollisions 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, one 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 one should 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, where one example is two keys different only in the fifth bytes with values 9 and 129, respectively. By using similar techniques as before, a collision can be found in expected time about 2^{22} encryptions. Table 4 lists 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 (3) are not independent as assumed.

Table 4. Collisions for hash functions of type (5) 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

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 eight rounds there are still many pairs of keys encrypting some plaintexts the same way. In the optimal case a pair of keys encrypt 2^{15} plaintexts into the same ciphertexts after eight rounds of encryption using our method. Also, a related-key attack is possible for SAFER K with eight rounds, the complexity is given in Table 2. However, using our methods collisions for hash modes using SAFER K with eight rounds cannot be found faster than the time of 2^{32} encryptions. For SAFER K with ten rounds the probability that two keys are related is too small to be of any use in both the related-key and the collision attacks.

5.2. *New Key Schedule for SAFER K*

One way to avoid the problems reported so far is to remove the second exor/addition of the key in every round. To find similar collisions as in the previous section would now require an incorporation of the PHT, which seems very unlikely to succeed.

Next the modified key schedule for SAFER K already described in Section 2.2 is discussed. As can be seen, there is a circular shift of the nine key bytes. In that way the eight user-selected key bytes $k_1, \ldots, k_8$ 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 at least two bytes in some rounds and in at least one byte in the remaining rounds. As an example, in SAFER K with six rounds, two keys will be different in two bytes in at least 9 out of the 13 round keys. In SAFER K with eight rounds, this will be the case in 13 out of the 17 round keys. Therefore our method of finding key-collisions will no longer be applicable. Also, note that if the key is chosen uniformly at random, any round key is uniformly random.

6. Differential Cryptanalysis of SAFER K

In [12] strong evidence was given that SAFER K is secure against differential cryptanalysis. It was argued that a five-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 other types of differentials than the ones given in [12] are considered. We use the notation of "expanded views" from [12] and denote a one-round differential by three tuples of each eight entries. The first tuple indicates the difference in the eight bytes of the inputs to the round, the second tuple indicates the difference of the bytes before the PHT and the third tuple indicates the difference of the bytes after the PHT, that is the difference of the outputs of the round. For convenience, when considering s-round differentials for $s > 1$, the third tuple in all but the last round is omitted, 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 the inputs and in the outputs [12]. This is avoided in the attacks to follow where a *difference* of two bytes x and x' is defined as

$$x - x' \bmod 256.$$

Also, we consider truncated differentials as defined in [8]:

Definition 1. 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 a truncated differential only a subset of all bits of the difference in ciphertexts is predicted. The remaining bits can take any value. In [12] ten tables of "PHT correspondences" are given. The truncated differentials to be described follow from these properties of the PHT. As an example, consider the following one-round differential with the expanded view:

$$[a, b, c, d, 0, 0, 0, 0], [e, f, -e, -f, 0, 0, 0, 0], [2g, g, 2h, h, 0, 0, 0, 0], \tag{6}$$

where $g = 2e + f$ and $h = e + f$. This truncated differential has probability 2^{-16} on the average for all values of a, b, c, d. Consider the first and second tuples of (6). 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 values of both the input and of the involved key bits [9]. 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 is linear with respect to the defined difference, that is, $\mathrm{PHT}(x) - \mathrm{PHT}(x') = \mathrm{PHT}(x - x')$. The PHT transforms the second tuple into the third tuple which is easily verified. As another example, consider the following one-round differential with the expanded view:

$$[0, 0, 0, 0, 0, 0, a, b], [0, 0, 0, 0, 0, 0, 0, 0, e, -e], [e, e, 0, 0, e, e, 0, 0]. \tag{7}$$

This truncated differential has probability 2^{-8} on the average for all values of a, b. In the above examples, no specific values are chosen for the nonzero bytes. It is not intended to predict the actual values of the nonzero bytes, merely predict the bytes which are zero. There are many one-round differentials like (6) and (7) above. To save space a

new notation is introduced. Denote a differential by the indices of the bytes which are nonzero, and write $1234 \rightarrow 1234$ for the differential (6) and, similarly, $78 \rightarrow 1256$ for the differential (7). In Tables 7 and 8 of Appendix B many such differentials are listed. As an example, the differential (6) can be found in Table 8 as Input: 1234, Output: 1234, Prob. 16. The probabilities are given as logarithms, so that "Prob. 16" means a probability of 2^{-16}.

As shown one can now concatenate the one-round differentials of Tables 7 and 8. 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 shown 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 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 also has a probability of 2^{-48}. 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 a total of eight differentials each of probability 2^{-48} covered by this truncated differential, which therefore will have a probability of about $8 \times 2^{-48} = 2^{-45}$.

6.1. *Differential Attacks on SAFER K*

In this section we consider differential attacks on SAFER K using truncated differentials. Consider SAFER K with three rounds and the three-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 one can choose n different plaintexts, all of them with the four rightmost bytes of equal values. From these n plaintexts one can form about $(n \times (n-1))/2 \approx n^2/2$ 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.

The attack on SAFER using truncated differentials goes as follows:

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

The probabilities of the above differentials are not accurate. First, the round keys of SAFER K are not independent as assumed, second, the many pairs processed are not independent. To justify the above method of estimating the probabilities, some tests were done on a mini-version of SAFER K. Instead of working on bytes let SAFER K work on nibbles (4 bits), yielding a 32-bit block cipher with a 32-bit key called SAFER K(32). 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 exor operations are on nibbles and additions are calculated modulo 16. We considered the five-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 $2^{-30.3}$. By using structures consisting of 2^{16} plaintexts, all different in the four leftmost nibbles and equal in the four rightmost nibbles, one obtains about 2^{31} pairs. Of these the expected number of right pairs is 1.6 and about $2^{31}/2^{16} = 2^{15} = 32{,}768$ 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 17 right pairs were found and 327,781 nondiscarded, wrong pairs, thus confirming the theory. In the following section it is shown how to attack five-round SAFER K, 64 bits, using truncated differentials.

6.2. *Five-Round SAFER K*

Consider the following four-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

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

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

$$1458 \rightarrow 1357 \rightarrow 2468 \rightarrow 13 \rightarrow 4, \tag{10}$$

$$1458 \rightarrow 2468 \rightarrow 2468 \rightarrow 13 \rightarrow 4. \tag{11}$$

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 7 and 8. 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. 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, which follows from Lemma 1, see also [12]. Therefore one obtains a five-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 $2^{-69.7}$. One can use structures of each 2^{32} plaintexts yielding 2^{63} pairs with the desired difference in the inputs. One needs about 2^{70} pairs to get one right pair and therefore about 128 structures, a total of 2^{39} plaintexts. The analysis can be performed on each structure and thus the memory requirements are 2^{32} 64-bit quantities. In the following the analysis will be described 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 consists of bytewise exoring and adding modulo 256 the last-round key. Therefore, right pairs for the above truncated differential will have the following form:

$$[z_1,\ x,\ 2x,\ z_2,\ z_3,\ x,\ 2x,\ z_4], \tag{12}$$

where the z_i's are values that cannot be predicted immediately. 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 and from the differential 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 correct values in the least significant bits. This second filtering process discards all but one out of 16 pairs, thus one is left with 2^{41} pairs. The expected difference before the output transformation is $[2x, x, 2x, x, 2x, x, 2x, x]$ for a right pair. On the 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. Note that according to Lemma 4 a key byte k and $k \oplus 1$ are indistinguishable in this test. 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 of the first round. Since there are two possible sets of four bytes with nonzero values after the first round according to (8)–(11) every pair will suggest $16 \times 2^{-15} = 2^{-11}$ values on the average of the four key bytes 1, 4, 5, and 8. Here it is exploited 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 $(\frac{1}{2}) \times 2^{30} \times 2^{32} = 2^{61}$. The time and space requirements of the filtering processes above can be made small. One method is the following, proposed by an anonymous referee of [9]. Let the ciphertexts be denoted $(c_1, \ldots, c_8)$. Hash each ciphertext to $(c_3 - 2 * c_2, c_6 - c_2, c_7 - 2 * c_2)$. The ciphertexts

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

Rounds	Time	Plaintexts	Storage
5	2^{61}	2^{39}	2^{32}
5	2^{49}	2^{44}	2^{32}
5	2^{37}	2^{45}	2^{32}

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 has probability $2^{-69.7}$, so by generating 2^{70} pairs one gets one right pair with probability 0.71. Thus the right key value is suggested with probability at least 0.71, since it will happen that the wrong pairs also suggest the correct key value, and a wrong key value is suggested with an average probability of $2^{30}/2^{32} = 0.25$. 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 the above basic attack is repeated. 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 $E(X(T)) = T \times 0.71$ and $E(Y(T)) = T \times 0.25$. By assuming that $X(T)$ 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], $\Pr(X(32) < Y(32)) \simeq 2^{-16}$ and $\Pr(X(64) < Y(64)) < 2^{-28}$. Thus, by repeating the attack 32 times using totally 2^{44} plaintexts, the right key value will be among the $2^{32} \times 2^{-16} = 2^{16}$ most suggested values with a high probability. To increase the probability of success, one can choose the 2^{17} most suggested values of the 32-bit 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^{49} encryptions. Every counter can be implemented as one byte, thus the storage needed for the counters is only one-eighth of the storage needed for the plaintexts. Another possibility is to repeat the attack 64 times using totally 2^{45} plaintexts. The right key value will be among the first 16 most suggested values with a high probability. Taking the 32 most suggested values and searching exhaustively for the remaining 32 bits, the time complexity of the attack is about 2^{37}. Table 6.2 summarizes the complexities of the attacks for SAFER K with five rounds.

In the above attack the four-round truncated differential $1458 \rightarrow 4$ with probability $2^{-69.7}$ was used. There are many other differentials that can be used in variants of the above attacks, which the reader can verify by studying Tables 7 and 8.

6.3. *Six-Round SAFER K*

For SAFER K with six rounds there is a similar truncated differential as the one above for SAFER K with four and five 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 six 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 six rounds. Also, there are truncated differentials predicting the values of four bytes after six rounds with similar probabilities. As an example, the six-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.

6.4. *SAFER K*-128, *SAFER SK*-64, *and SAFER SK*-128

The above attack for SAFER K with five 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. However, since the first and the last round keys depend only on the same 64 bits of the original key, one finds 32 bits of the 128-bit key by the above attack. By using other similar truncated differentials one can find the remaining 32 bits of the first and the last round keys. With the knowledge of these keys, one is left with a cipher easier to attack than the original.

The truncated differential used above for SAFER K with five rounds was chosen to minimize the number of counters for 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, the first part of the attack with time complexity 2^{61} is applicable. The 2^{41} nondiscarded pairs will suggest 16 values of round key bytes in positions 1, 4, 5 and 8 in the last round. These bytes correspond to byte nos. 2, 5, 6, and 9 in the original key, where byte 9 is the parity byte. 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, one gets 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 was done for SAFER K. However, simply trying all possible candidates is possible and an exhaustive search for the key at this point would require about $(\frac{1}{2}) \times 2^{62} = 2^{61}$ operations. It is left as an open problem to find other differentials to improve the attack on SAFER SK. 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 $2^{-69.7}$.

6.5. *Higher-Order Differentials*

A dth-order differential is a collection of 2^d plaintexts and the corresponding ciphertexts. It follows from [10] and [8] that a dth-order differential of a function f of nonlinear order at most d is a constant. In [12] it was shown that the nonlinear order of the S-boxes in SAFER is 7. Thus, one would expect that after two rounds of SAFER the nonlinear order is about 49. However, in this section an interesting property is shown for two rounds of

(any of the four variants of) SAFER. There exists a structure for SAFER, reminiscent of that of higher-order differentials, containing 256 texts with probability 1 after two rounds of encryption.

Consider a collection of 256 plaintexts, such that the first seven bytes are constant, and where the eighth byte take all 256 values. We define the "difference" of 256 bytes as the sum of the bytes modulo 256, $\Delta x = \Delta(x(0), \ldots, x(255)) = \sum_{j=0}^{255} x(j) \bmod 256$. Note that, with this definition, a difference is invariant of modulo addition of a key byte. Denote by $c_i^1(j)$, for $j = 0, \ldots, 255$, the ith bytes after one round of encryption for any fixed key. Since the 256 eighth plaintext bytes are all different, the 256 eighth bytes of the input to the PHT will also be different. Therefore the set $\{c_i^1(j) \mid j = 0, \ldots, 255\}$ equals the set $\{0, \ldots, 255\}$ for all i, which can be seen from (1). It follows that $\Delta c_i^1 = 128$, for $i = 1, \ldots, 8$. Note that $\sum_{j=0}^{255} j \bmod 256 = 128$. Let $c_i^2(j)$ be the ciphertext bytes after two rounds of encryption. Since the 256 input bytes $c_i^1(j)$ are all different for each i, the eight sets of 256 input bytes to the PHT will each contain 256 different elements. It follows from (1) that $\Delta c_1^2 = 128$ and $\Delta c_i^2 = 0$ for $i = 2, \ldots, 8$, where additions are modulo 256.

To sum up, a structure has been described reminiscent of a higher-order differential containing 256 texts with probability 1 after two rounds of encryption of (any of the four variants of) SAFER. The structure can be used to attack SAFER with three rounds, but is unclear how to extend such an attack to more rounds without a major increase in complexity. However, note that for a random permutation such a structure will have a probability of 2^{-64}.

7. Conclusion

In this paper the block cipher SAFER K was analyzed. We discovered a weakness in the key schedule and exploited it in related-key attacks and in collision attacks for SAFER K in the standard hashing modes much faster than by brute-force. Our analysis together with that of Murphy [14] led the designer of SAFER K to adopt our proposed strengthened key schedule for SAFER K, yielding the new block cipher SAFER SK with a recommended minimum of eight rounds. We considered truncated differentials for five-round SAFER K and established a differential attack, which finds the secret key in time much faster than an exhaustive search. The attack needs only a small amount of chosen plaintext compared with conventional differential attacks which illustrates the importance of truncated differentials. The success of the attacks does not depend on special properties of the S-boxes used in SAFER K and would work for most S-boxes. The differential attack is not directly applicable to SAFER SK, but it is not prevented in a significant way by a modified key schedule. The main property that makes our truncated differential attacks possible is the Pseudo-Hadamard Transformation. However, for SAFER K with more than five rounds our method of filtering out wrong pairs is not efficient enough to do a successful differential attack. Though it might be possible to improve our methods to attack SAFER K versions with six rounds, we strongly believe that SAFER SK with eight rounds, as now recommended, or more rounds are invulnerable to all our attacks. Finally we presented a structure of 256 plaintexts, for which the sum of the corresponding ciphertexts can be predicted with certainty after two rounds of encryption. The importance of this discovery for SAFER with six or more rounds remains an open problem.

Acknowledgments

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

Appendix A. Some Fixed Points of the PHT

Table 6. The 16 fixed points for the PHT with only entries 0 and $\gamma = 128$.

(0 0 0 0 0 0 0 0)	(0 0 0 0 γ γ 0 0)
(0 0 γ 0 0 0 γ 0)	(0 0 γ 0 γ γ γ 0)
(0 γ 0 γ 0 0 0 0)	(0 γ 0 γ γ γ 0 0)
(0 γ γ γ 0 0 γ 0)	(0 γ γ γ γ γ γ 0)
(γ 0 0 γ 0 γ γ γ)	(γ 0 0 γ γ 0 γ γ)
(γ 0 γ γ 0 γ 0 γ)	(γ 0 γ γ γ 0 0 γ)
(γ γ 0 0 0 γ γ γ)	(γ γ 0 0 γ 0 γ γ)
(γ γ γ 0 0 γ 0 γ)	(γ γ γ 0 γ 0 0 γ)

Appendix B. One-Round Differentials of SAFER

Table 7. 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.
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 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									

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, New York, 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, Berlin, 1995.

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

[5] ISO-10118. Information technology – security techniques – hash-functions, part 1: General and part 2: Hash-functions using an *n*-bit block cipher algorithm. ISO/IEC, 1994.

[6] J. Kelsey, B. Schneier, and D. Wagner. Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES. In Neal Koblitz, editor, *Advances in Cryptology*: *CRYPTO* '96, LNCS 1109, pages 237–251. Springer-Verlag, Berlin, 1996.

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

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

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

[10] X. Lai. Higher order derivatives and differential cryptanalysis. In R. Blahut, editor, *Communication and Cryptography, Two Sides of One Tapestry*. Kluwer Academic, Dordrecht, 1994. ISBN 0-7923-9469-0.

[11] J.L. Massey. SAFER K-64: A byte-oriented block-ciphering algorithm. In R. Anderson, editor, *Fast Software Encryption - Proc. Cambridge Security Workshop*, Cambridge, LNCS 809, pages 1–17. Springer-Verlag, Berlin, 1994.

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

[13] J.L. Massey. Strengthened key schedule for the cipher SAFER. Posted on USENET newsgroup sci.crypt, September 9, 1995.

[14] S. Murphy. An analysis of SAFER. Private communication, 1994.

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

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