A Key-schedule Weakness in SAFER K-64

Lars R. Knudsen*

Laboratoire d'Informatique
École Normale Supérieure
Paris, France

Abstract. In this paper we analyse SAFER K-64 and show a weakness in the key schedule. It has the effect that for almost every key K, there exists at least one different key K^*, such that for many plaintexts the outputs after 6 rounds of encryption are equal. The output transformation causes the ciphertexts to differ in one of the 8 bytes. Also, the same types of keys encrypt even more pairs of plaintexts different in one byte to ciphertexts different only in the same byte. This enables us to do a related-key chosen plaintext attack on SAFER K-64, which finds 8 bits of the key requiring from 2^{44} to about 2^{47} chosen plaintexts.
While our observations may have no greater impact on the security of SAFER K-64 when used for encryption in practice, it greatly reduces the security of the algorithm when used in hashing modes, which is illustrated. We give collisions for the well-known secure hash modes using a block cipher. Also we give a suggestion of how to improve the key schedule, such that our attacks are no longer possible.

1 Introduction

In [6] a new encryption algorithm, SAFER K-64, hereafter denoted SAFER, was proposed. Both the block and the key size is 64. The algorithm is an iterated cipher, such that encryption is done by iteratively applying the same function to the plaintext in a number of rounds. Finally an output transformation is applied to produce the ciphertext. For SAFER the suggested number of rounds is 6. Strong evidence has been given that SAFER is secure against differential cryptanalysis [7] and against linear cryptanalysis [2]. In [11] it was shown that by replacing the S-boxes in SAFER by random permutations, about 6% of the resulting ciphers can be broken faster than by exhaustive search.

In this paper we analyse SAFER and show a weakness in the key schedule. It has the effect that for virtually every key K, there exists at least one different key K^*, such that for a non-negligible fraction of all plaintexts the outputs after 6 rounds of encryption are equal. The output transformation causes the ciphertexts to differ in one of the 8 bytes. These pairs of plaintexts and ciphertexts can be found in time from about 2^{22} to 2^{28} encryptions. All estimates of complexity in this paper are the number of 6 round SAFER encryptions. Two keys encrypting a plaintext into the same ciphertext is called a "key-collision" in the literature

* Postdoctoral researcher sponsored by the Danish Technical Research Council.

D. Coppersmith (Ed.): Advances in Cryptology - CRYPTO '95, LNCS 963, pp. 274-286, 1995.

and in [10] a brute-force key-collision attack on the DES was given, which can be applied to any block cipher. Given a plaintext P the method finds two keys for which the two encryptions of P are equal and requires about 2^{32} operations for a 64 bit block cipher.

What we have found for SAFER is much stronger. For (almost every) given key K there exists (at least) one other key K^*, different from K only in one byte, such that the encryption functions induced by the two keys encrypt from 2^{22} to 1.7×2^{28} plaintexts the same way in the 6 rounds of encryptions. The output transformation makes the ciphertexts differ in one byte, the same byte in which the keys differ. For some keys, K, there are up to 9 other keys encrypting a non-negligible fraction of all plaintexts in the same way as K. Also, for the same types of keys, K and K^*, the encryption functions induced by the two keys encrypt from 2^{29} to 2^{35} pairs of plaintexts, P and P^*, different only in one byte, the same way in the 6 rounds of encryptions. The output transformation makes the ciphertexts differ in the same byte. Interestingly, the keys, the plaintexts and the ciphertexts differ in the same byte.

We use our observations to establish related-key chosen plaintext attacks, which using from 2^{44} to 2^{47} chosen plaintexts finds 8 bits of the secret key with probabilities from 1 to 2^{-59} depending on certain circumstances of the attacks. Related-key attacks are not the most realistic attacks, and our results may have no greater impact on the security of SAFER in practice when used for encryption. However, first of all, it can be avoided by re-constructing the key schedule, secondly it greatly reduces the security of the algorithm when used in hashing modes.

In hashing modes using a block cipher algorithm as building block the plaintext (and/or the key) is exclusive-or'ed to the ciphertext to produce a kind of one-wayness in the hash algorithm. We found collisions of such hash functions in estimated time about 2^{23} encryptions when SAFER is used as the underlying block cipher. This should be compared with a brute force collision attack, which requires about 2^{32} operations. The keys we used were well-chosen, but with our method collisions can be found faster than a brute force attack for most keys.

This paper is organised as follows. First we give a short description of SAFER. In Sect. 3 we describe the weakness in the key schedule and give examples of the above mentioned (pseudo)-collisions. Next we use our observations to establish a related-key chosen plaintext attack on SAFER. In Sect. 4 we describe attacks on hash modes using SAFER and give examples of collisions. In Sect. 5 we give different methods of how to improve SAFER to avoid the problems described in the preceding sections.

2 Description of SAFER

SAFER is an r round iterated cipher with both block and key size of 64 bits and with all operations done on bytes. The key is expanded to $2r + 1$ round keys each of 64 bits, described later. The designer's recommendation for r is 6 [6]. Each round takes 8 bytes of text input and two round keys each of 8 bytes.

The input and the round keys are divided into 8 bytes and the first round key is xor'ed, respectively added modulo 256, according to Fig. 1. The bytes are then processed using 2 permutations or S-boxes, $X(a) = (45^a \bmod 257) \bmod 256$, and the inverse of X, $L(a) = log_{45}(a) \bmod 257$ for $a \neq 0$ and where $L(0) = 128$. After the S-boxes each byte of the second round key is added modulo 256, respectively xor'ed, and finally the so-called *Pseudo-Hadamard Transformation (PHT)* is applied to produce the output of the round. PHT is defined by three layers of the 2-PHT, which is defined by

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

where each coordinate is taken modulo 256. Between two layers of 2-PHT's a permutation of the bytes is done, see Fig. 1. After the last round an output transformation is applied, which consists of xoring, respectively adding modulo 256, the last-round key.

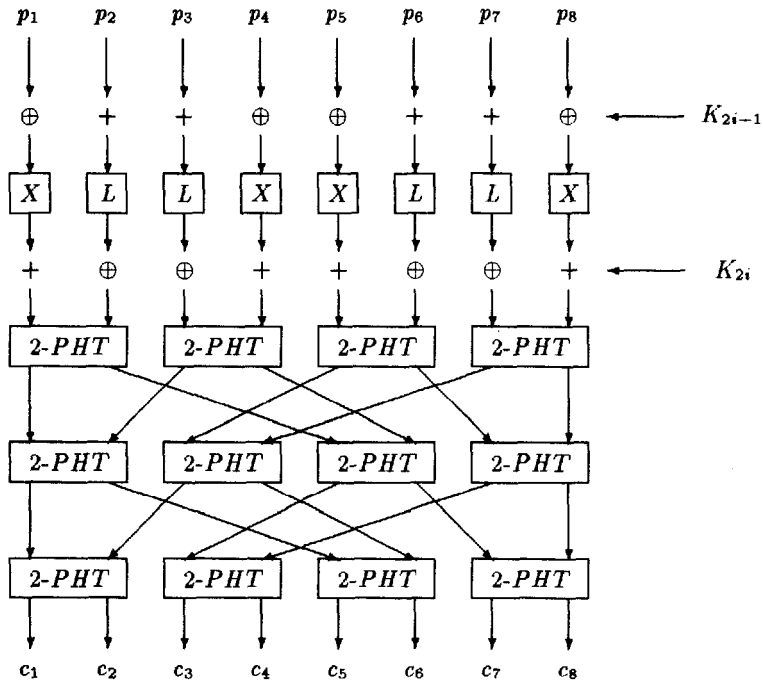

Fig. 1. One round of SAFER.

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

$$k_{i,j} = k_{i-1,j} << 3 \tag{1}$$

$$K_{i,j} = k_{i,j} + bias[i, j] \bmod 256 \tag{2}$$

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.

2.1 Some Properties of SAFER

The following lemma is used in this paper.

Lemma 1. *Let X be the exponentiation function of SAFER 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 was introduced to give the cipher *confusion* [6]. 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 [7].

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. 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.

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

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 such two keys is

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

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 (3) of the round probabilities is a good approximation for SAFER with 6 rounds. Since this phenomenon is isolated to one S-box we could easily do an exhaustive search for all such pairs of keys. We found that for two keys different only in the sixth 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 eighth 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 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 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.

Then 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 xoring, 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 collision was found in time 2^{22}, the second in time $2^{22.1}$. We summarize our results.

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 (hex notation).

Theorem 3. *For all but 2^{40} keys K in SAFER, 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 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 also the following result follows.

Theorem 4. *For all but 2^{17} keys K in SAFER, 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 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 3, 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 4 are given in the section about collisions of hash functions. We can use Theorem 4 to establish a related-key attack on SAFER.

3.1 A Related-key Chosen Plaintext Attack

In [3, 4, 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. 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 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 4 the above algorithm succeeds with probability at least $2^{29} \times 2^{-64} = 2^{-35}$ and therefore needs to be repeated at most about 2^{35} times, in the optimal cases only 2^{29} times. The number of chosen plaintexts needed in the worst cases is about $2 \times 2^8 \times 2^{35} = 2^{44}$. 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^{47} 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 4. 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 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 reduced to (the first) 4 and 5 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

Chosen plaintexts	Probability	Conditions
2^{44}	0.63×1	Two related keys
2^{44}	$0.63 \times 1/73$	The two keys differ in one known byte position.
2^{47}	$0.63 \times 1/73$	The two keys differ in one unknown byte position.
2^{47}	0.63×2^{-59}	The two keys are randomly chosen.

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

4 rounds		5 rounds		
Ch. pl.texts	Prob.	Ch. pl.texts	Prob.	Conditions
2^{30}	0.63×1	2^{37}	0.63×1	Two related keys.
2^{30}	$0.63 \times 1/14$	2^{37}	$0.63 \times 1/35$	The two keys differ in one known byte position.
2^{33}	$0.63 \times 1/14$	2^{40}	$0.63 \times 1/35$	The two keys differ in one unknown byte position.
2^{33}	0.63×2^{-57}	2^{40}	0.63×2^{-58}	The two keys are randomly chosen.

Table 3. Related-key chosen plaintext attacks on SAFER reduced to four and five rounds finding one byte of the key. (Worst case considerations.)

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 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^{47} chosen plaintexts. We assume here that the time to sort and compare ciphertexts is negligible compared to the time of getting the many encryptions. After 73 days Eve has used about 2^{53} chosen plaintexts and with a probability 0.63 found at least 8 key bits. The number of chosen plaintexts can be reduced to 2^{50}, if Eve can predict which byte of the secret key is changed from day to day. Similar attacks on SAFER with a reduced number of rounds will have much lower complexities.

3.2 The Rotations and Bias Additions

In this section we consider the rotations and bias additions used in the key schedule of SAFER. In [6] 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 [3, 4, 1]. We have found a reason to have byte rotations as well.

Lemma 5. *PHT has 256 fixed points.*

This result can be found by using Gauss-eliminations on the 8×8 matrix of PHT etc. 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 Table 4. If

(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 4. The 16 fixed points for the PHT with only entries 0 and 128.

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 with any number of rounds in the following way. Let $a_1, ..., a_{16}$ be the fixed points from Table 4. 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 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 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 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 [8, 9]:

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

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

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 preimage attacks and free-start collision attacks have time complexities 2^m and $2^{m/2}$ encryptions, respectively, of the underlying m-bit block cipher [5, 8]. In a free-start attack the attacker is free to choose the initial values. Using SAFER 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 collision for the schemes (4) and (5).

4.1 Free-start Collisions

In this section we exploit the phenomenon of Theorem 4. In the attacks to follow we choose two plaintexts different only in the byte for which both the keys and ciphertexts differ. We hope in this way to obtain plain- and ciphertexts and keys, such that

$$E_{K_1}(P_1) \oplus P_1 = E_{K_2}(P_2) \oplus P_2 \quad or$$
$$E_{K_1}(P_1) \oplus P_1 \oplus K_1 = E_{K_2}(P_2) \oplus P_2 \oplus K_2$$

We can speed up this search by choosing the inputs of SAFER to the third round, such that the keys encrypt equally in the second and third rounds. 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. In Table 5 we give 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.

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 5. Free-start collisions for hash functions of type (4) with SAFER.

It is possible to find free-start collisions for hash functions of type (5) also. We found such collisions in time about 2^{22}. In the next section we give examples of collisions for hash functions of type (5) with a fixed start.

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 6. Fixed-start collisions for hash functions of type (5) with SAFER.

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. Using our observations about SAFER this cannot be done for the hash round function (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 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 (5) it is possible to find fixed start collisions for the hash round function. For our pseudo-collisions for SAFER, 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. 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 6 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 (3) are not independent as assumed.

5 Improvements of SAFER

In this section we suggest modifications of SAFER, 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 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. The output transformation makes those ciphertexts differ in one byte. But in contrast to SAFER with 6 rounds collisions cannot be found faster than the time of 2^{32} encryptions. Still, it must be an undesirable property for a block cipher.

In the optimal case for SAFER 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.

5.2 New Key Schedule for SAFER

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 give a modified key schedule for SAFER with any number of rounds. Let $K = (k_{1,1}, ..., k_{1,8})$ be an 8 byte key and let

$$k_{1,9} = \bigoplus_{i=1}^{8} k_{1,i}$$

The round keys are defined as follows. $K_{1,j} = k_{1,j}$ for $j = 1, ..., 8$ and

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

for $i = 2, ..., 2r+1$, $j = 1, ..., 8$. There is a circular shift of the nine key bytes. In that way the 8 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 with 6 rounds, two keys will be different in two bytes in 9 out of the 13 round keys. In SAFER 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 Conclusion

We described a weakness in the key schedule of SAFER K-64 and exploited it to establish a related-key attack much faster than a brute force attack, and showed by examples that collisions for the standard hashing modes based on a block cipher using SAFER K-64 are easy to find. A new key schedule was suggested, so that the resulting cipher is invulnerable to our attacks. To conclude, we believe that the results presented in this paper show that a change in the key schedule of SAFER K-64 is necessary.

7 Acknowledgments

I would like to thank Jim Massey and Serge Vaudenay for fruitful discussions and Carlo Harpes, Xuejia Lai and Torben Pedersen for valuable comments.

References

1. E. Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.
2. C. Harpes, G.G. Kramer, and J.L. Massey. A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In L. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology - Eurocrypt'95, LNCS 921*, pages 24–38. Springer Verlag, 1995.
3. 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.
4. L.R. Knudsen. *Block Ciphers - Analysis, Design and Applications.* PhD thesis, Aarhus University, Denmark, 1994.
5. X. Lai. *On the Design and Security of Block Ciphers.* PhD thesis, ETH, Zürich, Switzerland, 1992.
6. J.L. Massey. SAFER K-64: A byte-oriented block-ciphering algorithm. In *Fast Software Encryption - Proc. Cambridge Security Workshop, Cambridge, U.K., LNCS 809*, pages 1–17. Springer Verlag, 1994.
7. J.L. Massey. SAFER K-64: One year later. In *Proc. - The Leuven Algorithms Workshop.* Springer Verlag, 1995. To appear.
8. B. Preneel. *Analysis and Design of Cryptographic Hash Functions.* PhD thesis, Katholieke Universiteit Leuven, January 1993.
9. 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.
10. J.-J. Quisquater and J.-P. Delescaille. How easy is collision search. Applications to DES. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - Eurocrypt '89, LNCS 434*, pages 429–433. Springer Verlag, 1990.
11. S. Vaudenay. On the need for multipermutations: Cryptanalysis of MD4 and SAFER. In *Proc. - The Leuven Algorithms Workshop.* Springer Verlag, 1994. To appear.