# Impossible Differential Attacks on Reduced-Round SAFER Ciphers[*] NES/DOC/KUL/WP5/30/1

Jorge Nakahara Jr[†] Bart Preneel
Joos Vandewalle
Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC, Belgium
{jorge.nakahara,bart.preneel,joos.vandewalle}@esat.kuleuven.ac.be

March 12, 2003

**Abstract**

This report describes impossible differential (ID) attacks on 3.75-round SAFER SK-64, using $2^{32}$ chosen texts, $2^{40}$ memory, and $2^{62}$ time. Moreover, an ID attack on both 2.75-round SAFER+ and on 2.75-round SAFER++ uses $2^{64}$ data, $2^{97}$ memory, and $2^{60}$ time. We used the miss-in-the-middle technique developed by Biham *et al.* These attacks do not endanger the security of SAFER ciphers, and indicate that ID attacks work better on ciphers with slow diffusion such as Skipjack.

## 1 Introduction

In [3], Biham *et al.* described a new cryptanalytic tool called Impossible Differential (ID) cryptanalysis. Unlike differentials in DC, which are used to predict ciphertext differences with the highest possible probabilities, the differentials used in ID cryptanalysis are chosen to predict (with probability one) impossible events (namely, that never happen). While in DC the

[*] The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

[†] Sponsored by a grant from the Katholieke Universiteit Leuven, and partially by GOA project Mefisto 2000/06 of the Flemish Government.

differentials are used to distinguish the correct key, in ID the procedure is complementary in the sense that the differential suggests incorrect keys, and the remaining key, is the correct key value. One idea for the construction of impossible differentials, suggested in [3], is called miss-in-the-middle. It consists in combining two differentials, each individually holding with probability one, but whose differences cannot be simultaneously satisfied. Their combination leads to a contradiction. Once the impossible event is verified, it can be used as a distinguisher for wrong keys, in order to find the correct one by elimination, a process called sieving. This paper is organized as follows: Sect. 2 describes the most relevant features of the SAFER ciphers. Sect. 5 describes key-recovery ID attacks on reduced-round variants of these ciphers, and the resulting complexities. Sect. 7 concludes the paper.

## 2 The SAFER Family of Block Ciphers

SAFER K-64 is a 64-bit block cipher, with 64-bit key, and iterating six rounds plus an output tranformation. SAFER K-64 was designed by Massey [7] as the first member of the SAFER family of block ciphers.

SAFER K-64 is a byte-oriented cipher namely, both encryption and decryption use only byte-wise operations. Briefly, all rounds have the same structure: a first key-mixing layer in which subkey bytes are combined with data via xor and addition (XOR/ADD), in alternated fashion; one non-linear (NL) layer, where two s-boxes are applied alternately, a second key mixing layer, alternating addition and xor (ADD/XOR) with another round subkey, and finally, a linear transformation layer (PHT) that mixes all bytes in a block. After the last round there is an output transformation which consists of an XOR/ADD key-mixing layer. The s-boxes, modular addition and xor are the same in all SAFER cipher members. The main differences among them is in the key schedule algorithms and the linear transformation performed by distinct Pseudo-Hadamard Tranform (PHT) matrix. There are two S-boxes in SAFER K-64: a discrete-eXponentiation based $X(a) = (45^a \bmod 257) \bmod 256$ (X-box, for short), and a discrete-Logarithmic based $L(a) = \log_{45}(a) \bmod 257$ (L-box, for short) for $a \neq 0$, with the special case $L(0) = 128$. Note that $X(L(a)) = L(X(a)) = a, \forall a \in \mathbb{Z}_{256}$.

In SAFER K/SK ciphers, the PHT layer can be represented by a matrix. Let the input to a PHT layer be denoted $Y = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$ and its output by $Z = (z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8)$, where $y_i, z_i \in \mathbb{Z}_{256}$, $1 \leq i \leq 8$. The PHT can be described by $Z = Y \cdot M$, where $M$ is the PHT matrix, and

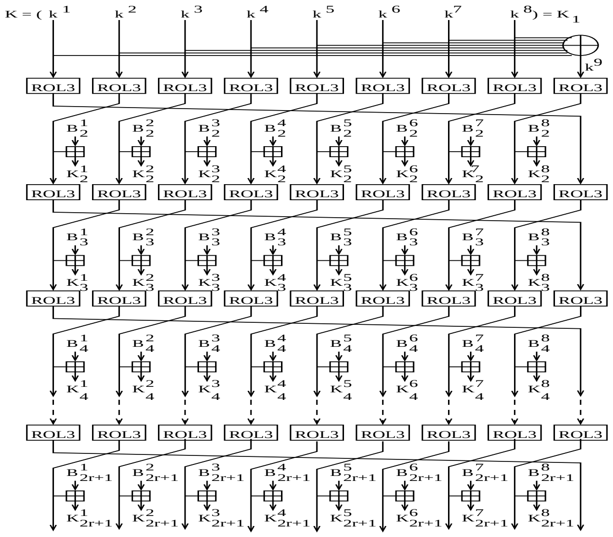


Figure 1: Computational graph of key schedule of SAFER SK-64.

$M^{-1}$ its inverse:

$$M = \begin{pmatrix} 8 & 4 & 4 & 2 & 4 & 2 & 2 & 1 \\ 4 & 2 & 4 & 2 & 2 & 1 & 2 & 1 \\ 4 & 2 & 2 & 1 & 4 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad M^{-1} = \begin{pmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 2 & -2 & -2 & 2 \\ -1 & 2 & 1 & -2 & 1 & -2 & -1 & 2 \\ 1 & -2 & -1 & 2 & -2 & 4 & 2 & -4 \\ -1 & 1 & 2 & -2 & 1 & -1 & -2 & 2 \\ 1 & -1 & -2 & 2 & -2 & 2 & 4 & -4 \\ 1 & -2 & -2 & 4 & -1 & 2 & 2 & -4 \\ -1 & 2 & 2 & -4 & 2 & -4 & -4 & 8 \end{pmatrix}$$

The key schedule of 64-bit SAFER members uses constant byte values called key biases, denoted $B_i$, $2 \le i \le 2r-1$. Each $B_i$ is composed of eight bytes $B_{i,j}$:

$$B_{i,j} = 45^{45^{9i+j} \bmod 257} \bmod 257 = X(X(9i+j)),$$

for $1 \le j \le 8$ (Fig. 1).

# 3 SAFER+

SAFER+ is a 128-bit block cipher with a variable key size of 128, 192 or 256 bits. SAFER+ was designed by Massey, Khachatrian and Kuregian [8]

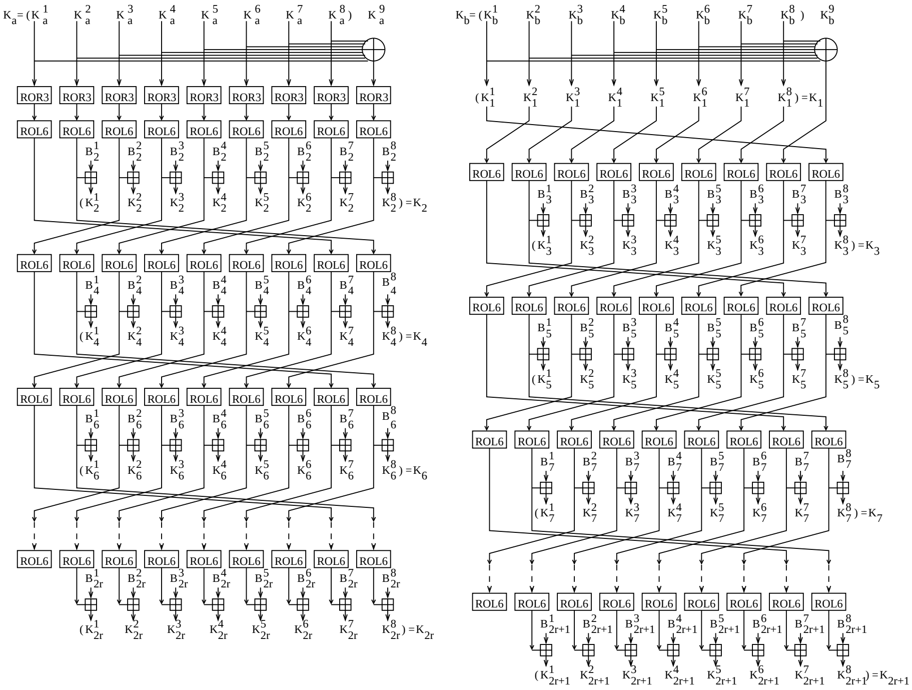

K_a = (K_a^1 K_a^2 K_a^3 K_a^4 K_a^5 K_a^6 K_a^7 K_a^8) K_a^9
ROR3 ROR3 ROR3 ROR3 ROR3 ROR3 ROR3 ROR3 ROR3
ROL6 ROL6 ROL6 ROL6 ROL6 ROL6 ROL6 ROL6 ROL6
B_2^1 B_2^2 B_2^3 B_2^4 B_2^5 B_2^6 B_2^7 B_2^8
(K_2^1 K_2^2 K_2^3 K_2^4 K_2^5 K_2^6 K_2^7 K_2^8) = K_2
B_4^1 B_4^2 B_4^3 B_4^4 B_4^5 B_4^6 B_4^7 B_4^8
(K_4^1 K_4^2 K_4^3 K_4^4 K_4^5 K_4^6 K_4^7 K_4^8) = K_4
B_6^1 B_6^2 B_6^3 B_6^4 B_6^5 B_6^6 B_6^7 B_6^8
(K_6^1 K_6^2 K_6^3 K_6^4 K_6^5 K_6^6 K_6^7 K_6^8) = K_6
B_2r^1 B_2r^2 B_2r^3 B_2r^4 B_2r^5 B_2r^6 B_2r^7 B_2r^8
(K_2r^1 K_2r^2 K_2r^3 K_2r^4 K_2r^5 K_2r^6 K_2r^7 K_2r^8) = K_2r
K_b = (K_b^1 K_b^2 K_b^3 K_b^4 K_b^5 K_b^6 K_b^7 K_b^8) K_b^9
(K_1^1 K_1^2 K_1^3 K_1^4 K_1^5 K_1^6 K_1^7 K_1^8) = K_1
B_3^1 B_3^2 B_3^3 B_3^4 B_3^5 B_3^6 B_3^7 B_3^8
(K_3^1 K_3^2 K_3^3 K_3^4 K_3^5 K_3^6 K_3^7 K_3^8) = K_3
B_5^1 B_5^2 B_5^3 B_5^4 B_5^5 B_5^6 B_5^7 B_5^8
(K_5^1 K_5^2 K_5^3 K_5^4 K_5^5 K_5^6 K_5^7 K_5^8) = K_5
B_7^1 B_7^2 B_7^3 B_7^4 B_7^5 B_7^6 B_7^7 B_7^8
(K_7^1 K_7^2 K_7^3 K_7^4 K_7^5 K_7^6 K_7^7 K_7^8) = K_7
B_2r+1^1 B_2r+1^2 B_2r+1^3 B_2r+1^4 B_2r+1^5 B_2r+1^6 B_2r+1^7 B_2r+1^8
(K_2r+1^1 K_2r+1^2 K_2r+1^3 K_2r+1^4 K_2r+1^5 K_2r+1^6 K_2r+1^7 K_2r+1^8) = K_2r+1

Figure 2: Computational graph of key schedule of SAFER SK-128.

for the AES Process [1]. SAFER+ is an iterated byte-oriented block cipher, with $r = 8$ rounds for 128-bit keys, $r = 12$ rounds for 192-bit keys and $r = 16$ rounds for 256-bit keys.

For all key sizes there is an output transformation that consists of a key-mixing layer with a 128-bit subkey.

The linear transformation in a round, with input $X = (x_1, \ldots, x_{16})$, has output given by $Y = (y_1, \ldots, y_{16})$, where $Y = X \cdot M_+$, where $M_+$ is the encryption PHT matrix and $M_+^{-1}$, is its inverse.

Fig. 3 shows the tweaked key schedule of SAFER+ for 128-, 192- and 256-bit keys. The key schedule uses a 256-bit key register. For 128-bit keys, the 256-bit key register is loaded with two copies of the 128-bit key. For 192-bit keys, the first 192 bits of the register are loaded with the key, and the last 64 bits, with the key bytes $K_9, \ldots, K_{16}$. For 256-bit keys, the register uses the key itself.

$$M_+ = \begin{pmatrix}
2 & 2 & 1 & 1 & 16 & 8 & 2 & 1 & 4 & 2 & 4 & 2 & 1 & 1 & 4 & 4 \\
1 & 1 & 1 & 1 & 8 & 4 & 2 & 1 & 2 & 1 & 4 & 2 & 1 & 1 & 2 & 2 \\
1 & 1 & 4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 2 & 2 & 1 & 1 \\
1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 1 & 1 & 1 & 1 \\
4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 1 & 1 & 1 & 1 & 2 & 2 \\
2 & 2 & 2 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 4 & 2 & 4 & 2 & 16 & 8 & 2 & 1 & 2 & 2 & 4 & 4 & 1 & 1 \\
1 & 1 & 2 & 1 & 4 & 2 & 8 & 4 & 2 & 1 & 1 & 1 & 2 & 2 & 1 & 1 \\
2 & 1 & 16 & 8 & 1 & 1 & 2 & 2 & 1 & 1 & 4 & 4 & 4 & 2 & 4 & 2 \\
2 & 1 & 8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 1 \\
4 & 2 & 4 & 2 & 4 & 4 & 1 & 1 & 2 & 2 & 1 & 1 & 16 & 8 & 2 & 1 \\
2 & 1 & 4 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 8 & 4 & 2 & 1 \\
4 & 2 & 2 & 2 & 1 & 1 & 4 & 4 & 1 & 1 & 4 & 2 & 2 & 1 & 16 & 8 \\
4 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 1 & 8 & 4 \\
16 & 8 & 1 & 1 & 2 & 2 & 1 & 1 & 4 & 4 & 2 & 1 & 4 & 2 & 4 & 2 \\
8 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 4 & 2
\end{pmatrix}$$

$$M_+^{-1} = \begin{pmatrix} 2 & -2 & 1 & -2 & 1 & -1 & 4 & -8 & 2 & -4 & 1 & -1 & 1 & -2 & 1 & -1 \\ -4 & 4 & -2 & 4 & -2 & 2 & -8 & 16 & -2 & 4 & -1 & 1 & -1 & 2 & -1 & 1 \\ 1 & -2 & 1 & -1 & 2 & -4 & 1 & -1 & 1 & -1 & 1 & -2 & 2 & -2 & 4 & -8 \\ -2 & 4 & -2 & 2 & -2 & 4 & -1 & 1 & -1 & 1 & -1 & 2 & -4 & 4 & -8 & 16 \\ 1 & -1 & 2 & -4 & 1 & -1 & 1 & -2 & 1 & -2 & 1 & -1 & 4 & -8 & 2 & -2 \\ -1 & 1 & -2 & 4 & -1 & 1 & -1 & 2 & -2 & 4 & -2 & 2 & -8 & 16 & -4 & 4 \\ 2 & -4 & 1 & -1 & 1 & -2 & 1 & -1 & 2 & -2 & 4 & -8 & 1 & -1 & 1 & -2 \\ -2 & 4 & -1 & 1 & -1 & 2 & -1 & 1 & -4 & 4 & -8 & 16 & -2 & 2 & -2 & 4 \\ 1 & -1 & 1 & -2 & 1 & -1 & 2 & -4 & 4 & -8 & 2 & -2 & 1 & -2 & 1 & -1 \\ -1 & 1 & -1 & 2 & -1 & 1 & -2 & 4 & -8 & 16 & -4 & 4 & -2 & 4 & -2 & 2 \\ 1 & -2 & 1 & -1 & 4 & -8 & 2 & -2 & 1 & -1 & 1 & -2 & 1 & -1 & 2 & -4 \\ -1 & 2 & -1 & 1 & -8 & 16 & -4 & 4 & -2 & 2 & -2 & 4 & -1 & 1 & -2 & 4 \\ 4 & -8 & 2 & -2 & 1 & -2 & 1 & -1 & 1 & -2 & 1 & -1 & 2 & -4 & 1 & -1 \\ -8 & 16 & -4 & 4 & -2 & 4 & -2 & 2 & -1 & 2 & -1 & 1 & -2 & 4 & -1 & 1 \\ 1 & -1 & 4 & -8 & 2 & -2 & 1 & -2 & 1 & -1 & 2 & -4 & 1 & -1 & 1 & -2 \\ -2 & 2 & -8 & 16 & -4 & 4 & -2 & 4 & -1 & 1 & -2 & 4 & -1 & 1 & -1 & 2 \end{pmatrix}$$

# 4 SAFER++

SAFER++ is a cipher with two variants: a 128-bit block version with a variable key size of 128 or 256 bits, and a 64-bit (legacy) version with 128-bit key. Both ciphers were designed by Massey, Khachatrian and Kuregian [9] for the NESSIE Project [10].

The PHT in a SAFER++ round, with input $X = (x_1, \ldots, x_{16})$, has

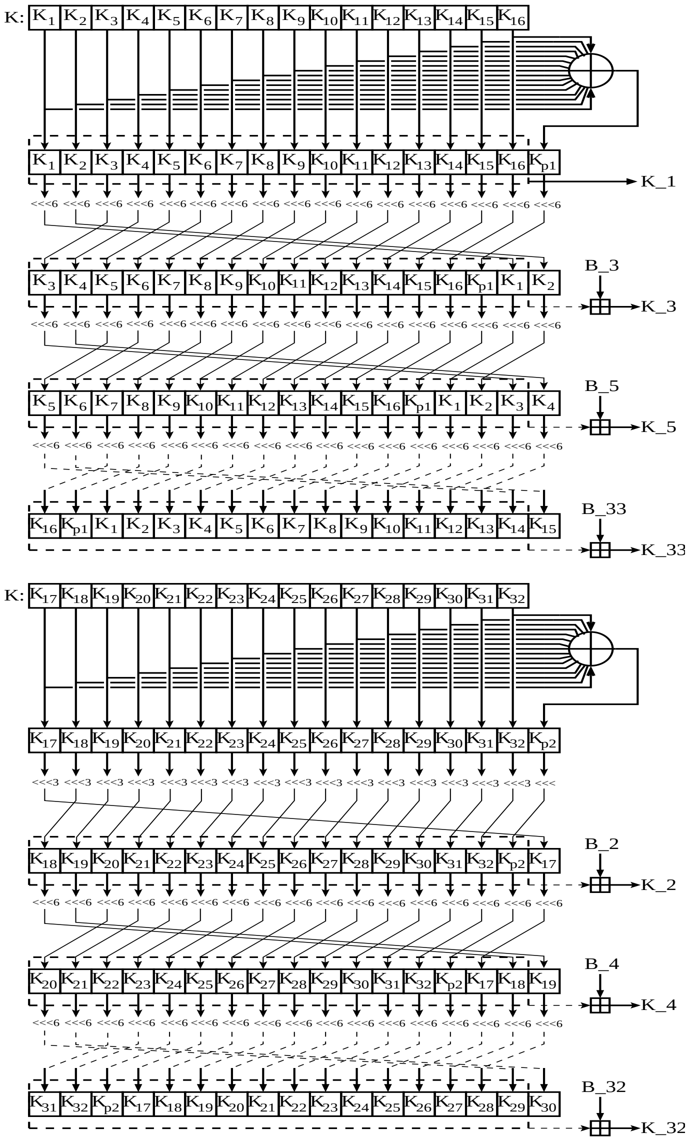


Figure 3: Tweaked key schedule of SAFER+ (KR: Key Register; $K_{p1}, K_{p2}$ key parity bytes).

output given by $Y = (y_1, \ldots, y_{16})$, where $Y = X \cdot M_{++}$, and $M_{++}$ is:

$$M_{++} = \begin{pmatrix}
1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & 2 & 2 & 1 & 1 & 2 & 1 \\
2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 2 & 4 & 2 & 2 \\
2 & 2 & 4 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 \\
4 & 2 & 2 & 2 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 \\
1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 2 & 4 & 2 & 2 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 1 & 1 & 1 \\
1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\
1 & 1 & 2 & 1 & 4 & 2 & 2 & 2 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 2 & 4 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 \\
1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 2 & 4 & 2 \\
2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 4 & 2 & 2 & 2 \\
2 & 4 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\
2 & 1 & 1 & 1 & 2 & 2 & 4 & 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 \\
1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

The inverse linear transformation uses the inverse matrix $M_{++}^{-1}$:

$$M_{++}^{-1} = \begin{pmatrix}
0 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & -4 & 0 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & -1 & 16 & -1 & 0 & -4 & 1 & 0 & -4 & 0 & 1 & -4 & -1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 \\
1 & 0 & 0 & -1 & 0 & 0 & 0 & -4 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 1 & -1 & 0 & 1 & 1 & 0 \\
-4 & 0 & 0 & 1 & 0 & 0 & 0 & 16 & -1 & -1 & -4 & 1 & 0 & -4 & -1 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & -4 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 1 & -1 \\
0 & 1 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 1 & 0 \\
-1 & -4 & 0 & 1 & -4 & -1 & -1 & 1 & 0 & 0 & 0 & 16 & 0 & 0 & -4 & 1 \\
0 & 0 & 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -4 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & -4 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -4 \\
0 & -1 & -4 & 1 & 0 & -4 & 0 & 1 & -4 & 0 & -1 & 1 & -1 & 0 & 0 & 16
\end{pmatrix}$$

The key schedule of 128-bit block SAFER++ follows the same procedure as the one for the tweaked key schedule of SAFER+ (Fig. 3) but with the appropriate number of rounds. For 128-bit keys, the same key is used to generated odd- and even-numbered round subkeys. For 256-bit keys, the first half (16 key bytes) is used to generate the odd-numbered subkeys and the second half is used for the even-numbered subkeys.

# 5 Impossible Differential Attacks

All of our impossible differential attacks use the following difference operator:

$$X' = \Delta X = (X - X^*) \bmod 256.$$

The reason for this choice of difference is that it leads to many truncated differentials that propagate across the PHT layers of the various ciphers with probability one. Our attacks distinguish only zero from non-zero differences in a similar approach as of Knudsen in [5] (the exact values on the non-zero difference is not important). Our attacks use the fact that some subkeys map to the same key bytes according to the key schedules of the different SAFER ciphers. Even though the key bytes are individually rotated and added with (known) constants, there is little interaction between key bytes. This has two consequences: first, once a subkey, that is not the parity byte, is guessed, it can be mapped directly to a master key byte; second, if the same key byte (or a rotated instance) is to be recovered at both ends of the cipher, the complexity of key-recovery decreases because there are less subkeys to find.

## 5.1 Impossible Differential Attack on SAFER SK

The ID attack on the SAFER SK ciphers use the following truncated differentials: the first one has difference $(a, 0, 0, -a, b, 0, 0, -b)$, where $a, b \neq 0$, at the input to the ADD/XOR key mixing layer of the first round, and causes with probability one the difference $(6a + 3b, 3a + 3b, 2a + b, a + b, 2a + b, a + b, 0, 0)$ after the PHT layer; the second truncated differential has input difference $(0, c, -c, 0, 0, d, -d, 0)$ at the output of the XOR/ADD key mixing layer of the third round, and causes (in the decryption direction) the difference $(0, e, 0, f, g, 0, h, 0)$, where $e, f, g, h \neq 0$, with probability one. The contradiction arises in four output byte differences of the first differential. Three of them are non-zero, and the corresponding bytes from the second differential (propagating from bottom up) are zero. The fourth byte from the first differential and the corresponding one from the second differential have the opposite status. This 1.75-round impossible differential is placed between two NL layers, so that subkeys standing before or after non-zero differences can be recovered. Thus, the attack covers 2.75 rounds. The attack proceeds as follows. Choose a structure with $2^{32}$ plaintexts, in which the 2nd, 3rd, 6th and 7th bytes are identical, and the 1st, 4th, 5th, and 8th bytes assume all possible values. There are about $2^{32} \cdot (2^{32} - 1)/2 \approx 2^{63}$ plaintext pairs with difference $(P'_1, 0, 0, P'_4, P'_5, 0, 0, P'_8)$, where $P'_1$, $P'_4$, $P'_5$, $P'_8 \neq 0$. Collect about $2^{31}$ pairs with ciphertext difference zero in the 1st, 4th, 5th and 8th bytes. For each such pair try all $2^{32}$ possible subkeys $K_1^1$, $K_1^4$, $K_1^5$, $K_1^8$ and partially encrypt $P_1$, $P_4$, $P_5$, $P_8$ (the corresponding bytes of the two plaintexts in the pair) across the NL layer. Collect about $2^{16}$ possible 32-bit subkeys satisfying the non-zero differences $(a, 0, 0, -a, b, 0, 0, -b)$ at the input to the ADD/XOR key layer. This take $2 \cdot 2^9$ time and $2 \cdot 2^8$ memory complexity. Try all $2^{32}$ subkeys $K_6^2$, $K_6^3$, $K_6^6$, $K_6^7$ in each of the two ciphertexts of the pair. Collect about $2^{16}$ 32-bit subkeys that result in difference $(0, c, -c, 0, 0, d, -d, 0)$. This takes $2 \cdot 2^9$ time, and $2 \cdot 2^8$ memory. Make a joint list of $2^{32}$ 56-bit subkeys ($K_1^1$, $K_1^4$, $K_1^5$, $K_1^8$, $K_6^2$, $K_6^6$, $K_6^7$) from both steps ($K_6^3 = \text{ROL}_7(K_1^8) \boxplus B_6^3$ from the key schedule). These subkeys cannot be the correct values because they satisfy a text pair of the impossible differential. Each pair suggests a list of about $2^{32}$ incorrect 56-bit subkeys. There are about $2^{31}$ pairs per structure. Using one structure, the expected number of wrong subkeys remaining is: $2^{56}(1 - \frac{2^{32}}{2^{56}})^{2^{31}} = 2^{56}(1 - 2^{-24})^{2^{31}} \approx 2^{56} \cdot e^{-128} \approx 2^{-128.66}$. So, the correct subkey can be uniquely identified. The $2^{31}$ pairs correspond to $2^{32}$ chosen plaintexts. The memory complexity is $2^{50}$ blocks for the steps and joint list of keys. The time complexity is equivalent to $2^{31} \cdot (2^{10} + 2^{10}) \approx 2^{42}$ half-round SAFER SK-64 computations. The remaining $64 - 56 = 8$ bits can be recovered by exhaustive search.

In the case of SAFER SK-128, the eight subkeys searched come from different key bytes (according to the key schedule). Using $2^7$ structures, the number of remaining wrong subkeys is: $2^{64}(1 - \frac{2^{32}}{2^{64}})^{2^{31} \cdot 2^7} = 2^{64}(1 - 2^{-32})^{2^{32} \cdot 2^6} \approx 2^{64} \cdot e^{-64} \approx 2^{-28.33}$. The $2^{31} \cdot 2^7 = 2^{38}$ pairs correspond to $2^{7+32} = 2^{39}$ chosen plaintexts. Memory complexity is about $2^{58}$ blocks for the individual steps and joint list. Time complexity is equivalent to $2^{31+7} \cdot 2^{11} = 2^{49}$ half-round SAFER SK-128 computations. The remaining $128 - 64 = 64$ key bits can be computed by exhaustive search, resulting in total complexity $2^{64}$. An attack on 3.75-round SAFER SK-64 consists in guessing the subkeys in an additional half-round at the top and at the bottom of 2.75-round SAFER SK-64, and performing the previous attack, which starts after the first half-round, with the key-mixing layer of $K_2^i$, and finishes with an output transformation with $K_9^i$, $1 \leq i \leq 8$ (Fig. 4). Due to the difference operator, only the xor-combined subkey bytes are recovered. Due to the key schedule of SAFER SK-64, the subkeys $K_2^i$ and $K_9^i$ can be mapped to the same master key bytes: $K_3, K_4, K_7, K_8$. The attack steps are:

- guess the subkeys $(K_9^1, K_9^4, K_9^5, K_9^8)$, or $(K_2^7, K_2^2, K_2^3, K_2^6)$, encrypt the top half-round and decrypt the bottom half-round of 3.75-round SAFER SK-64.
- for each 32-bit subkey guess, and for each structure in an attack on 2.75 rounds, find all pairs with zero difference in the 1st, 4th, 5th, 8th bytes at the output of the third round.
- given $K_2^2$, the value of $K_3^1$ is known as $\mathrm{ROL}_6(K_2^2) \boxplus b_3^1$. Then $K_4^3$ can be guessed such that it results in the same difference after the NL layer of the 2nd round. Similarly, given $K_9^8$ (or $K_2^6$), the value of $K_3^5$ is determined. Then $K_3^8$ can be guessed such that it causes the same difference after the NL layer of the 2nd round. On average, only one value of $K_4^3$ and one value of $K_3^8$ might be suggested per pair.
- obtain about $2^8$ subkeys $(K_8^2, K_8^3)$ that cause the same difference before the NL layer of the 3rd round. In total, 32 key bits of the 40 subkey bits $(K_3^4, K_3^8, K_8^7, K_8^2, K_8^3)$ are suggested per pair, and these values are impossible. After one structure, the remaining wrong subkeys are: $2^{40} \cdot (1 - \frac{2^{32}}{2^{40}})^{2^{31}} = 2^{40} \cdot (1 - 2^{-8})^{2^{31}} \approx 2^{40} \cdot e^{-2^{23}} \approx e^{-8388580}$. So, only the correct subkey might be suggested.

The attack requires one text structure, that is, $2^{32}$ chosen plaintexts. Time complexity consists of $2 \cdot 2^{32} \cdot 2^{32} = 2^{65}$ half-round computations which is equivalent to about $2^{62}$ 3.75-round computations. Data complexity is $2^{40}$ bits for the lists of impossible subkeys.

### 5.2 Impossible Differential Attack on SAFER+/128

For SAFER+ with a 128-bit key, the impossible differential consists of the following truncated differentials, that hold with probability one: the first one has input difference $(a, 0, 0, b, c, 0, 0, -b, d, 0, 0, -d, -a, 0, 0, -c)$, with $a, b, c, d \neq 0$, at the input to the ADD/XOR key mixing layer of the first round, and causes the difference $(-2a - 4c, 0, -a + c + 12d, -a + b + 6d, 15a - 2b + 3c - d, 7a - b + c - d, -2a - 6b + 3c + d, -3a - 3b + c + d, 3a + 2b + 14c, a + b + 6c, 7b - c + 3d, 3b + 3d, -a - b - c - 4d, -b - 2d, -12a - 2c + 2d, -4a + d)$ at the output of the PHT layer of the first round; the second

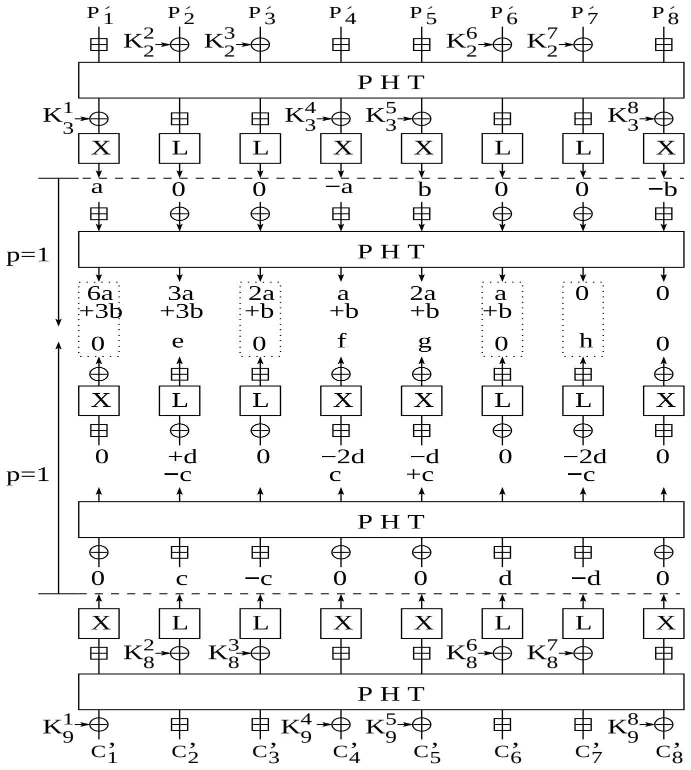
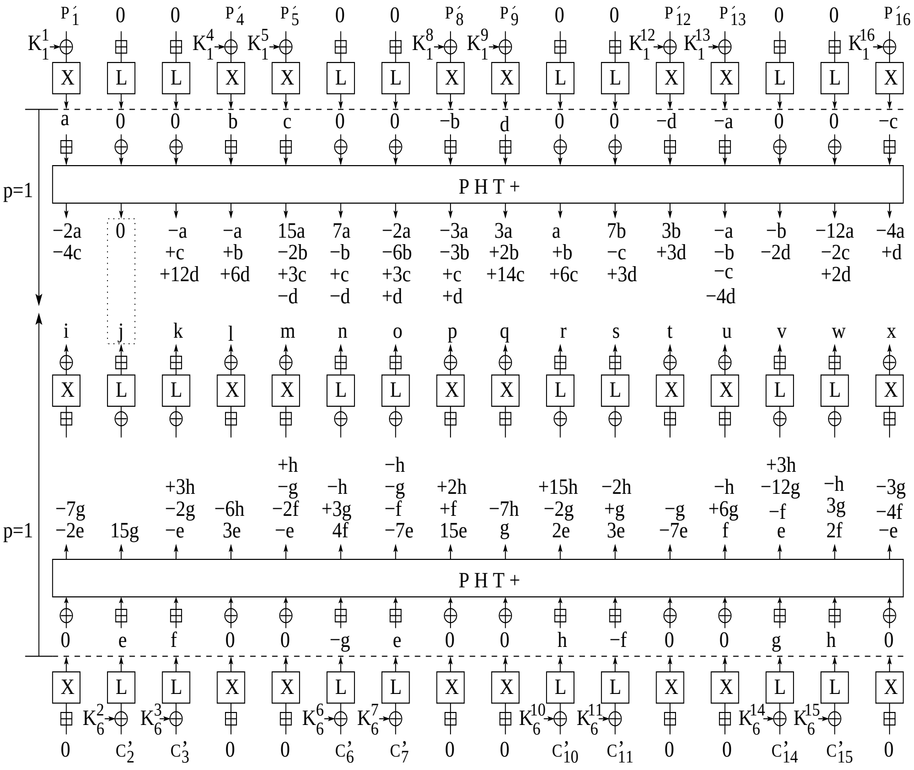

P'1 P'2 P'3 P'4 P'5 P'6 P'7 P'8
$K_2^2$ $K_2^3$ $K_2^6$ $K_2^7$
P H T
$K_3^1$ $K_3^4$ $K_3^5$ $K_3^8$
X L L X X L L X
a 0 0 −a b 0 0 −b
p=1
P H T
6a +3b
3a +3b
2a +b
a +b
2a +b
a +b
0
0
0 e 0 f g 0 h 0
X L L X X L L X
0
+d −c
0
−2d c
−d +c
0
−2d −c
0
p=1
P H T
0 c −c 0 0 d −d 0
X L L X X L L X
$K_8^2$ $K_8^3$ $K_8^6$ $K_8^7$
P H T
$K_9^1$ $K_9^4$ $K_9^5$ $K_9^8$
C'1 C'2 C'3 C'4 C'5 C'6 C'7 C'8

Figure 4: Impossible differential attack on 3.75-round SAFER SK.



P´1 0 0 P´4 P´5 0 0 P´8 P´9 0 0 P´12 P´13 0 0 P´16
K11 K14 K15 K18 K19 K112 K113 K116
X L L X X L L X X L L X X L L X
a 0 0 b c 0 0 −b d 0 0 −d −a 0 0 −c
p=1
PHT+
−2a −4c
0
−a +c +12d
−a +b +6d
15a −2b +3c −d
7a −b +c −d
−2a −6b +3c +d
−3a −3b +c +d
3a +2b +14c
a +b +6c
7b −c +3d
3b +3d
−a −b −c −4d
−b −2d
−12a −2c +2d
−4a +d
i j k l m n o p q r s t u v w x
X L L X X L L X X L L X X L L X
−7g −2e
15g
+3h −2g −e
−6h 3e
+h −g −2f −e
−h +3g 4f
−h −g −f −7e
+2h +f 15e
−7h g
+15h −2g 2e
−2h +g 3e
−g −7e
−h +6g f
+3h −12g −f e
−h 3g 2f
−3g −4f −e
p=1
PHT+
0 e f 0 0 −g e 0 0 h −f 0 0 g h 0
X L L X X L L X X L L X X L L X
K62 K63 K66 K67 K610 K611 K614 K615
0 C'2 C'3 0 0 C'6 C'7 0 0 C'10 C'11 0 0 C'14 C'15 0

Figure 5: Impossible differential attack on 2.75-round SAFER+.

differential has difference (0, $e$, $f$, 0, 0, $-g$, $e$, 0, 0, $h$, $-f$, 0, 0, $g$, $h$, 0) at the output of the XOR/ADD key mixing layer of the third round and causes (in the decryption direction) the difference ($i$,$j$,$k$,$l$,$m$,$n$,$o$,$p$,$q$,$r$,$s$,$t$,$u$,$v$,$w$,$x$), with $i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x \neq 0$ at the input to the second round (Fig. 5).

The impossible differential has input difference ($a$, 0, 0, $b$, $c$, 0, 0, $-b$, $d$, 0, 0, $-d$, $-a$, 0, 0, $-c$) which cannot cause the output difference (0, $e$, $f$, 0, 0, $-g$, $e$, 0, 0, $h$, $-f$, 0, 0, $g$, $h$, 0) after 1.75 rounds. The contradiction is in one byte difference: the 2nd output byte difference from the first differential is zero, while the corresponding byte of the second differential is always non-zero (because $15g \equiv 0 \bmod 256 \Leftrightarrow g = 0$). This 1.75-round impossible differential is placed between two NL layers, so that some subkeys surrounding them can be recovered. Thus, the attack covers 2.75 rounds.

The attack proceeds as follows. Choose a structure of about $2^{64}$ texts with $P_2$, $P_3$, $P_6$, $P_7$, $P_{10}$, $P_{11}$, $P_{14}$, $P_{15}$ set to fixed values, and with all possible values for $P_1$, $P_4$, $P_5$, $P_8$, $P_9$, $P_{12}$, $P_{13}$, $P_{16}$. There are about $2^{64} \cdot (2^{64} - 1)/2 \approx 2^{127}$ pairs in such a structure. Collect about $2^{127} \cdot 2^{-64} = 2^{63}$ pairs that have difference zero in the 1st, 4th, 5th, 8th, 9th, 12th, 13th, and 16th bytes of the ciphertext difference. For each pair, try all $2^{64}$ possible subkeys $K_1^1$, $K_1^4$, $K_1^5$, $K_1^8$, $K_1^9$, $K_1^{12}$, $K_1^{13}$, $K_1^{16}$, and encrypt partially the corresponding plaintext bytes across the NL layer. Collect about $2^{32}$ 64-bit subkeys satisfying the difference ($a$, 0, 0, $b$, $c$, 0, 0, $-b$, $d$, 0, 0, $-d$, $-a$, 0, 0, $-c$) after the first NL layer. This can be done in $4 \cdot 2^{10}$ time and $4 \cdot 2^8$ memory complexity. Try all $2^{64}$ possible subkeys $K_6^2$, $K_6^3$, $K_6^6$, $K_6^7$, $K_6^{10}$, $K_6^{11}$, $K_6^{14}$, $K_6^{15}$ and decrypt the corresponding bytes in each ciphertext of the pairs across one NL layer. Collect about $2^{32}$ subkeys that result in difference (0, $e$, $f$, 0, 0, $-g$, $e$, 0, 0, $h$, $-f$, 0, 0, $g$, $h$, 0) at the input to the 3rd NL layer. This can be done in $4 \cdot 2^{10}$ time and $4 \cdot 2^8$ memory. Make a list of $2^{64}$ 104-bit subkeys $K_1^1$, $K_1^4$, $K_1^5$, $K_1^8$, $K_1^9$, $K_1^{12}$, $K_1^{13}$, $K_1^{16}$, $K_6^2$, $K_6^6$, $K_6^{10}$, $K_6^{14}$, $K_6^{15}$ that result from the previous steps. Due to the tweaked key schedule of SAFER+, $K_6^3 = \mathrm{ROL}_7(K_1^8) \boxplus B_6^3$, $K_6^7 = \mathrm{ROL}_7(K_1^{12}) \boxplus B_6^7$, $K_6^{11} = \mathrm{ROL}_7(K_1^{16}) \boxplus B_6^{11}$, for a 128-bit master key. These subkeys cannot be the correct value because they lead to a pair of the impossible differential. Each pair suggests a list of $2^{64}$ wrong 104-bit subkeys. With $2^{47}$ pairs (out of the $2^{63}$ available in a structure), the number of wrong subkeys remaining is: $2^{104}(1 - \frac{2^{64}}{2^{104}})^{2^{47}} = 2^{104}(1 - 2^{-40})^{2^{47}} \approx 2^{104} \cdot e^{-128} \approx 2^{-80.66}$.

The $2^{47}$ pairs needed correspond to $2^{64}$ chosen plaintexts. The memory complexity is about $2^{97}$ blocks for the individual steps and joint list of wrong subkeys. The time complexity is equivalent to $2^{47}(2^{12} + 2^{12}) = 2^{60}$ half-round SAFER+ computations. The remaining $128 - 104 = 24$ key bits can be computed by exhaustive search.

### 5.3 Impossible Differential Attack on SAFER++/128

For SAFER++ with a 128-bit key, the impossible differential consists of the following differentials, that hold with probability one: the first one has input difference ($a$, 0, 0, $-a$, $b$, 0, 0, $-b$, $c$, 0, 0, $-c$, $d$, 0, 0, $-d$), with $a, b, c, d \neq 0$, at the input to the ADD/XOR key mixing layer of the first round, and causes the difference $(3b - c, a, b + c - d, b, 3c - d, -a + c + d, b, c, 3a - b, a + c - d, d, a, -a + 3d, b - c + d, a - b + d, d)$ at the output of the PHT layer of the first round; the second differential has difference (0, $e$, $e$, 0, 0, $f$, $g$, 0, 0, $-g$, $-f$, 0, 0, $h$, $-h$, 0) at the input of the third NL layer, and causes (in the decryption direction) the difference ($i$, $j$, $k$, 0, $l$, $m$, $n$, $o$, 0, $p$, $q$, $r$, 0, $s$, $t$, $u$), with $i, j, k, l, m, n, o, p, q, r, s, t, u \neq 0$ at the input to the second round

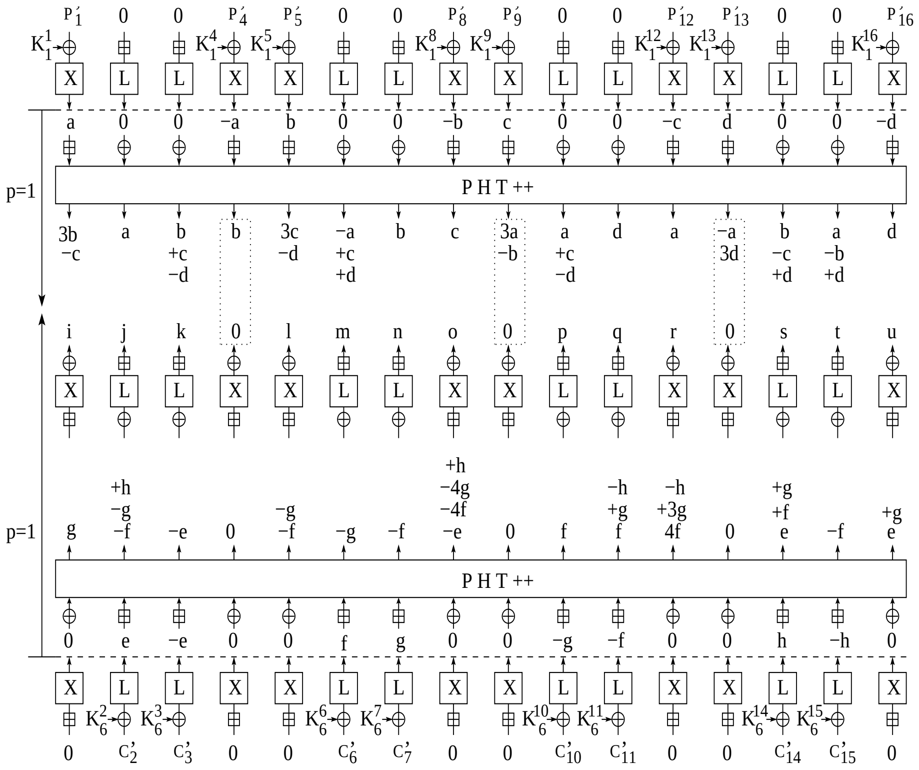

P'1 0 0 P'4 P'5 0 0 P'8 P'9 0 0 P'12 P'13 0 0 P'16
K1 1 K4 1 K5 1 K8 1 K9 1 K12 1 K13 1 K16 1
X L L X X L L X X L L X X L L X
a 0 0 −a b 0 0 −b c 0 0 −c d 0 0 −d
p=1
P H T ++
3b −c
a
b +c −d
b
3c −d
−a +c +d
b
c
3a −b
a +c −d
d
a
−a 3d
b −c +d
a −b +d
d
i j k 0 l m n o 0 p q r 0 s t u
X L L X X L L X X L L X X L L X
g
+h −g −f
−e
0
−g −f
−g
−f
+h −4g −4f −e
0
f
−h +g f
−h +3g 4f
0
+g +f e
−f
+g e
p=1
P H T ++
0 e −e 0 0 f g 0 0 −g −f 0 0 h −h 0
X L L X X L L X X L L X X L L X
K2 6 K3 6 K6 6 K7 6 K10 6 K11 6 K14 6 K15 6
0 C'2 C'3 0 0 C'6 C'7 0 0 C'10 C'11 0 0 C'14 C'15 0

Figure 6: Impossible differential attack on 2.75-round SAFER++.

(Fig. 6). The impossible differential has input difference $(a, 0, 0, -a, b, 0, 0, -b, c, 0, 0, -c, d, 0, 0, -d)$ and cannot cause the output difference $(0, e, -e, 0, 0, f, g, 0, 0, -g, -f, 0, 0, h, -h, 0)$ after 1.75 rounds. The contradiction is in (at least) one byte difference: the 4th output byte difference from the first differential is non-zero, $b$, while the corresponding byte of the second differential is always zero. This 1.75-round impossible differential is placed between two NL layers, so that some subkeys surrounding them can be recovered. Thus, the attack covers 2.75 rounds and proceeds similarly to that for SAFER+/128, with the key schedule of SAFER++ allowing to reduce the key-recovery from 128 down to 104 key bits. The data, memory and time complexities are the same as for the attack on SAFER+/128.

# 6 Square Attacks on SAFER Ciphers

In [5], Knudsen described a Square attack (although not with this name) on 3.25-round SAFER K/SK, using the addition in $\mathbb{Z}_{256}$ as the notion of integral [6]. A $\Lambda$-set pattern [4] consists of the first seven plaintext bytes fixed, and the 8th byte assuming all possible 8-bit values (active). It will cause the first text byte after 2.75 rounds to be active. This property can be used to recover four subkeys $K_1^7$, $K_4^7$, $K_5^7$, $K_6^7$, because the additive subkeys can be pushed up the $\text{PHT}^{-1}$ layer. The time complexity of the attack is $2^{32} \cdot 2^8 + 2^{24} \cdot 2^8 + 2^{16} \cdot 2^8 + 2^8 \cdot 2^8 + 1 \cdot 2^8 \approx 2^{40}$ half-round computations, or approximately $\frac{1}{6} \cdot 2^{40} \approx 2^{38}$ 3.25-round SAFER K/SK computations. Five $\Lambda$-sets are needed to discard wrong key candidates, since the single active byte after 2.75 rounds provides an 8-bit condition. For SAFER+ a similar attack can be applied to 3.25 rounds using a $\Lambda$-set with the 4th plaintext byte active and all other 15 bytes fixed (passive). After 2.75 rounds, the 1st, 3rd, 5th, and 13th text bytes are active. Eight key bytes: $K_7^1$, $K_7^4$, $K_7^5$, $K_7^8$, $K_7^9$, $K_7^{12}$, $K_7^{13}$, $K_7^{16}$ can be recovered, with an effort of $2^{64} \cdot 2^8 + 2^{32} \cdot 2^8 + 1 \cdot 2^8 \approx 2^{72}$ half-round computations, which is equivalent to $\frac{1}{6} \cdot 2^{72} \approx 2^{70}$ 3.25-round SAFER+ computations. The data complexity is three $\Lambda$-sets, or $3 \cdot 2^8$ chosen plaintexts. For SAFER++, similarly, a Square attack on 3.25 rounds can use a $\Lambda$-set with the 1st plaintext byte active and all other 15 bytes passive. After 2.75 rounds, the 4th, 8th, 12th, and 16th text bytes are active. Eight key bytes: $K_7^1$, $K_7^4$, $K_7^5$, $K_7^8$, $K_7^9$, $K_7^{12}$, $K_7^{13}$, $K_7^{16}$ can be recovered, with an effort about $2^{72}$ half-round computations, which is equivalent to $2^{70}$ 3.25-round SAFER++ computations. The data complexity is $3 \cdot 2^8$ chosen plaintexts.

Table 1: Complexities of some attacks on reduced-round SAFER ciphers.

| Cipher | Block Size | Key Size | Attack Type | #Rounds | Complexities |  |  |
|---|---|---|---|---|---|---|---|
|  |  |  |  |  | Data | Memory | Time |
| SAFER SK-64 | 64 | 64 | ID | 2.75 | $2^{32}$ | $2^{50}$ | $2^{42}$ |
|  | 64 | 64 | Square | 3.25 | $2^{10.3}$ | $2^{10.3}$ | $2^{38}$ |
|  | 64 | 64 | ID | 3.75 | $2^{32}$ | $2^{40}$ | $2^{62}$ |
| SAFER SK-128 | 64 | 128 | ID | 2.75 | $2^{39}$ | $2^{58}$ | $2^{64}$ |
|  | 64 | 128 | Square | 3.25 | $2^{10.3}$ | $2^{10.3}$ | $2^{38}$ |
| SAFER+ | 128 | 128 | ID | 2.75 | $2^{64}$ | $2^{97}$ | $2^{60}$ |
|  | 128 | 128 | Square | 3.25 | $2^{9.6}$ | $2^{9.6}$ | $2^{70}$ |
| SAFER++ | 128 | 128 | ID | 2.75 | $2^{64}$ | $2^{97}$ | $2^{60}$ |
|  | 128 | 128 | Square | 3.25 | $2^{9.6}$ | $2^{9.6}$ | $2^{70}$ |

## 7 Conclusion

This paper reports on an Impossible Differential analysis of reduced-round variants of some SAFER ciphers. A summary of the attack complexities is shown in Table 1, compared to Square attacks by Knudsen [5], and to Square attacks to SAFER+ and SAFER++ adapted by the authors. The ID attacks on SAFER ciphers demand too much memory and time. This is an indication that ID attacks work better on ciphers with slow diffusion as Skipjack [2].

## 8 Acknowledgements

The authors would like to thank A. Biryukov for the many explanations concerning the Impossible Differential technique.

## References

[1] AES. The Advanced Encryption Standard Development Process. http://csrc.nist.gov/encryption/aes/, 1997.

[2] E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials. Tech Report CS0947 revised, Technion, CS Dept., 1998.

[3] E. Biham, A. Biryukov, and A. Shamir. Miss-in-the-Middle Attacks on IDEA, Khufu and Khafre. In L.R. Knudsen, editor, *6th Fast Software Encryption Workshop*, LNCS 1636, pages 124–138. Springer-Verlag, 1999.

[4] J. Daemen, L.R. Knudsen, and V. Rijmen. The Block Cipher SQUARE. In E. Biham, editor, *4th Fast Software Encryption Workshop*, LNCS 1267, pages 149–165. Springer-Verlag, 1997.

[5] L.R. Knudsen. A Detailed Analysis of SAFER K. *Journal of Cryptology*, 13(4):417–436, 2000.

[6] L.R. Knudsen and D. Wagner. Integral Cryptanalysis. In J. Daemen and V. Rijmen, editors, *9th Fast Software Encryption Workshop*, LNCS 2365, pages 112–127. Springer-Verlag, 2002.

[7] J.L. Massey. SAFER K–64: a Byte-Oriented Block-Ciphering Algorithm. In R. Anderson, editor, *1st Fast Software Encryption Workshop*, LNCS 809, pages 1–17. Springer-Verlag, 1994.

[8] J.L. Massey, G.H. Khachatrian, and M.K. Kuregian. Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard. 1st AES Conference, California, USA, Jun 1998. http://csrc.nist.gov/encryption/aes/.

[9] J.L. Massey, G.H. Khachatrian, and M.K. Kuregian. The SAFER++ Block Encryption Algorithm. 1st NESSIE Workshop, Heverlee, Belgium, Nov 2000. http://cryptonessie.org.

[10] NESSIE. New European Schemes for Signatures, Integrity and Encryption, Jan 2000. http://cryptonessie.org.