New Impossible Differential Attack on SAFER$_+$ and SAFER$_{++}$

Jingyuan Zhao[1,2], Meiqin Wang[1,2] Jiazhe Chen[1,2], Yuliang Zheng[1,2,3]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China
[2] School of Mathematics, Shandong University, Jinan 250100, China
[3] Department of Software and Information Systems, UNC Charlotte, 9201 University City Blvd, Charlotte, NC 28223, USA
mqwang@sdu.edu.cn

Abstract. SAFER+ was a candidate block cipher for AES with 128-bit block size and a variable key sizes of 128, 192 or 256 bits. Bluetooth uses customized versions of SAFER+ for security. The numbers of rounds for SAFER+ with key sizes of 128, 192 and 256 are 8, 12 and 16, respectively. SAFER++, a variant of SAFER+, was among the cryptographic primitives selected for the second phase of the NESSIE project. The block size is 128 bits and the key size can take either 128 or 256 bits. The number of rounds for SAFER++ is 7 for keys of 128 bits, and 10 for keys of 256 bits. Both ciphers use PHT as their linear transformation. In this paper, we take advantage of properties of PHT and S-boxes to identify 3.75-round impossible differentials for SAFER++ and 2.75-round impossible differentials for SAFER+, which result in impossible differential attacks on 4-round SAFER+/128(256), 5-round SAFER++/128 and 5.5-round SAFER++/256. Our attacks significantly improve previously known impossible differential attacks on 3.75-round SAFER+/128(256) and SAFER++/128(256). Our attacks on SAFER+/128(256) and SAFER++/128(256) represent the best currently known attack in terms of the number of rounds.

Keywords: SAFER+, SAFER++, Impossible Differential, PHT, Bluetooth

1 Introduction

SAFER+, designed by Massey, Khachatrian and Kuregian, was a candidate block cipher for AES with 128-bit block size and a variable key sizes of 128, 192 or 256 bits, denoted by SAFER+/128, SAFER+/192 and SAFER+/256, respectively [8]. Since some weaknesses to the key schedules of SAFER+/192 and SAFER+/256 were discovered, Massey et al. changed the key schedule algorithms later. In this paper, we will use the remedied key schedule algorithms as in [12]. Bluetooth uses custom algorithms based on SAFER+ for key derivation and authentication as MAC [4]. SAFER++ was submitted to the NESSIE project [13] and was among the primitives selected for the second phase of this project [9]. The block size is 128-bit and the key size can be taken as 128-bit and 256-bit. The two ciphers have common S-boxes derived from exponentiation and discrete logarithm functions and share the Pseudo-Hadamard-like mixing transforms (PHT) but have different ways to use it. They also share the methods to perform key-mixing with two-commutative operations.

Several cryptanalytic results on SAFER+ and SAFER++ have been published. Nakahara et al. gave the non-homomorphic linear cryptanalysis for 3.25 rounds of SAFER+/128 and 3 rounds of SAFER++/128 and SAFER++/256 [10,11]. Piret et al. gave the integral cryptanalysis for 4.25 rounds of SAFER++/128 and 4.75 rounds of SAFER++/256 [14]. Biryukov

et al. gave the multiset attack on 4.5 rounds of SAFER++/128 and the boomerang attack on 5.5 rounds of SAFER++/128.

For the impossible differential cryptanalysis, Nakahara et.al also gave the impossible differential cryptanalysis for 2.75 rounds of SAFER+/128 and SAFER++/128 [11,12]. Then Behnam et al. claimed they could attack 4 rounds of SAFER++/128 with the impossible differential cryptanalysis [1], however, their attack only worked for 4-round SAFER++/128 without the final whitening-key layer, so their attack was a 3.75-round attack. Zheng et al. gave the impossible differential attacks on 3.75 rounds of SAFER+/128 (SAFER+/256) and 3.75 rounds of SAFER++/128 (SAFER++/256).

The impossible differential attack, which was independently proposed by Biham et al. [2] and Knudsen [5], is a popular cryptanalytic method. The attack starts with finding an input difference that can never result in an output difference, which will produce an impossible differential. By adding rounds before and/or after the impossible differential, one can collect pairs with certain plaintext and ciphertext differences. If there exists a pair that meets the input and output values of the impossible differential under some subkey bits, these bits must be wrong. In this way, we discard as many wrong keys as possible and exhaustively search the rest of the keys, this phase is called key recovery phase. The early abort technique is usually used during the key recovery phase, that is, one does not guess all the subkey bits at once, but guess some subkey bits instead to discard some pairs that do not satisfy certain conditions step by step. In this case, we can discard the unwished pairs as soon as possible to reduce the time complexity.

Our Contributions By delicately choosing the positions and the number of the active S-boxes in the first round, we can identify 3.75 rounds impossible differentials for SAFER++, which are significantly better than the previous 2.75-round impossible differentials [1,16].

At the same time, we also identify 2.75-round impossible differentials for SAFER+. Although our impossible differentials work for the same number of rounds as those in [1,16], they will result in less active S-boxes in the first round or the last round, and then the number of guessed subkey bytes will be reduced during the key recovery phase, so we can attack four full rounds with the final whitening key layer; the attack is better than the 3.75-round attack in [16] and the four-round attack without the final whitening key layer in [1]. Our attacks on bluetooth ciphers SAFER+/128 and SAFER+/256 are the best attacks according to the number of rounds. Specially, our attack on SAFER+/128 is the first attack on half of the full-round SAFER+.

Our attack on SAFER++/128 can work for 5 rounds with the final whitening key layer, which is much better than the previous impossible differential attack for 3.75 rounds in [1,16]. However, the best attack on SAFER++/128 is the boomerang attack for 5.5 rounds [3]. Our attack on SAFER++/256 can work for 5.5 rounds. Although our attack on SAFER++/128 is not as good as those in [3], we greatly improve the impossible differential attack in [1,16] and our attacks are the best chosen plaintext attacks.

The only difference for the components of round functions for SAFER+ and SAFER++ is the linear transformation; the linear transformation of SAFER++ is more complicated than that of SAFER+, so the designers use less rounds for SAFER++ than SAFER+. It seems that the linear transformation for SAFER++ is much secure than SAFER+, however, our attack shows that SAFER++ is less resistant to impossible differential attack than SAFER+, because the diffusion of the inverse linear layer for SAFER++ is much weaker.

We summarize our results of SAFER+ and SAFER++, as well as the major previous results in Table 1.

The rest of the paper is organized as follows. We give the brief descriptions of SAFER+ and SAFER++ in Sect. 2. Section 3 identifies the impossible differentials for SAFER+ and SAFER++. The impossible differential cryptanalysis of SAFER+/128 and SAFER+/256 is

Table 1. Summary of attacks on SAFER+ and SAFER++

Cipher	Attack	#Rounds	Data	Time (Encryptions)	Memory (Bytes)	Source
+/128	ID	2.75	2^{64}CP	2^{58}	2^{104}	[12]
+/128	LNH	3.25	2^{101}KP	2^{141}	2^{108}	[10]
+/128	ID	3.75	2^{78}CP	2^{72}	2^{68}	[16]
+/128	**ID**	**4**	$\mathbf{2^{122.4}CP}$	$\mathbf{2^{121}}$	$\mathbf{2^{87.4}}$	**Sect.4**
+/256	ID	3.75	2^{78}CP	2^{72}	2^{68}	[16]
+/256	**ID**	**4**	$\mathbf{2^{124.4}CP}$	$\mathbf{2^{216}}$	$\mathbf{2^{89.4}}$	**Sect.4**
++/128	LNH	3	2^{81}KP	2^{105}	2^{88}	[10]
++/128	ID	2.75	2^{64}CP	2^{58}	2^{104}	[12]
++/128	Integral	4	2^{64}CP	2^{117}	2^{71}	[14]
++/128	Integral	4.25	–	–	–	[3]
++/128	Multiset	4.5	2^{48}CP	2^{100}	2^{55}	[3]
++/128	Boomerang	5.5	2^{108}CP/ACC	2^{116}	2^{55}	[3]
++/128	ID	3.75	2^{23}CP	2^{84}	2^{75}	[1]
++/128	ID	3.75	2^{78}CP	2^{63}	2^{62}	[16]
++/128	**ID**	**5**	$\mathbf{2^{124}CP}$	$\mathbf{2^{121}}$	$\mathbf{2^{97}}$	**Sect.5**
++/256	LNH	3	2^{81}KP	2^{105}	2^{88}	[10]
++/256	Integral	4	2^{64}CP	2^{149}	2^{71}	[14]
++/256	Integral	4.75	–	–	–	[14]
++/256	ID	3.75	2^{78} CP	2^{71}	2^{70}	[16]
++/256	**ID**	**5.5**	$\mathbf{2^{124}CP}$	$\mathbf{2^{246}}$	$\mathbf{2^{97}}$	**Sect.5**

CP: Chosen Plaintext; KP: Known Plaintext; ACC: Adaptive Chosen Ciphertext
ID: Impossible Differential; LNH: Linear(Non-Homomorphic).

presented in Sect. 4. Section 5 gives the impossible differential cryptanalysis of SAFER++/128 and SAFER++/256. Finally, Sect. 6 concludes this paper.

2 Brief Descriptions of SAFER$_+$ and SAFER$_{++}$

This section contains short descriptions of SAFER+ and SAFER++. For more details, see [8, 9]. Throughout this paper we will number bytes and S-boxes from left to right, starting from 0.

SAFER+ (SAFER++) is a 128-bit SPN block ciphers with variable key sizes of 128, 192 or 256 bits, denoted by SAFER+/128, SAFER+/192 and SAFER+/256 (SAFER++/128 and SAFER++/256). The round function of SAFER+ (SAFER++) consists of an upper key layer, a nonlinear layer, a lower key layer and a linear transformation. After the final round, an additional key-addition whitening similar to the upper key layer is added. The numbers of round of SAFER+/128 and SAFER+/256 are 8 and 16, respectively. The numbers of round of SAFER++/128 and SAFER++/256 are 7 and 10, respectively. Among the components of the round functions of SAFER+ and SAFER++, only the linear transformation is different. SAFER+ uses a 2-point pseudo Hadamard transformation(2-PHT) while SAFER++ uses a 4-point pseudo Hadamard transformation(4-PHT). The linear layer is accordingly denoted by M_+ for SAFER+ and M_{++} for SAFER++.

2.1 The Keyed Non-Linear Layer

Since SAFER+ and SAFER++ are byte-oriented ciphers, the input plaintext block is initially splited into 16 bytes to combine with the 16 bytes subkey. Bytes 0, 3, 4, 7, 8, 11, 12, and

15 of the subkey are XORed to the corresponding bytes of the block, while bytes 1, 2, 5, 6, 9, 10, 13, and 14 of the subkey are combined with the corresponding bytes using addition modulo 256. The nonlinear layer is based on two different 8-to-8 bit functions, X and L,

$$\mathrm{X}(a) = (45^a \bmod 257) \bmod 256,$$
$$\mathrm{L}(a) = \log_{45}{}^a \bmod 257,$$

with the special case that $\mathrm{L}(0) = 128$, making X and L mutually inverse. We call the layer including X and L as S-box layer. In this layer, bytes 0, 3, 4, 7, 8, 11, 12, and 15 are sent through the function X, and L is applied to bytes 1, 2, 5, 6, 9, 10, 13, and 14. The lower key layer mixes a 16-byte subkey to the output blocks from the X and L functions. Bytes 2, 3, 6, 7, 10, 11, 14 and 15 of the subkey are XORed to the corresponding bytes of the block and bytes 1, 4, 5, 8, 9, 12, 13 and 16 of the subkey and blocks are combined using addition modulo 256.

2.2 The Linear Layer

The linear transformation of SAFER+ (SAFER++) is constructed by two parts: the first is a permutation and the second is a 2-PHT(4-PHT) to two group of 2-branch(four group of 4-branch). The 2-PHT(4-PHT) can be implemented with two(six) modular additions. The linear layers can be expressed by matrices $M_+(M_{++})$ and the inverse linear layers are $M_+^{-1}(M_{++}^{-1})$. The matrixes $M_+(M_{++})$ and $M_+^{-1}(M_{++}^{-1})$ are shown in Appendix A.

2.3 The Key Schedule

The key schedule of SAFER++ is same as that of SAFER+ for the same key size and the key schedules of 128-bit and 256-bit master keys are different. Firstly we introduce the 128-bit key schedule: K=(k^1, k^2, k^3, k^4, k^5, k^6, k^7, k^8, k^9, k^{10}, k^{11}, k^{12}, k^{13}, k^{14}, k^{15}, k^{16}) is the 128-bit master key. From the 16 bytes of master key we get the 17-th byte:

$$k^{sp1} = \bigoplus_{i=1}^{16} k^i.$$

Table 2 gives the relations between the subkey and the master key according to which master key byte they depend on for SAFER+/128 and SAFER++/128. In the first column of Table 2, K_{ri} is the subkey where r is the round number and $i = 1$ and $i = 2$ denote the subkey of the upper key layer and the lower key layer, respectively. As we only attack no more than 5 full rounds for SAFER++/128, we only list the relations of the subkey for the first 5.5 rounds in Table 2.

The 256-bit mater key is K=(k^1, k^2, k^3, k^4, k^5, k^6, k^7, k^8, k^9, k^{10}, k^{11}, k^{12}, k^{13}, k^{14}, k^{15}, k^{16}, k^{17}, k^{18}, k^{19}, k^{20}, k^{21}, k^{22}, k^{23}, k^{24}, k^{25}, k^{26}, k^{27}, k^{28}, k^{29}, k^{30}, k^{31}, k^{32}). Different from 128-bit key schedule, the 256-bit master key is splitted into two 128-bit blocks. The first one is used to produce the upper key layer of each round and the final key addition, and the second one is used to produce the lower key layer of each round. k^{sp1} is computed as in SAFER+/128 and SAFER++/128. In addition, another subkey byte k^{sp2} can be computed with

$$k^{sp2} = \bigoplus_{i=17}^{32} k^i.$$

Similarly, Table 3 lists the relations between the subkeys and the master key according to which master key byte they depend on for SAFER+/256 and SAFER++/256. In Table 3, K_{ri} has the same meaning as that in Table 2. As we only attack no more than 5.5 rounds for SAFER++/256, we only list the relations of the subkeys for the first 6 rounds in Table 3.

Table 2. Relations between subkey and master key for SAFER+/128 and SAFER++/128

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
K_{11}	k^1	k^2	k^3	k^4	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}
K_{12}	k^2	k^3	k^4	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}
K_{21}	k^3	k^4	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1
K_{22}	k^4	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2
K_{31}	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3
K_{32}	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4
K_{41}	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5
K_{42}	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5	k^6
K_{51}	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5	k^6	k^7
K_{52}	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5	k^6	k^7	k^8
K_{61}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5	k^6	k^7	k^8	k^9

Table 3. Relations between subkey and master key for SAFER+/256 and SAFER++/256

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
K_{11}	k^1	k^2	k^3	k^4	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}
K_{12}	k^{18}	k^{19}	k^{20}	k^{21}	k^{22}	k^{23}	k^{24}	k^{25}	k^{26}	k^{27}	k^{28}	k^{29}	k^{30}	k^{31}	k^{32}	k^{sp2}
K_{21}	k^3	k^4	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1
K_{22}	k^{20}	k^{21}	k^{22}	k^{23}	k^{24}	k^{25}	k^{26}	k^{27}	k^{28}	k^{29}	k^{30}	k^{31}	k^{32}	k^{sp2}	k^{17}	k^{18}
K_{31}	k^5	k^6	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3
K_{32}	k^{22}	k^{23}	k^{24}	k^{25}	k^{26}	k^{27}	k^{28}	k^{29}	k^{30}	k^{31}	k^{32}	k^{sp2}	k^{17}	k^{18}	k^{19}	k_{20}
K_{41}	k^7	k^8	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5
K_{42}	k^{24}	k^{25}	k^{26}	k^{27}	k^{28}	k^{29}	k^{30}	k^{31}	k^{32}	k^{sp2}	k^{17}	k^{18}	k^{19}	k_{20}	k^{21}	k^{22}
K_{51}	k^9	k^{10}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5	k^6	k^7
K_{52}	k^{26}	k^{27}	k^{28}	k^{29}	k^{30}	k^{31}	k^{32}	k^{sp2}	k^{17}	k^{18}	k^{19}	k_{20}	k^{21}	k^{22}	k^{23}	k^{24}
K_{61}	k^{11}	k^{12}	k^{13}	k^{14}	k^{15}	k^{16}	k^{sp1}	k^1	k^2	k^3	k^4	k^5	k^6	k^7	k^8	k^9
K_{62}	k^{28}	k^{29}	k^{30}	k^{31}	k^{32}	k^{sp2}	k^{17}	k^{18}	k^{19}	k^{20}	k^{21}	k^{22}	k^{23}	k^{24}	k^{25}	k^{26}

3 Impossible Differentials of SAFER$_+$ and SAFER$_{++}$

In this section, we will show how to identify the impossible differentials for SAFER+ and SAFER++. As a result, 2.75 rounds impossible differentials for SAFER+ and 3.75 rounds impossible differentials for SAFER++ are presented.

3.1 Notations

In this paper we use the following notations: T_r^I denotes the input of the r-th round, T_r^U, T_r^S, T_r^L and T_r^A denote the output values of the upper key layer, the S-boxes, the lower key layer and the linear layer in round r, respectively. So $T_r^I = T_{r-1}^A$ for $r \geq 2$. Δ represents the modular subtraction difference in $\mathbb{F}_{2^8}$. $*$ means the undetermined value. $(\Delta T_r^i)_j$ stands for the j-th byte of ΔT_r^i, $0 \leq j \leq 15$. C_j means the j-th byte of the ciphertext, $0 \leq j \leq 15$.

3.2 Impossible Differentials of SAFER$_+$ and SAFER$_{++}$

Firstly, we will introduce three propositions related with S-boxes, XOR and the modular addition.

Proposition 1 (see [7]) *For any byte pair (p, p'), if $(p - p') \equiv 0x80$ (mod 256), then the output difference $X(p) \boxminus X(p')$ is always odd.*

Proposition 2 (see [6]) *For any byte pair* (p, p'), $p \oplus p' = 0x80$ *always means* $(p - p') \equiv 0x80 \ (mod\ 256)$, *and vice versa.*

Proposition 3 (see [16]) *For any given byte pair* (p, p'), *if* $p \oplus p'$ *is odd, then* $(p \boxplus k) \oplus (p' \boxplus k)$ *is odd. Also, if* $p \boxplus p'$ *is odd,* $(p \oplus k) \boxminus (p' \oplus k)$ *is odd. Here,* k *can take any value in* $\mathbb{Z}_{256}$.

Based on the propositions, we can get 2.75-round impossible differentials for SAFER+ and 3.75-round impossible differentials for SAFER++.

Theorem 1 *For SAFER+, if the output difference of the S-boxes in the first round* ΔT_1^S *is* $(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 80_x, 0)$ *and the output difference of the upper key layer in the fourth round* ΔT_4^U *is (0, a, 0, 0, 0, b, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), where a and b are any non-zero value. Such 2.75-round differential is impossible if a and b satisfy one of the three following conditions:* $a + b = 0$, $8a + b = 0$, $a + 8b = 0$.

Proof. We list the 2.75 rounds impossible differentials with $a+b = 0$ in Fig.1. In the forward direction: as $\Delta T_1^S = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 80_x, 0)$, we have $(\Delta T_1^L)_{14} = (\Delta T_1^S)_{14}$ according to Proposition 2. Then we have

$$\Delta T_2^I = \Delta T_1^A = \Delta T_1^L \times M_+ = (0, 0, 80_x, 80_x, 0, 0, 80_x, 80_x, 0, 0, 0, 80_x, 0, 0, 0, 0).$$

According to Proposition 2 again, after the upper key layer, ΔT_2^U=ΔT_2^I. From Proposition 1 we have: ΔT_2^S=(0, 0, m_1, od, 0, 0, n_1, od, 0, 0, 0, od, 0, 0, 0, 0), where m_1 and n_1 are undetermined non-zero values, and od is the odd value. Proposition 3 tells us ΔT_2^L=(0, 0, m_2, od, 0, 0, n_2, od, 0, 0, 0, od, 0, 0, 0, 0), where m_2 and n_2 are undetermined non-zero values. Then $\Delta T_2^A = \Delta T_2^L \times M_+$=($*$, $*$, $*$, od, $*$, $*$, od, $*$, od, $*$, $*$, $*$, od, od, $*$, $*$), where $*$ is undetermined value.

In the reverse direction: If $a + b = 0$, from ΔT_4^U=(0, a, 0, 0, 0, $-a$, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), we have ΔT_3^A=ΔT_4^I=(0, a, 0, 0, 0, $-a$, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0). ΔT_3^L=$\Delta T_3^A \times M_+^{-1}$=($*, *, *, 0, *, *, *, *, 0, *, *, *, *, *, *, *$). After the S-box layer and the upper key layer in the third round, we have ΔT_2^A=ΔT_3^I=($*, *, *, 0, *, *, *, *, 0, *, *, *, *, *, *, *$).

So the 4-th and 9-th byte of the output difference of the second round is zero, which is contradiction with the fact that the 4-th and 9-th byte of the output difference of the second round is odd. Therefore, the 2.75-round differential is an impossible differential.

The proofs of the other two kinds of impossible differentials are similar. □

Theorem 2 *For SAFER++, if the output difference of the S-boxes in the first round* ΔT_1^S *is* $(0, 80_x, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 80_x, 0)$, *and the output difference of the upper key layer in the fifth round is*

$$\Delta T_5^U = (0, a, -a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$

where a is any non-zero value, such 3.75-round differential is impossible.

Proof. The 3.75 rounds impossible differentials have been shown in Fig.2. In the forward direction: as ΔT_1^S=(0, 80_x, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 80_x, 0), we have ΔT_1^L=ΔT_1^S according to Proposition 2. Then

$$\Delta T_1^A = \Delta T_1^L \times M_{++} = (0, 0, 0, 0, 80_x, 80_x, 0, 80_x, 0, 0, 0, 0, 80_x, 0, 80_x, 80_x) = \Delta T_2^I.$$

According to Proposition 2 again, after the upper key layer, $\Delta T_2^U = \Delta T_2^I$. From Proposition 1 we have: ΔT_2^S=(0, 0, 0, 0, od, m_3, 0, od, 0, 0, 0, 0, od, 0, n_3, od), where m_3 and n_3 are undetermined non-zero values, and od is the odd value. Proposition 3 tells us ΔT_2^L=(0, 0, 0, 0, od, m_4, 0, od, 0, 0, 0, 0, od, 0, n_4, od), where m_4 and n_4 are undetermined non-zero values. Then $\Delta T_2^A = \Delta T_2^L \times M_{++}$=($*$, $*$, $*$, $*$, od, $*$, $*$, $*$, $*$, $*$, $*$, $*$, $*$, $*$, $*$, $*$), where $*$ is undetermined.

In the reverse direction: ΔT_5^U=(0, a, $-a$, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), we have ΔT_4^A=ΔT_5^I=(0, a, $-a$, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ΔT_4^L=$\Delta T_4^A \times M_{++}^{-1}$=(0, $*$, $*$, 0, $*$, $*$, $*$, $*$, 0, 0, 0, 0, $*$, $*$, $*$, $*$). After the S-boxes layer and the upper key layer in the fourth round, we have $\Delta T_3^A = \Delta T_4^I = (0, *, *, 0, *, *, *, *, 0, 0, 0, 0, *, *, *, *)$. Then $\Delta T_3^L = \Delta T_3^A \times M_{++}^{-1} = (*, *, *, *, 0, *, *, *, *, *, *, *, *, *, *, *)$, $\Delta T_3^S = (*, *, *, *, 0, *, *, *, *, *, *, *, *, *, *, *) = \Delta T_3^I$. So the fifth byte of the input difference of round 3 is zero, which contradicts the fact that the fifth byte of the output difference of the second round is odd. Therefore, the 3.75-round differential is an impossible differential. □

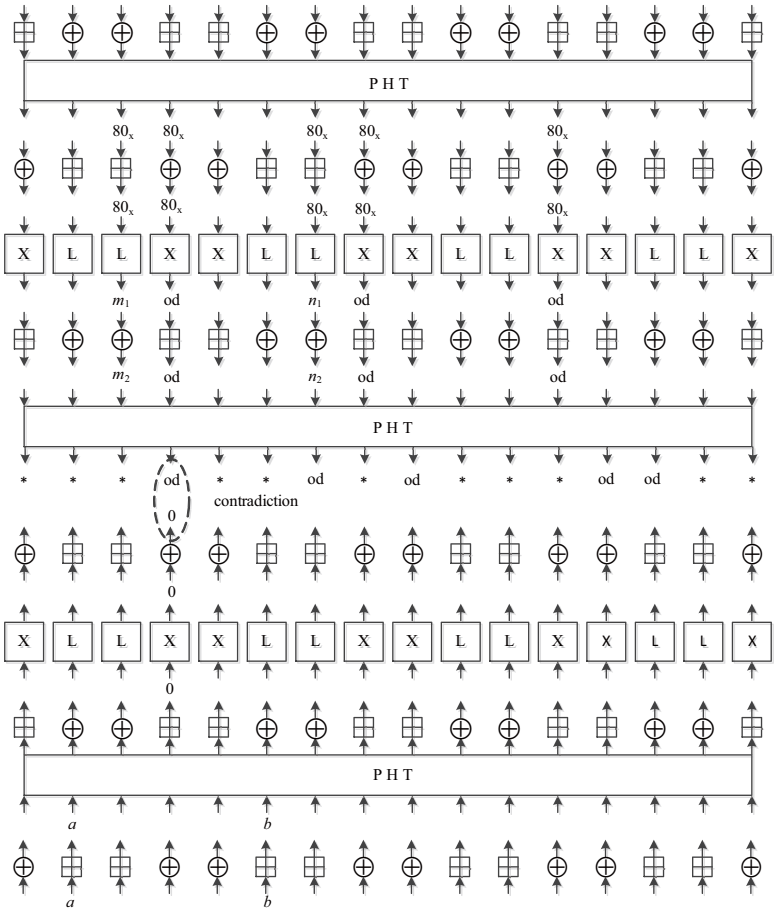

Fig. 1. 2.75-Round Impossible Differential of SAFER+

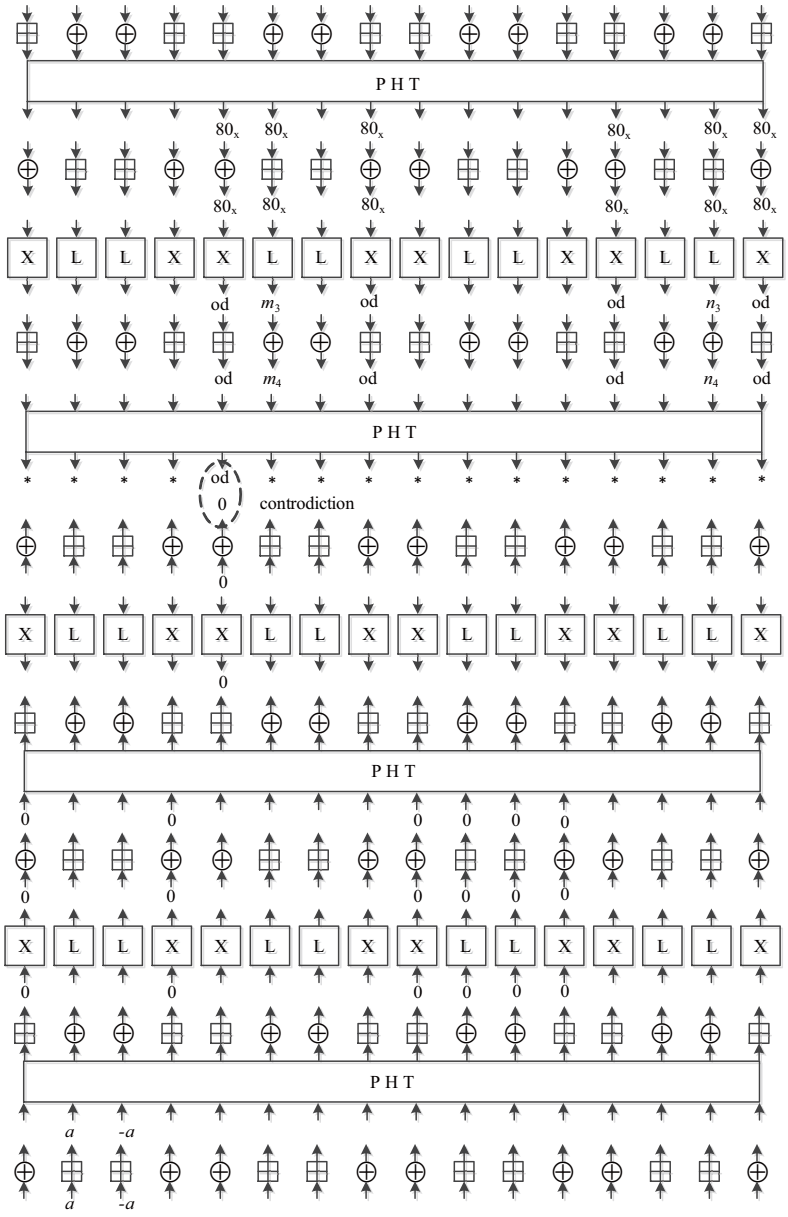

Fig. 2. 3.75-Round Impossible Differential of SAFER++

4 Impossible Differential Attacks on SAFER$_+$

In this section, we will use our 2.75-round impossible differential to recover the keys for four rounds of SAFER+/128 in Fig.3 and four rounds of SAFER+/256 in Fig.4. First of all, in order to filter out the pairs as soon as possible, we derive the relation between the ciphertext bytes difference in Proposition 4.

Proposition 4 *For four full-round of SAFER+/128 or SAFER+/256, if the pairs have the difference ΔT_5^U=(0, a, −a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), the differences for their corresponding ciphertext pairs have the following relations,*

$$\Delta C_1 - \Delta C_2 = 0, 2\Delta C_1 - \Delta C_6 = 0, \Delta C_5 - \Delta C_{10} = 0, \quad (1)$$

$$\Delta C_5 + \Delta C_9 - 5\Delta C_{13} = 0, \Delta C_6 + 2\Delta C_{14} - 6\Delta C_{13} = 0, \Delta C_1 + \Delta C_5 + \Delta C_6 - 7\Delta C_{13} = 0. \quad (2)$$

Proof. From Fig. 3 and Fig. 4, we know ΔT_4^U=(0, a, $-a$, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), then we get ΔT_4^L=ΔT_4^S=(0, A, B, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), where both A and B are non-zero undetermined difference values. By multiplying the matrix M_+, we can get $\Delta T_4^A = (A + 2B, A + 2B, A + 2B, A + B, 8A + 2B, 4A + B, 2A + 4B, A + 2B, 2A + 8B, A + 4B, 4A + B, 2A + B, A + B, A + B, 2A + B, 2A + B)$. So the ciphertext difference $\Delta C = (*, A+2B, A+2B, *, *, 4A+B, 2A+4B, *, *, A+4B, 4A+B, *, *, A+B, 2A+B, *)$. Therefore, we have

$$\Delta C_1 = \Delta C_2 = A + 2B, \quad 2\Delta C_1 = \Delta C_6 = 2A + 4B, \quad \Delta C_5 = \Delta C_{10} = 4A + B,$$
$$\Delta C_5 + \Delta C_9 = 4A + B + A + 4B = 5(A + B) = 5\Delta C_{13},$$
$$\Delta C_6 + 2\Delta C_{14} = 2A + 4B + 2(2A + B) = 6(A + B) = 6\Delta C_{13},$$
$$\Delta C_1 + \Delta C_5 + \Delta C_6 = A + 2B + 4A + B + 2A + 4B = 7(A + B) = 7\Delta C_{13}.$$

□

4.1 Impossible Differential Attack on SAFER$_+$/128

By placing the 2.75-round impossible differential on round 0.5-3.25, we can attack from round 1 to round 4. This is described in Fig.3. In order to show the effect of the key schedule, we denote our guessed subkey bits with their related master key bytes instead of themselves in Fig.3, and the similar denotation is used in the following figures for the attacks on SAFER+/256, SAFER++/128 and SAFER++/256.

Data Collection. We first construct $2^{114.4}$ structures of plaintexts, where in each structure the plaintext byte P_{14} takes all values, whereas the other bytes are fixed. For each structure, ask for the encryption of the plaintexts to get the corresponding ciphertexts. In order to filter out the wrong pairs with Equation (1) in Proposition 4, we construct a hash table indexed by $(C_1 - C_2|2C_1 - C_6|C_5 - C_{10})$ and put 2^8 corresponding ciphertexts into the hash table. Then we combine the ciphertext pairs in the same entity in the hash table which can satisfy Equation (1) in Proposition 4. On average there are about $2^{15}/2^{24} = 2^{-9}$ remaining pairs for each structure. Then we will further filter out the wrong pairs with Equation (2) in Proposition 4, so we construct another hash table indexed by $(C_5 + C_9 - 5C_{13}|C_6 + 2C_{14} - 6C_{13}|C_1 + C_5 + C_6 - 7C_{13})$ and put the 2^{-9} pairs into the hash table. There will remain pairs in the same entity in the hash table which can satisfy Equation (2) in Proposition 4. Now we can easily get the value of A and B in Fig.3 from the ciphertext difference for each remaining ciphertext pair. On average, there are $2^{-9}/2^{24} = 2^{-33}$ remaining pairs for each structure.

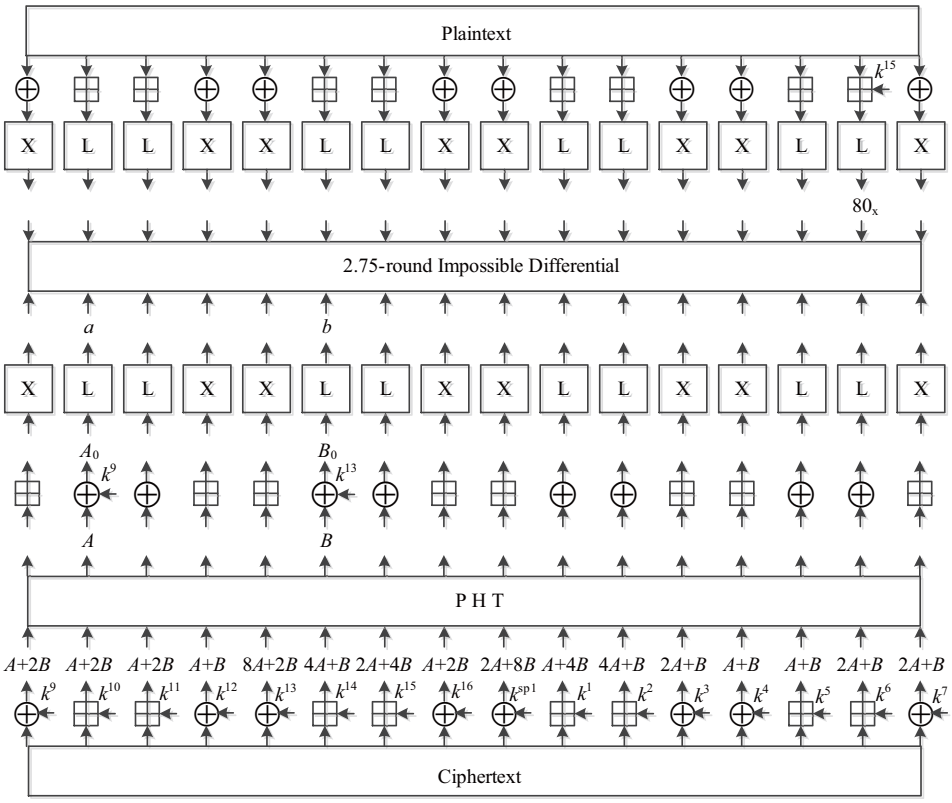

Fig. 3. Impossible Differential Attack on 4-Round SAFER+/128

Key Recovery. In order to find if there are pairs obtained from the data collection phase that may follow the differential in Fig.3, we need to guess the key bits and sieve the pairs in round 1 and round 4. From Fig.3, in round 1, we need to guess the 15-th subkey byte in the upper key layer which is related to the master key byte k^{15} from Table 2. In round 4, 16 subkey bytes of the lower key layer which are related to the master key bytes (k^9, k^{10}, k^{11}, ..., k^{16}, k^{sp1}, k^1, k^2,..., k^7) and we will guess partial bits for these 16 key bytes. We also need to guess the second and the sixth subkey bytes of the upper key layer which are related to the master key bytes (k^9, k^{13}). We proceed the key recovery phase for the remaining pairs as follows:

- Step 1. For all 2^8 possible values for the 15-th subkey byte of the upper key layer of the first round which depends on k^{15}, encrypt each plaintext of 2^{-33} remaining pairs for $\frac{1}{2}$ round to get the output differences of the S-boxes in the first round, which should satisfy $(\Delta T_1^S)_{14} = 80_x$. Then the number of remaining pairs is about 2^{-41}. The total number of guessed subkey bits in this step is 8.
- Step 2.
  - Step 2.1 In the final whitening key layer, there are eight XOR operations. As we get the ciphertext differences for the eight bytes, we can directly get the value for the least significant bit of $(\Delta T_4^A)_j, j \in \{0, 3, 4, 7, 8, 11, 12, 15\}$ without guessing the corresponding subkey value. Because we have known the value for A and B in the data collection phase, we can derive the difference values for the eight least significant bits from A and B. Then we can sieve the pairs with the eight conditions, as a result, $2^{-41}/2^8 = 2^{-49}$ pairs remain for each structure.
  - Step 2.2 For all 2^8 values of the least significant bits of the eight subkey bytes which depend on k^9, k^{12}, k^{13}, k^{16}, k^{sp1}, k^3, k^4, and k^7, respectively, compute the second least significant bits for $(\Delta T_4^A)_j, j \in \{0, 3, 4, 7, 8, 11, 12, 15\}$ for all remaining pairs and verify if they equal to the corresponding values obtained from A and B. If not, we discard the pair. In a similar way, we guess the eight subkey bytes from the second least significant bit to the seventh least significant bit one by one and sieve the pairs according the conditions derived from A and B. As a result, about $2^{-49}/2^{8*7} = 2^{-105}$ pairs are obtained. The total number of new guessed subkey bits in this step is 56.
- Step 3. In this step, we will compute the value for a and b corresponding to $(\Delta T_4^U)_1$ and $(\Delta T_4^U)_5$. With M_+^{-1},

$$\begin{aligned}(T_4^U)_1 &= S^{-1}((T_4^L)_1 - (K_{42})_1)\\ &= S^{-1}([-2, 4, -2, 4, -1, 1, -4, 4, -1, 1, -2, 2, -8, 16, -1, 2] \times (T_4^A) - k^9),\\ (T_4^U)_5 &= S^{-1}((T_4^L)_5 - (K_{42})_5))\\ &= S^{-1}([-1, 2, -4, 4, -1, 1, -2, 2, -1, 1, -8, 16, -2, 4, -2, 4] \times (T_4^A) - k^{13}),\end{aligned}$$

  so in order to calculate the values of $(T_4^U)_1$ and $(T_4^U)_5$, the bits depending on the following keys should be guessed:

  $(T_4^U)_1$: k^{13}, k^{14}, k^{sp1}, k^1, k^6, k^9; the seven least significant bits of k^{11}, k^2, k^3, k^7; the six least significant bits of k^{10}, k^{12}, k^{15}, k^{16}; the five least significant bits of k^4; the four least significant bits of k^5.

  $(T_4^U)_5$: k^9, k^{13}, k^{14}, k^{sp1}, k^1; the seven least significant bits of k^{10}, k^{15}, k^{16}, k^4, k^6; the six least significant bits of k^{11}, k^{12}, k^5, k^7; the five least significant bits of k^2; the four least significant bits of k^3.
  Here some subkey bits have been guessed in the previous steps, so the total number of the new involved subkey bits in this step is 54.

For each pair obtained from Step 2.2, compute a and b to verify if they satisfy any one of the three relations for the three impossible differentials. If so, the 54-bit subkey should be discarded. After processing all the pairs, if any values for the 54-bit subkey remain, we output them with the guessed 64-bit subkey, and exhaustively search them with the remaining 10 bits subkey by trial encryption. Otherwise, we try another guess for 64-bit subkey from Step 1 and Step 2.

The data complexity of the attack is $2^{122.4}$ chosen plaintexts. In the data collection phase, the time complexity is about $2^{122.4} \times 3 = 2^{124}$ modular subtraction operations which is equivalent to 2^{119} encryptions and the memory complexity is about $2^{81.4} \times 2 \times 32 = 2^{87.4}$ bytes for the remaining pairs. In Step 1, the time complexity is about $2 \times 2^8 \times 2^{114.4-33} \times \frac{1}{2} \times \frac{1}{16} \times \frac{1}{4} = 2^{83.4}$ encryptions and the memory complexity for remaining pairs is less than that in the data collection phase. In Step 2, the time complexity is about $2 \times 2^8 \times 2^{114.4-41} \times 8 \times 8 \approx 2^{88.4}$ XOR operations and $2^{87.4}$ modular subtraction operations. The memory complexity for remaining pairs is less than that in Step 1. In Step 3, The expected number of remaining 118-bit subkey guesses is about $2^{118} \times (1 - \frac{3}{2^8})^{2^{114.4-105}} \approx 2^{108}$. Since each of the remaining key guesses has to be exhaustively searched with the other 2^{10} key values, so the time complexity of this step is about $2 \times 2^{118} \times [1 + (1 - \frac{3}{2^8}) + (1 - \frac{3}{2^8})^2 + \ldots + (1 - \frac{3}{2^8})]^{2^{9.4}} \times \frac{2}{16} \times \frac{1}{4} + 2^{108+10} \approx 2^{120.7}$ encryptions. Thus the total time complexity is about 2^{121} encryptions and the memory complexity is about $2^{87.4}$ bytes.

4.2 Impossible Differential Attack on SAFER$_+$/256

The attack on 4 rounds of SAFER+/256 is shown in Fig.4. The only difference between the attack on SAFER+/256 and SAFER+/128 is the difference in the key schedule.

Data Collection. Because this phase is not related to the key, it is completely the same as the data collection in the attack on SAFER+/128 except that the number of structures is $2^{116.4}$. We do not describe it here.

Key Recovery.

- Step 1. From Fig.4, the key byte to be guessed in the first round is k^{15}, which is the same as that in the attack on SAFER+/128, so this step is the same as Step 1 in the attack on SAFER+/128.
- Step 2. In this step, the guessed new key bits are completely the same as those of Step 2 in the attack on SAFER+/128.
- Step 3. In this step, we will guess the necessary subkey bits compute the values for a and b. The total number of the new involved subkey bits in this step is 70.
  For each pair obtained from Step 2, compute a and b to verify if they satisfy any one of the three relations for the three impossible differentials. If so, the 70-bit subkey should be discarded. After processing all the pairs, if any values for the 70-bit subkey remain, we output them with the guessed 64-bit subkey, and exhaustively search them with the remaining 122 bits subkey by trial encryption. Otherwise, we try another guess for 64-bit subkey from Step 1 and Step 2.

The data complexity of the attack is $2^{124.4}$ chosen plaintexts. In data collection phase, Step 1 and Step 2, the time complexity is four times bigger than that in the attack on SAFER+/128. In Step 3, the expected number of remaining 134-bit subkey guesses is about $2^{134} \times (1 - \frac{3}{2^8})^{2^{116.4-105}} \approx 2^{94}$. Since each of the remaining key guesses has to be exhaustively searched with the other 2^{122} key values, so the time complexity of this step is about $2 \times 2^{134} \times [1 + (1 - \frac{3}{2^8}) + (1 - \frac{3}{2^8})^2 + \ldots + (1 - \frac{3}{2^8})]^{2^{11.4}} \times \frac{2}{16} \times \frac{1}{4} + 2^{94+122} \approx 2^{216}$ encryptions. So the total time complexity is 2^{216} encryptions and the memory complexity is about $2^{89.4}$ bytes.

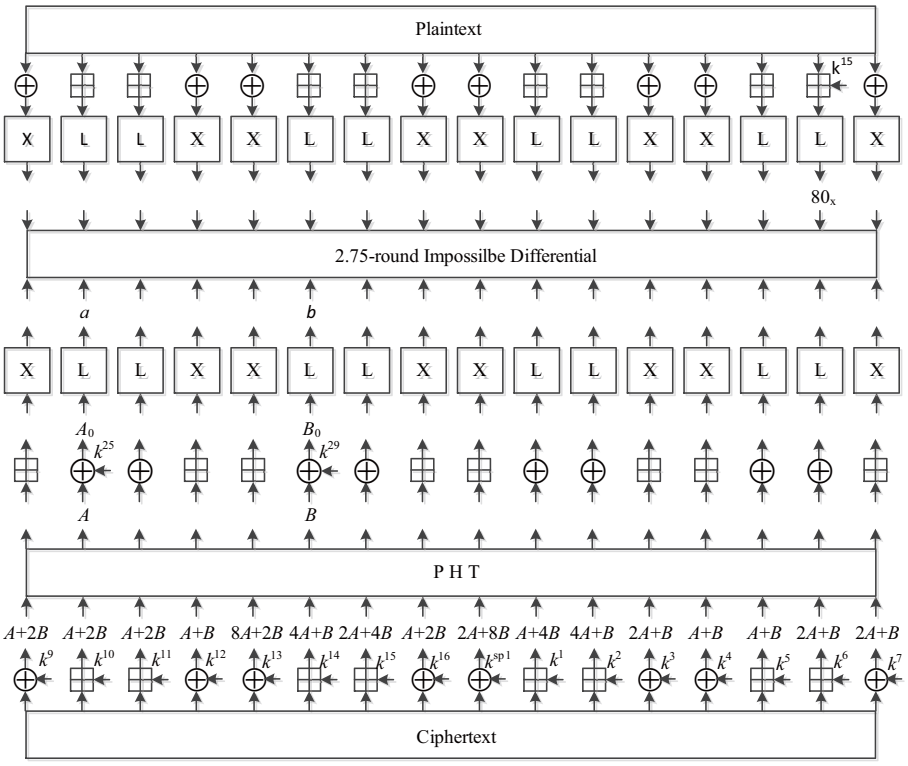

Fig. 4. Impossible Differential Attack on 4-Round SAFER+/256

5 Impossible Differential Attacks on SAFER$_{++}$

In this section, we will use the 3.75-round impossible differentials for SAFER$_{++}$ in Section 3 to recover the keys for SAFER$_{++}$/128 in Fig.5 and SAFER$_{++}$/256 in Fig.6. First of all, in order to filter out the pairs as soon as possible, we derive the relations between the ciphertext bytes difference in Proposition 5.

Proposition 5 *For five full rounds SAFER++/128 or SAFER++/256, if the pairs have the difference* ΔT_5^U*=(0, a, b, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), their corresponding output difference has the following relations,*

$$(\Delta T_6^U)_5 - (\Delta T_6^U)_{10} = 0, (\Delta T_6^U)_1 - (\Delta T_6^U)_9 = 0, (\Delta T_6^U)_6 - (\Delta T_6^U)_{14} = 0, \tag{3}$$

$$\begin{array}{c} (\Delta T_6^U)_1 + (\Delta T_6^U)_6 - 3(\Delta T_6^U)_5 = 0, \\ (\Delta T_6^U)_2 + (\Delta T_6^U)_{13} - 5(\Delta T_6^U)_5 = 0, \\ 3(\Delta T_6^U)_1 + (\Delta T_6^U)_{13} - 7(\Delta T_6^U)_5 = 0. \end{array} \tag{4}$$

The proof can be finished from Fig.5 and Fig.6 with a similar proving method for Proposition 4, so we will not describe it here.

5.1 Impossible Differential Attack on SAFER$_{++}$/128

By placing the 3.75-round impossible differential on round 0.5-4.25, we can attack SAFER$_{++}$/128 from round 1 to round 5. This is described in Fig.5.

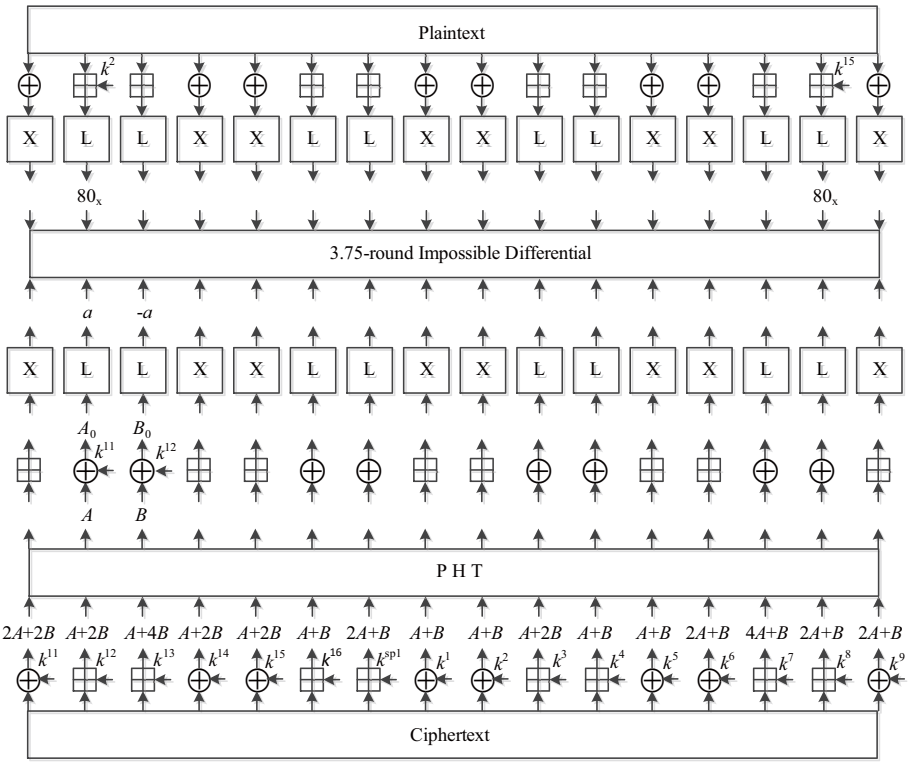

Fig. 5. Impossible Differential Attack on 5-Round SAFER++/128

Data Collection. We first construct 2^{108} structures of plaintexts, where in each structure the plaintext bytes P_1 and P_{14} take all values, whereas the other bytes are fixed. For each structure, ask for the encryption of the plaintexts to get the corresponding ciphertexts. In order to filter out the wrong pairs with Equation (3) in Proposition 5, we construct a hash table indexed by $(C_5 - C_{10}|C_1 - C_9|C_6 - C_{14})$ and put 2^{16} corresponding ciphertexts into the hash table. Then we combine the ciphertext pairs in the same entity in the hash table which can satisfy Equation (3) in Proposition 5. On average there are about $2^{31}/2^{24} = 2^7$ remaining pairs for each structure. Then we will further filter out the wrong pairs with Equation (4) in Proposition 5, so we construct another hash table indexed by $(C_1 + C_6 - 3C_5|C_2 + C_{13} - 5C_5|3C_1 + C_{13} - 7C_5)$ and put the 2^7 pairs into the hash table. There will remain pairs in the same entity in the hash table which can satisfy Equation (4) in Proposition 5. Now we can easily get the value of A and B in Fig.5 from the ciphertext difference for any remaining ciphertext pair. On average, there are $2^7/2^{24} = 2^{-17}$ remaining pairs for each structure.

Key Recovery. In order to find if there are pairs obtained from the data collection phase that may follow the differential in Fig.5, we need to guess the key bits and sieve the pairs in round 1 and round 5. From Fig.5, in round 1, we need to guess the second and 15-th subkey bytes in the upper key layer which are related to the master key bytes k^2 and k^{15}, respectively. In round 5, 16 final whitening subkey bytes are related to the master key bytes $(k^{11}, k^{12}, k^{13}, \ldots, k^{16}, k^{sp1}, k^1, k^2, \ldots, k^9)$ and we will guess partial bits for the 16 subkey bytes. We also need to guess the second and the third subkey bytes of the lower key layer in round 5 which are related to the master key bytes (k^{11}, k^{12}), respectively. We proceed the key recovery phase for the remaining pairs as follows:

– Step 1. For all 2^{16} possible values for the second and the 15-th bytes of upper key layer of the first round which depend on k^2 and k^{15}, for each structure encrypt each plaintext pair of the 2^{-17} remaining pairs for $\frac{1}{2}$ round to get the output differences of the S-boxes in the first round, which should satisfy $(\Delta T_1^S)_1 = (\Delta T_1^S)_{14} = 80_T$. Then the number of remaining pairs is about 2^{-33}. The total number of guessed subkey bits in this step is 16.
– Step 2.
  - Step 2.1 In the final whitening key layer, there are eight XOR operations. As we get the ciphertext differences for the eight bytes, we can directly get the value for the least significant bit of $(\Delta T_5^A)_j, j \in \{0, 3, 4, 7, 8, 11, 12, 15\}$ without guessing the corresponding subkey value. Because we have known the value for A and B in the data collection phase, we can derive the 8 bits difference for the least significant bits from A and B. Then we can sieve the pairs with the eight conditions, as a result, $2^{-33}/2^8 = 2^{-41}$ pairs remain for each structure.
  - Step 2.2 For all 2^8 values for the least significant key bit of eight subkey bytes which depend on k^{11}, k^{14}, k^{15}, k^1, k^2, k^5, k^6, k^9, respectively, compute the second least significant bits for $(\Delta T_5^A)_j, j \in \{0, 3, 4, 7, 8, 11, 12, 15\}$ for all remaining pairs and verify if they equal to the corresponding value derived from A and B. If not, we discard the pair. In a similar way, we guess eight subkey bytes from the second least significant bits to the seventh least significant bits one by one which depend on k^{11}, k^{14}, k^{15}, k^1, k^2, k^5, k^6, k^9, respectively, then we sieve the pairs according to the conditions derived from A and B. As a result, about $2^{-41}/2^{8*7} = 2^{-97}$ pairs are obtained. The total number of new guessed subkey bits in this step is 42.
– Step 3. In this step, we will compute the value for a and $-a$ corresponding to $(\Delta T_5^U)_1$ and $(\Delta T_5^U)_2$. Similar to the attack on SAFER+, we only guess the subkey bits that are necessary, the total number of the new involved subkey bits in this step is 52.
  For each pair obtained from Step 2.2, compute the value for $(\Delta T_5^U)_1$ and $(\Delta T_5^U)_2$ to verify if $(\Delta T_5^U)_1 = -(\Delta T_5^U)_2$. If so, the 52-bit subkey should be discarded. After processing all the pairs, if any values for the 52-bit subkey remain, we output them with the guessed 58-bit subkey, and exhaustively search them with the remaining 18 bits key by trial encryption. Otherwise, we try another guess for 58-bit subkey from Step 1 and Step 2.

The data complexity of the attack is 2^{124} chosen plaintexts. In the data collection phase, the time complexity is about $2^{124} \times 3 = 2^{125.6}$ modular subtraction operations, which is equivalent to $2^{120.6}$ times of encryptions and the memory complexity is about $2^{91} \times 2 \times 32 = 2^{97}$ bytes for the remaining pairs. In Step 1, the time complexity is about $2 \times 2^{16} \times 2^{91} \times \frac{1}{2} \times \frac{1}{16} \times 14 = 2^{101}$ encryptions. In Step 2, the time complexity is about $2 \times 2^{16} \times 2^{75} \times 8 \times 8 \approx 2^{98}$ XOR operations and 2^{97} modular subtraction operations. In Step 3, the expected number of remaining 110-bit key guesses is about $2^{110} \times (1 - \frac{1}{2^8})^{2^{108-97}} \approx 2^{100}$. Sine each of the remaining key guesses has to be exhaustively searched with the other 2^{18} key values, so the time complexity of this step is about $2 \times 2^{110} \times [1 + (1 - \frac{1}{2^8}) + (1 - \frac{1}{2^8})^2 + \ldots + (1 - \frac{1}{2^8})]^{2^{11}} \times \frac{2}{16} \times \frac{1}{4} + 2^{118} \approx 2^{118}$ encryptions. Thus the total time complexity is about 2^{121} encryptions and the memory complexity is about 2^{97} bytes.

5.2 Impossible Differential Attack on SAFER++/256

The attack on 5.5 rounds of SAFER++/256 is shown in Fig.6. We put the 3.75-round impossible differential from round 0.5 to round 4.25 and we will recover the key for 5.5-round SAFER++/256.

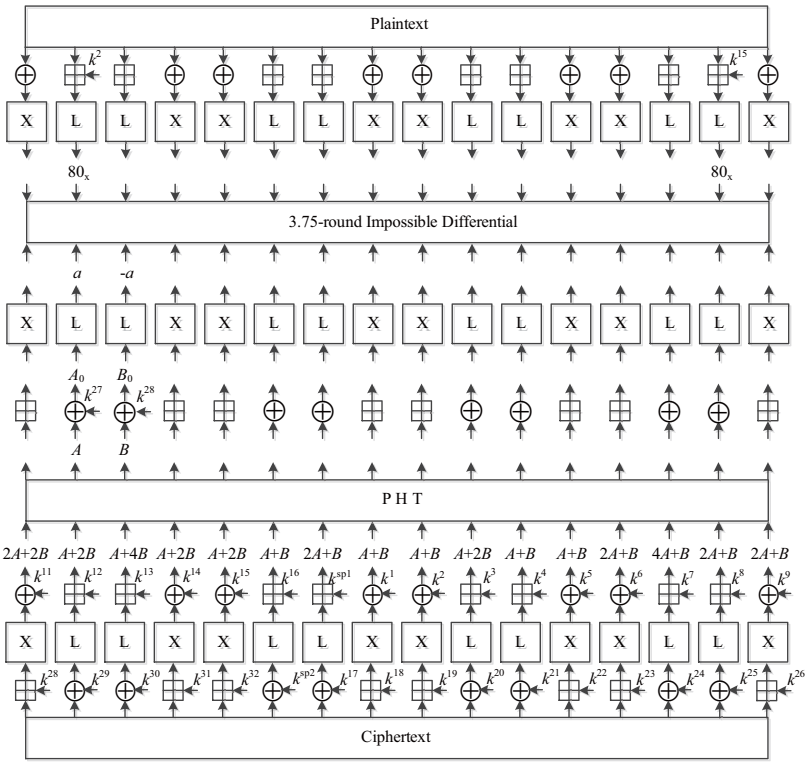

Fig. 6. Impossible Differential Attack on 5.5-Round SAFER++/256

Attack Procedure. We first construct 2^{108} structures of plaintexts, where in each structure the plaintext bytes P_1 and P_{14} take all values, whereas the other bytes are fixed. For each structure, ask for the encryption of the plaintexts to get the corresponding ciphertexts. In order to find if there are pairs obtained that may follow the differential in Fig.6, we need to guess the key bits and sieve the pairs in round 1, round 5 and round 6. We proceed the key recovery phase as follows,

- Step 1. Guess eight subkey bytes of the lower key layer in round 6 which depend on $k^j, j \in \{29, 30, sp2, 17, 20, 21, 24, 25\}$ and compute $(\Delta T_6^U)_l, l \in \{1, 2, 5, 6, 9, 10, 13, 14\}$ for the 2^{16} ciphertexts of each structure. We sieve the pairs satisfying the 48 conditions from Proposition 5. As a result, we obtain $2^{31-48} = 2^{-17}$ pairs for each structure. For the remaining pairs, we can get the values A and B from the ciphertext pairs.
- Step 2. Guess two subkey bytes in the first round which depend on k^2 and k^{15}, respectively and compute $(\Delta T_1^L)_1$ and $(\Delta T_1^L)_{14}$ from the plaintexts of the remaining pairs. Finally, we get 2^{-33} pairs for each structure.
- Step 3.
  - Step 3.1 As we have guessed k^{15} in Step 2, we only guess k^{32} and we can compute $(\Delta T_5^A)_4$. We sieve the pairs satisfying $(\Delta T_5^A)_4 = A + 2B$. As a result, the number of remaining pairs is $2^{-33-8} = 2^{-41}$.
  - Step 3.2 As we have guessed k^2 in Step 2, we only guess k^{19} and we can compute $(\Delta T_5^A)_8$. We sieve the pairs satisfying $(\Delta T_5^A)_8 = A + B$. As a result, the number of remaining pairs is $2^{-41-8} = 2^{-49}$.
  - Step 3.3 We guess 8-bit for k^{28} denoted as $k^{28}_{7\sim0}$ and the seven least significant subkey bits which depend on k^{11} denoted as $(k^{11})_{6\sim0}$ (we describe them as 15-bit tuples $(k^{28}_{7\sim0}, (k^{11})_{6\sim0})$)to compute $(\Delta T_5^A)_0$ for the remaining 2^{-49} ciphertext pairs for each

structure. We sieve the pairs satisfying $(\Delta T_5^A)_0 = 2A + 2B$. As a result, the number of remaining pairs is $2^{-49-8} = 2^{-57}$. In a similar way, we guess other 15-bit tuples $(k_{7\sim0}^{31}|(k^{14})_{6\sim0}), (k_{7\sim0}^{18}|(k^{1})_{6\sim0}), (k_{7\sim0}^{22}|(k^{5})_{6\sim0}), (k_{7\sim0}^{23}|(k^{6})_{6\sim0}), (k_{7\sim0}^{26}|(k^{9})_{6\sim0})$ one by one to sieve the pairs according to the corresponding values for ΔT_5^A. At last, we obtain $2^{-57-40} = 2^{-97}$ pairs for each structure. The number of the new guessed subkey bits is 90.

- Step 4. In this step, we will compute the value for a and $-a$ corresponding to $(\Delta T_5^U)_1$ and $(\Delta T_5^U)_2$. The total number of the new involved subkey bits in this step is 51.
  For each pair obtained from Step 3.3, compute the value for a and $-a$ corresponding to $(\Delta T_5^U)_1$ and $(\Delta T_5^U)_2$ to verify if $(\Delta T_5^U)_1 = -(\Delta T_5^U)_2$. If so, the 51-bit key should be discarded. After processing all the pairs, if any values for the 51-bit subkey remain, we output them with the guessed 186-bit subkey, and exhaustively search them with the remaining 19 bits key by trial encryption. Otherwise, we try another guess for 186-bit subkey from Step 1.

The data complexity of the attack is 2^{124} chosen plaintexts. The memory complexity is about 2^{97} bytes. The time complexity is dominated by Step 4. In this step, the expected number of remaining 237-bit key guesses is about $2^{237} \times (1 - \frac{1}{2^8})^{2^{108-97}} \approx 2^{227}$. Since each of the remaining key guesses has to be exhaustively searched with the other 2^{19} key values, the time complexity of this step is about $2 \times 2^{237} \times [1 + (1 - \frac{1}{2^8}) + (1 - \frac{1}{2^8})^2 + \ldots + (1 - \frac{1}{2^8})^{2^{11}}] \times \frac{2}{16} \times \frac{1}{4} + 2^{227+19} \approx 2^{246}$ encryptions.

6 Conclusion

This paper introduces impossible differential attacks on SAFER+ and SAFER++ block ciphers. We first derive 2.75-round and 3.75-round impossible differentials for SAFER+ and SAFER++, which improves the previous 2.75-round impossible differentials for SAFER++. With the impossible differentials, attacks on 4-round SAFER+/128(256), 5-round SAFER++/128 and 5.5-round SAFER++/256 can be achieved. Our method can also be applied to other ciphers that have similar structures to SAFER+.

7 Acknowledgment

We would like to thank anonymous reviewers for their very important comments. This work was supported by NSFC Projects(No.61133013 and No.61070244), by 973 Project (No.2013CB834205) as well as Interdisciplinary Research Foundation of Shandong University(No.2012JC018)

References

1. B. Behnam, E. Taraneh, and R. A. Mohammad. Impossible Differential Cryptanalysis of SAFER++. Proceedings of the 2008 International Conference on Security Management, SAM 2008, pp. 10–14. CSREA Press, 2008.
2. E. Biham, A. Biryukov and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds. EUROCRYPT 1999, LNCS 1592, pp. 12–23, Springer-Verlag, 1999.
3. A. Biryukov, C. De Canniere, G. Dellkrantz. Cryptanalysis of SAFER++. CRYTO 2003, LNCS 2729. pp. 195–211, Springer-Verlag, 2003.
4. BLUETOOTH SPECIFICATION Version 1.0B, 29 Nov. 1999, `http://www.bluetooth.com/link/spec/bluetooth_b.pdf`.

5. L. Knudsen. DEAL-A 128-bit Block Cipher. NIST AES proposal. Technial report 151, February 21, 1998 [retrieved 27.02.07].
6. L. Knudsen. A Detailed Analysis of SAFER K. Journal of Cryptplogy 2000, 13(4):417-436, 2000.
7. J. L. Massey. SAFER K-64: One Year Later. FSE 1995, LNCS 1008, pp. 212–241, Springer-Verlag, 1995.
8. J. L. Massey, G. H. Khachatrian, and M. K. Kuregian. 1st AES Conference on Nomination of SAFER+ as Candidate Algorithm for The Advanced Encryption Standard. California, USA, June 1998, `http://csrc.nist.gov/encryption/aes/`.
9. J. L. Massey, G. H. Khachatrian, and M. K. Kuregian. 1st NESSIE Workshop on The SAFER++ Block Encryption Algorithm. Heverlee, Belgium, November 2000, `http://cryptonessie.org`.
10. J. Nakahara, B. Preneel, and J. Vandewalle. Linear Cryptanalysis of Reduced-Round Versions of the SAFER Block Cipher Family. FSE 2000, LNCS 1978, pp. 244–261, Springer-Verlag, 1998.
11. J. Nakahara. Cryptanalysis and Design of Block Ciphers. PhD thesis. Katholidke University, Leuven, 2003.
12. J. Nakahara, and B. Preneel. Impossible Differential Attacks on Reduced-Round SAFER Ciphers. NESSIE Public Report, NES/DOC/KUL/WP5/30/1, 2003.
13. NESSIE Project–New European Schemes for Signatures, Integrity and Encryption. `http://cryptonessie.org`.
14. G. Piret, and J. Quisauater. Integral Cryptanalysis on Reduced-Round SAFER++–A Way to Extend The Attack? `http://eprint.iacr.org/2003/033.pdf`, 2003.
15. Y. Yemo, and I. Park. Optimization of Integral Cryptanalysis on Reduced-Round SAFER++. Joho Shori Gakkai Shinpojiumu Ronbunshu(published in Japan). 2003(15), 2003.
16. S. H. Zheng, C. L. Wang, and Y. X. Yang. A New Impossible Differential Attack on SAFER Ciphers. Computers and Electrical Engineering 36(2010): 180–189, 2010.

A Appendix

This appendix contains the matrices corresponding to the linear layer and its inverse for SAFER+ and SAFER++.

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

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

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

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