SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2018, Vol. 61 032106:1–032106:13
doi: 10.1007/s11432-016-9075-6

Impossible differential attack on Simpira v2

Rui ZONG[1], Xiaoyang DONG[1,2*] & Xiaoyun WANG[1,2]

[1]*Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan* 250100*, China;*
[2]*Institute for Advanced Study, Tsinghua University, Beijing* 100084*, China*

Received 17 December 2016/Accepted 16 March 2017/Published online 30 August 2017

Abstract Simpira v2 is a family of cryptographic permutations proposed at ASIACRYPT 2016, and can be used to construct high throughput block ciphers by using the Even-Mansour construction, permutation-based hashing, and wide-block authenticated encryption. This paper shows a 9-round impossible differential of Simpira-4. To the best of our knowledge, this is the first 9-round impossible differential. To determine some efficient key recovery attacks on its block cipher mode (Even-Mansour construction with Simpira-4), we use some 6/7-round shrunken impossible differentials. Based on eight 6-round impossible differentials, we propose a series of 7-round key recovery attacks on the block cipher mode; each 6-round impossible differential helps recover 32 bits of the master key (512 bits), and in total, half of the master key bits are recovered. The attacks require 2^{57} chosen plaintexts and 2^{57} 7-round encryptions. Furthermore, based on ten 7-round impossible differentials, we add one round on the top or at the bottom to mount ten 8-round key recovery attacks on the block cipher mode. This helps recover the full key space (512 bits) with a data complexity of 2^{170} chosen plaintexts and time complexity of 2^{170} 8-round encryptions. Those are the first attacks on the round-reduced Simpira v2 and do not threaten the Even-Mansour mode with the full 15-round Simpira-4.

Keywords Simpira-4, impossible differential attack, super S-box, the Even-Mansour construction, security claim

Citation Zong R, Dong X Y, Wang X Y. Impossible differential attack on Simpira v2. Sci China Inf Sci, 2018, 61(3): 032106, doi: 10.1007/s11432-016-9075-6

1 Introduction

Since the selection of the block cipher Rijndael [1] designed by Daemen and Rijmen as the advanced encryption standard (AES) in 2001 by NIST, it has been researched worldwide through various cryptanalysis methods, for example, impossible differential attack [2–4], SQUARE attack [5], collision attack [6], and meet-in-the-middle attack [7–9]. Although the full versions of AES-192 and AES-256 have been theoretically broken under the related-key model [10, 11], the attacks do not threaten the practical use of AES. Recently, some new 5-round distinguishers of AES were proposed [12, 13]; these extend the long-standing 4-round distinguisher by 1 round.

Nowadays, Intel, AMD, and ARM introduce AES instructions to their modern processors to reduce encryption overheads. As such, it becomes meaningful to design a permutation based on the AES round function as we can directly introduce the AES instruction during software implementation. As there are cipher suites that allow message blocks processed independently for encryption, the fixed block size of AES becomes a limitation.

* Corresponding author (email: dongxiaoyang@mail.sdu.edu.cn)

To achieve a higher throughput, Gueron and Mouha proposed Simpira in ASIACRYPT 2016 [14]. It is a family of cryptographic permutations that accepts arbitrarily large input sizes of $x \times 128$ bits, where $x \in \mathbb{N}^+$. Furthermore, to take advantage of the security of AES round function and the AES instructions set for well-optimized software implementations, Simpira uses two rounds of AES as the basic building block and uses a Feistel structure for $x \geqslant 2$ that operates on x input subblocks of 128 bits each.

One application of Simpira recommended by its designer is as a permutation in the Even-Mansour construction [15,16] for constructing a block cipher without round keys. The Even-Mansour construction has a trade-off security claim that when D plaintext-ciphertexts are available, the secret key K can be recovered in $2^n/D$ evaluations of the permutation [15]. In addition, the designer established a security claim about the permutation according to which Simpira can be used in constructions where an adversary cannot query a distinguisher more than 2^{128} times.

Two related studies that focus on Simpira v1 are as follows. In SAC 2016 [17], Dobraunig et al. showed that for Simpira v1, the underlying assumptions of independence and thus the derived bounds are incorrect. They provided differential trails with only 40 (instead of 75) active S-boxes for Simpira-v1 with $x = 4$. Based on these trails, they proposed full-round collision attacks on the proposed Davies-Meyer hash constructions based on Simpira v1 with $x = 4$. In addition, Rønjom reported on the invariant subspaces in Simpira v1 with $x = 4$ [18]. He showed that the whole coset of dimension 56 over $\mathbf{F}_{2^8}^{64}$ and the invariant subspaces result from the AES-based round function and the particular choice of Feistel configuration.

To solve these problems, Simpira v2 was designed by ensuring that every subblock will only be operated once. Simpira v2 has more complex round constants, and uses a more logical Feistel structure. In the following text, we use simply Simpira to denote Simpira v2.

In this study, we explored the security of Simpira against impossible differential cryptanalysis. The impossible differential cryptanalysis was independently proposed by Knudsen [19] and Biham [20]. Its main concept is to use impossible differentials that hold with probability zero to discard the wrong keys until only one key remains. Recently, inspired by Sun's work in [21,22], a new automatic search tool [23] was proposed for searching impossible differentials.

Our contribution. In the current study, we focus on the block cipher mode of Simpira v2 with four branches ($x = 4$), that is, the Even-Mansour construction with Simpira-4. We first present a 9-round impossible distinguisher, which is the first 9-round impossible differential on Simpira v2 with $x = 4$. In addition, we mount two impossible differential key recovery attacks: one on a 7-round Simpira with $x = 4$, a data complexity of 2^{57} plaintexts, and a time complexity of 2^{57} encryption units to recover 256 of 512 key bits with 6-round impossible differentials; and the other on an 8-round Simpira with $x = 4$, a data complexity of 2^{170} plaintexts, and a time complexity of 2^{170} encryption units to recover all 512 key bits with 7-round impossible differentials.

2 Preliminaries

2.1 Notations

$\bigoplus$: Bitwise XOR.
P: Plaintext.
C: Ciphertext.
S: Internal state.
F: Basic building block of Simpira.
ΔS: Difference between S and S'.
S_h: Input of the hth round, $h \geqslant 0$.
S^i: The ith subblock of S, $i \in \{0,1,2,3\}$.
$S^i[j]$: The jth byte of S^i, $j \in \{0, 1, 2, \ldots, 15\}$.
Simpira-x: Simpira with x subblocks, $x \in \mathbb{N}^+$.
0: Nibbles and subblocks with zero difference.

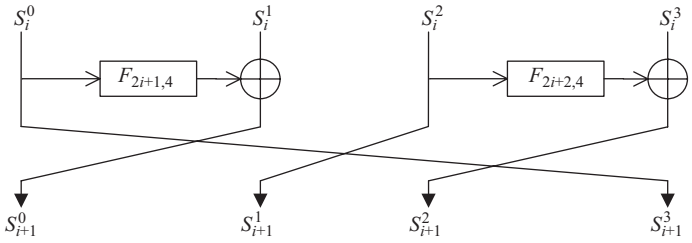

Figure 1 Round function of Simpira-4.

*: Nibbles and subblocks with nonzero difference.

f^{-1}: Inverse operation of function f.

a, b, α, β: To express the difference pattern of a subblock.

2.2 Description of Simpira

Simpira is a family of cryptographic permutations that supports $128 \times x$ bits, where x is a positive integer. Its design goal is to achieve high throughput on virtually all modern 64-bit processor architectures.

We only provide the details of Simpira-4, as all attacks were performed on it. For more information about Simpira, refer to [14]. Figure 1 presents the round function of Simpira-4; thus the state update rule is as follows, with $0 \leqslant i \leqslant 14$:

$$S_{i+1}^0 = S_i^1 \oplus F_{2i+1,4}(S_i^0), \quad S_{i+1}^1 = S_i^2,$$
$$S_{i+1}^2 = S_i^3 \oplus F_{2i+2,4}(S_i^2), \quad S_{i+1}^3 = S_i^0.$$

Note that when the number of rounds is not a multiple of 4, the state words are output in a permuted order to allow for more efficient implementations.

The Feistel update function is represented as $F = F_{c,x}$, where x is the number of subblocks, that is, 4, for Simpira-4 and c is a counter counted from 1. The function is made up of two rounds of AES, except the second AddRoundKey operation, which is the specific round constant updating process. Beyond that, SubBytes, ShiftRows, and MixColumns are identical to those in AES. For more details, refer to [1].

Every subblock can be expressed as a 4×4 matrix of bytes as follows:

$$S = (s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}) = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}.$$

Moreover, we refer to s_i as $S[i]$.

For convenience when referring to the internal states inside the F function for an input S, we use the same notations as in [17]:

$$S \xrightarrow{\text{SB}} S^{\text{SB}_1} \xrightarrow{\text{SR}} S^{\text{SR}_1} \xrightarrow{\text{MC}} S^{\text{MC}_1} \xrightarrow{\text{AC}} S^{\text{AC}} \xrightarrow{\text{SB}} S^{\text{SB}_2} \xrightarrow{\text{SR}} S^{\text{SR}_2} \xrightarrow{\text{MC}} S^{\text{MC}_2} = F(S).$$

2.3 The Even-Mansour construction

The (single-key) Even-Mansour construction [16] encrypts a plaintext P to a ciphertext under a secret key K as follows:

$$C = E_K(P) = \pi(P \oplus K) \oplus K,$$

where π is an n-bit permutation.

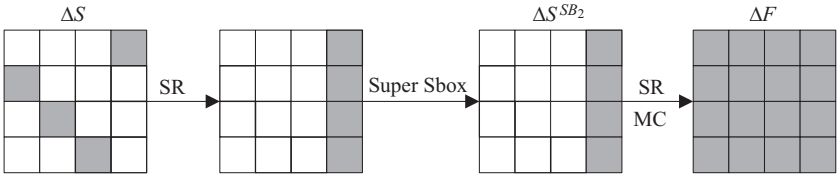

Figure 2 Super Sbox of AES.

2.4 Our attack assumptions

In this study, we focused on the impossible differential cryptanalysis of round-reduced Simpira-4. As recommended by the designer, Simpira-4 can be used as a permutation to construct block ciphers without round keys, for example, the Even-Mansour scheme with a 512-bit key.

In 2012 [15], Dunkelman and Shamir made a security claim that when D plaintext-ciphertexts are available, the secret key K of the Even-Mansour construction can be recovered in $2^n/D$ (offline) evaluations of the permutation π. If we use Simpira-4 as the permutation in the Even-Mansour scheme, then the product of the time complexity and the data complexity of an attack must be less than 2^{512} encryption units.

Furthermore, the designer made another security claim about Simpira [14]: Simpira can be used in constructions that require a random permutation; however, no statements were made for adversaries that exceed 2^{128} queries. As a result, both the data and time complexities of an attack should be less than 2^{128}.

Therefore, for different security claims, we mount two attacks of the Simpira-4-based Even-Mansour construction: one on a 7-round Simpira-4 with a data complexity of 2^{57} plaintexts and a time complexity of 2^{57} encryption units, and the other on an 8-round Simpira-4 with a data complexity of 2^{170} plaintexts and a time complexity of 2^{170} encryption units.

3 Impossible differential attacks on Simpira-4

In this section, we first present some useful observations and properties of Simpira-4, and then present the impossible differential distinguisher and attack procedure.

3.1 Some observations

In [24], Daemen and Rijmen introduced the structure of Super S-box to analyze the two-round differentials of AES. For clarity, we quote the definition of Super S-box as follows.

Definition (Super S-box)**.** The AES Super S-box maps a 4-byte array (s_0, s_1, s_2, s_3) to a 4-byte array (e_0, e_1, e_2, e_3) and takes a 4-byte key k. It consists of a sequence of four transformations: Subbytes, MixColumns, AddRoundKey, and SubBytes.

Property (Differential property of super S-box)**.** Given that Δinput and Δoutput are two nonzero differences in F_2^{32}, the equation of Super S-box can be written as follows:

$$\text{Super} - S(x) \oplus \text{Super} - S(x \oplus \Delta\text{input}) = \Delta\text{output},$$

and has one solution in average for each key value.

Observation 1. Consider the computational process of F-function: If there exists at least one inactive column of ΔS^{SR_1}, the number of all possible values of ΔF will not be less than 2^{128}.

Proof. Without loss of generality, we set the difference pattern of ΔS as follows:

$$\Delta S = (0, *, 0, 0, 0, 0, *, 0, 0, 0, 0, *, *, 0, 0, 0),$$

and swap the order of the first SubBytes and ShiftRows operations (Figure 2) to obtain an integrated super S-box structure.

Then, after the ShiftRows operation, the difference pattern will become

$$(0,0,0,0,0,0,0,0,0,0,0,0,*,*,*,*).$$

The difference pattern of the output of the Super S-box will become

$$\Delta S^{\mathrm{SB}_2} = (0,0,0,0,0,0,0,0,0,0,0,0,*,*,*,*).$$

Thus, although all 16 bytes of ΔF are active, the number of possible values is only 2^{32} instead of 2^{128}.

Observation 2 (The 9-round impossible differential; Figure 3)**.** If ΔS_0^1 is the only active subblock of the input difference ΔS_0 and ΔS_9^0 is the only active subblock of the output difference ΔS_9, the differential

$$(0,\Delta S_0^1,0,0) \xrightarrow{9R} (\Delta S_9^0,0,0,0)$$

is impossible when the difference patterns $\mathrm{SR}^{-1}\circ\mathrm{MC}^{-1}(\Delta F(S_0^1))$ and $\mathrm{SR}^{-1}\circ\mathrm{MC}^{-1}(\Delta F(S_9^0))$ are not the same.

For example, when the difference pattern of ΔS_0^1 is

$$(*,0,*,*,*,*,0,*,*,*,*,0,0,*,*,*),$$

the difference pattern of $\mathrm{SR}^{-1}\circ\mathrm{MC}^{-1}(\Delta F(S_0^1))$ is

$$(*,*,*,*,*,*,*,*,*,*,*,*,0,0,0,0).$$

Further, when the difference pattern of ΔS_9^0 is

$$(0,*,0,0,0,0,*,0,0,0,0,*,*,0,0,0),$$

the difference pattern of $\mathrm{SR}^{-1}\circ\mathrm{MC}^{-1}(\Delta F(S_9^0))$ is

$$(0,0,0,0,0,0,0,0,0,0,0,0,*,*,*,*).$$

In this case, if there exists i such that $\mathrm{SR}^{-1}\circ\mathrm{MC}^{-1}(\Delta F(S_0^1))[i]$ is zero but $\mathrm{SR}^{-1}\circ\mathrm{MC}^{-1}(\Delta F(S_0^9))[i]$ is nonzero, or vice versa, we say their difference patterns are different.

Proof. We denote the difference pattern of ΔS_0^1 as a and the difference pattern of $F(\Delta S_0^1)$ as α, that is, $\alpha = F(a)$. Similarly, we use b and β to denote the difference patterns of ΔS_9^0 and $F(\Delta S_9^0)$, respectively, then $\beta = F(b)$.

In the forward direction, when the input difference pattern is

$$\Delta S_0 : (0,a,0,0),$$

the first 4-round difference pattern will be

$$\Delta S_0 : (0,a,0,0) \to (a,0,0,0) \to (\alpha,0,0,a) \to (F(\alpha),0,a,\alpha) \to (F^2(\alpha),a,\alpha,F(\alpha)) : \Delta S_4.$$

In addition, in the backward direction, when the output difference pattern of the 9-round distinguisher is

$$\Delta S_9 : (b,0,0,0),$$

the last 4-round difference pattern will be

$$\Delta S_9 : (b,0,0,0) \to (0,b,0,0) \to (0,0,b,\beta) \to (\beta,F(\beta),0,b) \to (b,\beta,F(\beta),F^2(\beta)) : \Delta S_5.$$

As $S_4^2 = S_5^1$, ΔS_4^2 and ΔS_5^1 will share the same difference pattern, that is, there exists at least one value of S_4^2 and S_5^1 with the difference pattern of α and β, respectively, satisfying $\Delta S_4^2 = \Delta S_5^1$. In other words, $F(a)$ and $F(b)$ share the same difference pattern.

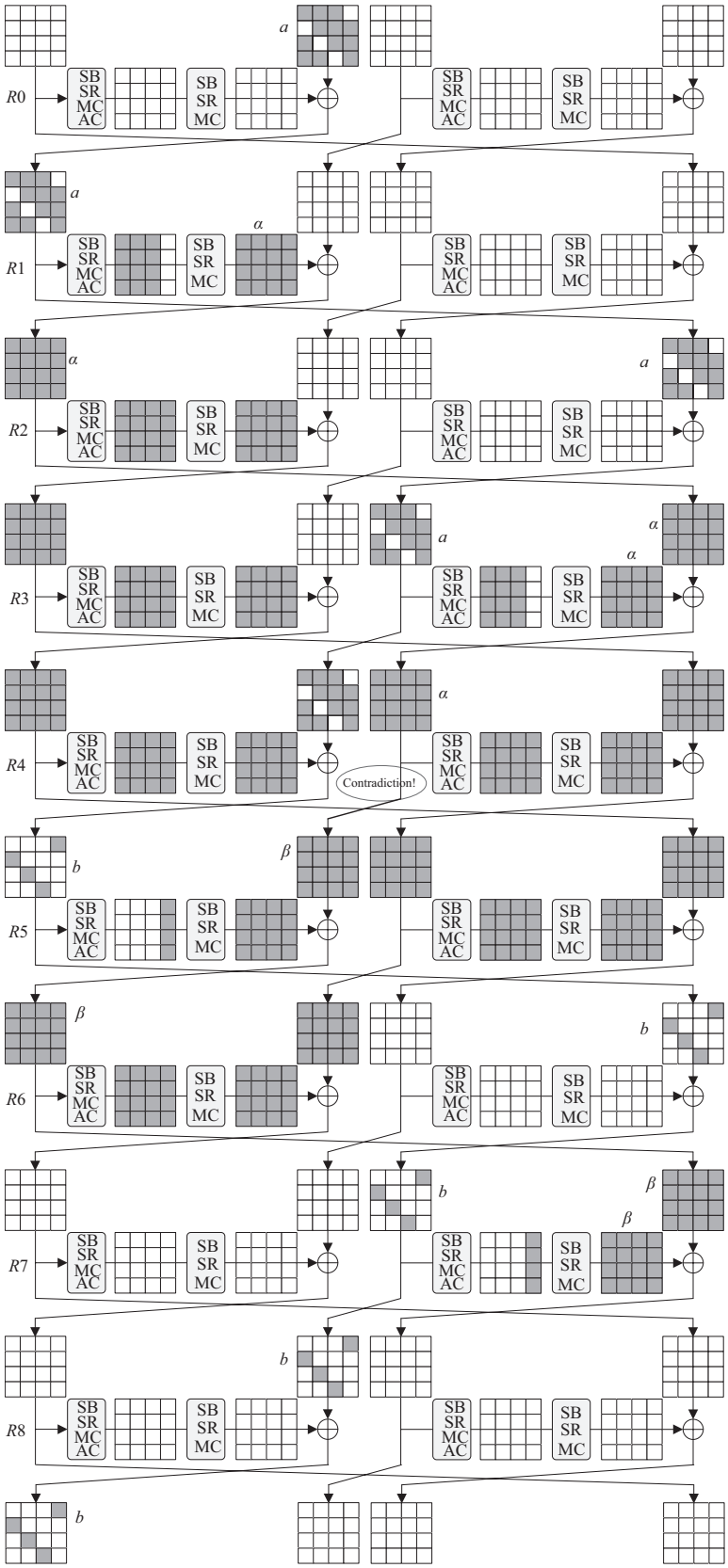

Figure 3 9-round impossible differential.

As the inverse of ShiftRows and MixColumns are both linear operations, values of the same difference pattern will also share the same difference pattern through these operations, that is, $\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(F(a))$ and $\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(F(b))$ will also share the same difference pattern.

This contrasts with our assumption; thus, the observation is proved.

As shown in the above example, when we assume the difference pattern of a as

$$(*, 0, *, *, *, *, 0, *, *, *, *, 0, 0, *, *, *),$$

and the difference pattern of b as

$$(0, *, 0, 0, 0, 0, *, 0, 0, 0, 0, *, *, 0, 0, 0),$$

then the difference pattern of $\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(F(a))$ will be

$$\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(\alpha) = (*, *, *, *, *, *, *, *, *, *, *, *, 0, 0, 0, 0),$$

and the difference pattern of $\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(F(b))$ will be

$$\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(\beta) = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *, *, *, *).$$

Obviously, they are different; thus,

$$(0, a, 0, 0) \xrightarrow{9R} (b, 0, 0, 0)$$

is an impossible differential.

3.2 Attack on 7-round Simpira-4

Owing to the security claim about Simpira according to which an adversary cannot query the distinguisher more than 2^{128} times, we cannot directly use the 9-round distinguisher to mount an attack. Instead, by using the idea of the contradiction in the 9-round distinguisher, we deduced a 6-round impossible distinguisher.

As shown in Figure 4, when S_1^3 is the only active subblock of S_1, then after a 3-round encryption, the difference pattern of S_4 will be

$$(\Delta S_4^0, \Delta S_4^1, \Delta S_4^2, \Delta S_4^3) = (a, \alpha, F(\alpha), 0).$$

Therefore, when $\Delta S_5^0 = \Delta F(S_4^0) \bigoplus \Delta S_4^1$, the difference pattern of ΔS_5^0 will be α.

In addition, when S_7 satisfies $\Delta S_7^1 = b$ and $\Delta S_7^2 = \beta$, in the backward direction, the difference pattern of ΔS_5^0 will be β after a 2-round decryption.

As a result, $\alpha = \beta$, and we obtain the contradiction proved in the 9-round distinguisher; thus, the differential in Figure 4 is impossible.

By adding one round on the top of the 6-round distinguisher, we achieve a 7-round attack on Simpira-4 under the Even-Mansour construction. The differential of the first round is depicted in Figure 5.

The attack process is as follows:

(1) Construct 2^n structures such that each structure is made up of 2^{48} plaintexts. We set $P^0[1, 12]$ and $\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(P^1)[12, 13, 14, 15]$ to be the six active bytes, then each structure provides 2^{95} pairs.

(2) Encrypt the plaintexts and only choose the pairs that satisfy $\Delta C^1 = b$ and $\Delta C^2 = \beta$.

As this is a 64-bit filter, there are approximately $2^{n+95-64} = 2^{n+31}$ pairs remaining after this step.

(3) For each remaining pair, $\Delta S_0^0 = \Delta P^0$. When $\Delta F(S_0^0) = \Delta P^1$, we obtain the input difference of the distinguisher. As ShiftRows and MixColumns are both linear operations, we obtain the value of $S_0^0[1, 6, 11, 12]$ according to the Super S-Box property. XOR the value of $S_0^0[1, 6, 11, 12]$ with the value of $P^0[1, 6, 11, 12]$ to deduce $K^0[1, 6, 11, 12]$, which should be eliminated.

(4) Repeat step(3) until only one value of the 32-bit key value remains; and this is the correct value of $K^0[1, 6, 11, 12]$.

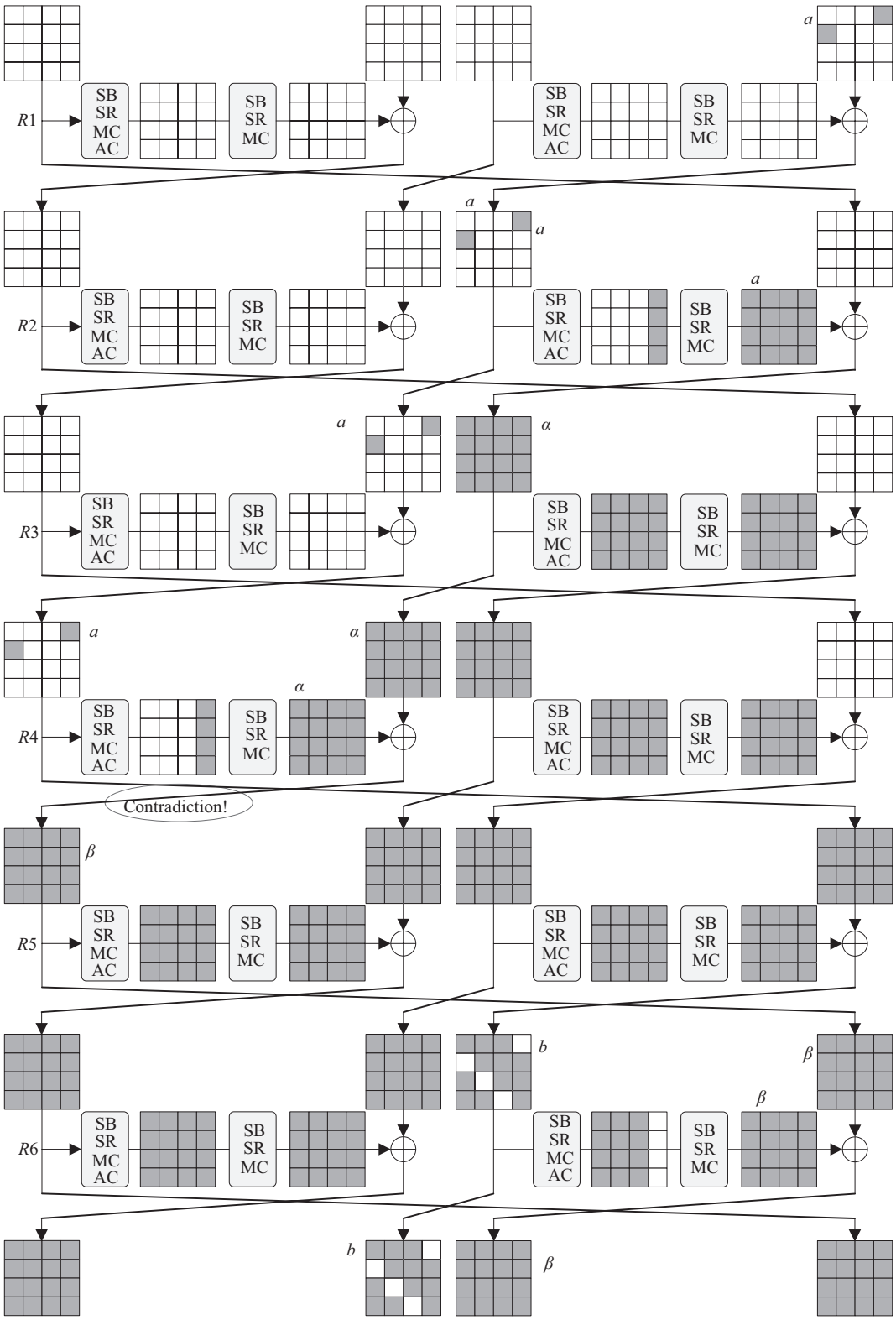

Figure 4 6-round impossible differential.

By changing the positions of the active nibbles of the structure, we can obtain all 256-bit values of K^0 and K^2. Table 1 lists the positions of active nibbles with their corresponding key values.

Complexity. To recover $K^0[1, 6, 11, 12]$, we must analyze the remaining 2^{n+31} pairs. The number of remaining 32-bit key values is $N = 2^{32} \times (1 - 2^{-32})^{2^{n+31}}$. To ensure that $N \approx 1$, we choose $n = 6$. Then,

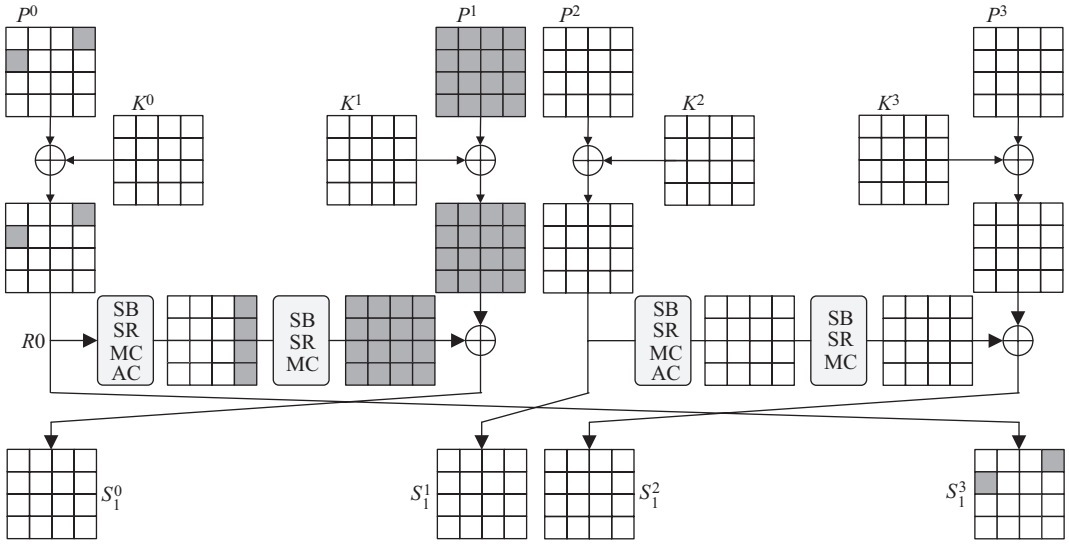

Figure 5 First round of the 7-round attack.

Table 1 Corresponding key bytes of the 7-round attack

$\Delta P^0(\Delta P^2)$	$\Delta C^1(\Delta C^3)$	Responding bytes of $K^0(K^2)$
(*00000000000000*)	(0****0****0****0)	[0,5,10,15]
(0*0000000000*000)	(*0****0****00***)	[1,6,11,12]
(00*0000000000*00)	(**0****00****0**)	[2,7,8,13]
(000*0000000000*0)	(***00****0****0*)	[3,4,9,14]

the data complexity is 2^{54} chosen plaintexts. The time complexity of the attack is obviously dominated by encrypting the plaintexts, and is thus 2^{54} encryption units. Similarly, to recover the other 224-bit values of K^0 and K^2, we must repeat a similar attack procedure eight times. That is, to recover K^0 and K^2, the data complexity is 2^{57} chosen plaintexts and the time complexity is 2^{57} encryption units.

3.3 Attack on 8-round Simpira-4

Owing to the security claim of the Even-Mansour scheme, we could not use the 9-round distinguisher to attack Simpira-4. To attack the 8-round Simpira-4, we propose a 7-round distinguisher (Figure 6). Its key concept is also the same as that of the 9-round distinguisher. When the input difference $\Delta S_1 = (0, a, 0, 0)$ and the output difference $\Delta S_8 = (*, 0, b, \beta)$, we obtain the contradiction.

By using the 7-round impossible differential, we recover all 512-bit key values of Simpira-4 under the Even-Mansour construction. The attack can be partitioned into two phases:

(a) Mounting an 8-round attack to recover the 256-bit key by adding one round on the top of the 7-round impossible differential;

(b) Recovering the other 256-bit key by adding one round on the bottom of the distinguisher.

Figure 7 depicts the difference characteristic of the first round. The process of the first phase is as follows.

Phase a:

(1) Construct 2^n structures such that plaintexts in each structure traverse 8 bytes: $P^2[1, 6, 11, 12]$ and $\mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(P^3)[12, 13, 14, 15]$. Thus, in each structure, there are 2^{64} plaintexts providing 2^{127} pairs.

(2) Encrypt the plaintexts in each structure, and only choose the pairs that satisfy: (a) $\Delta C^1 = 0$; (b) $\Delta C^2 = b$; and (c) $\Delta C^3 = \beta$. This step performs a 192-bit filter; thus, we expect approximately $2^{n+127-192} = 2^{n-65}$ pairs.

(3) For each remaining pair, we can directly obtain the values of ΔS_0^2 and ΔS_0^3 from ΔP^2 and ΔP^3, respectively. When $\Delta F(S_0^2) = \Delta S_0^3$, S_1^1 will be the only active subblock in S_1; thus we obtain the input difference of the distinguisher. By applying the differential property of the Super S-box, we can

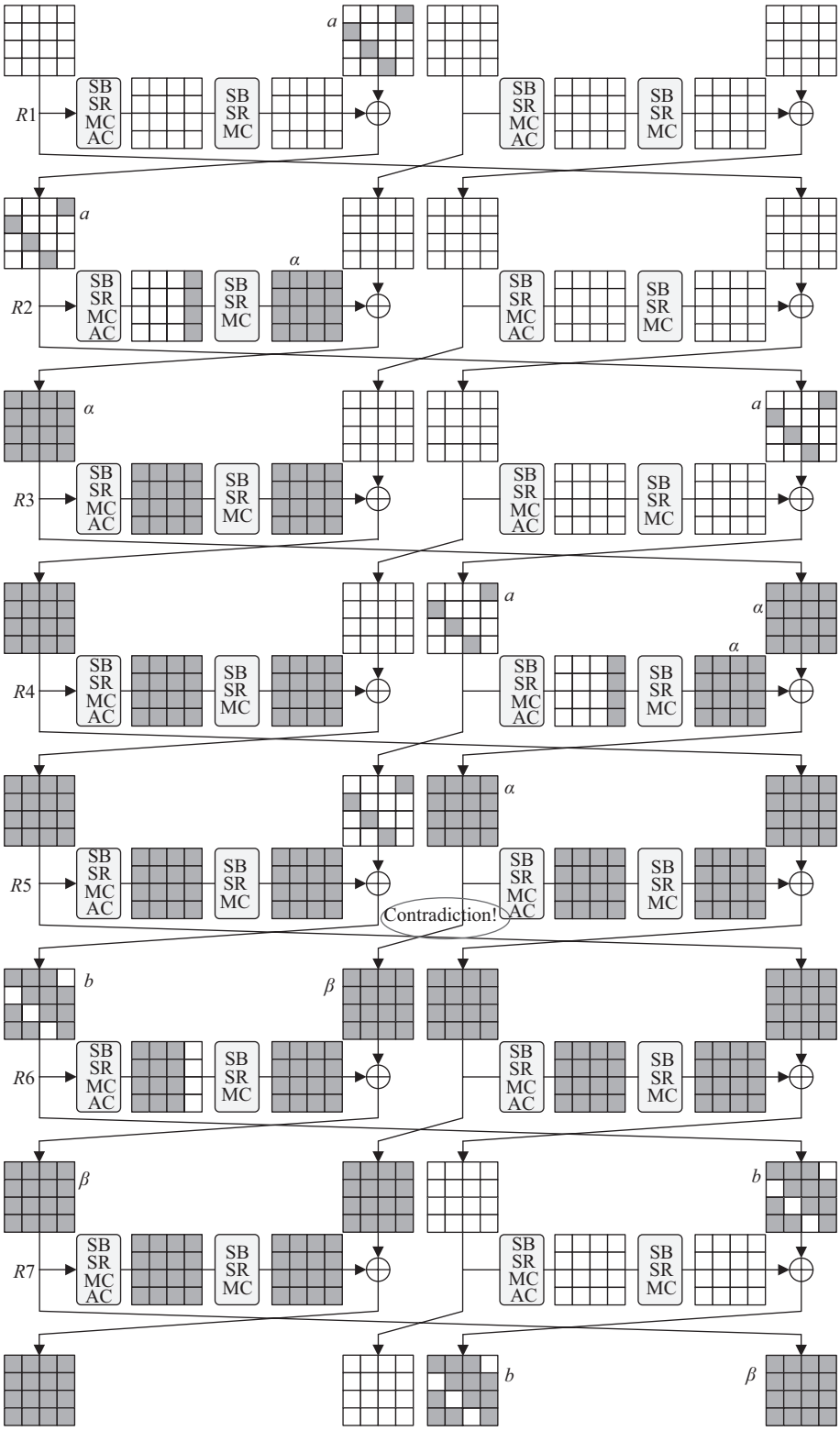

Figure 6 7-round distinguisher.

easily obtain the value of $S_0^2[1,6,11,12]$. Furthermore, by combining $S_0^2[1,6,11,12]$ and $P^2[1,6,11,12]$, we obtain one wrong value of $K^2[1,6,11,12]$.

(4) Repeat step (3) until there is only one value of $K^2[1,6,11,12]$ remaining; this is the correct value. By changing the position of active bytes of P^2 and C^2, we can recover all 256-bit values of K^2 as shown

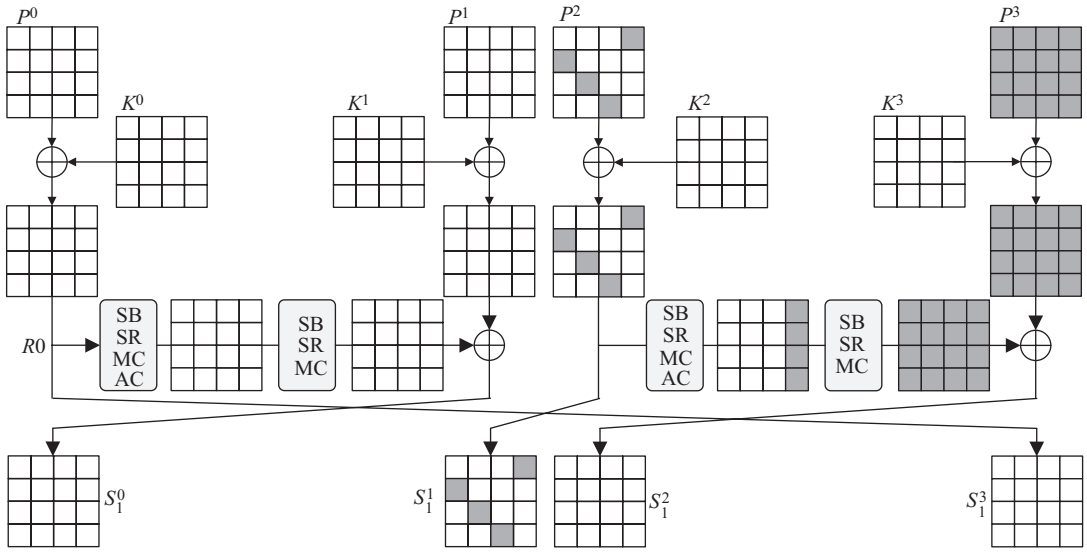

Figure 7 First round of Phase a.

Table 2 Corresponding key bytes of Phase a

$\Delta P^2(\Delta P^0)$	$\Delta C^2(\Delta C^0)$	Responding bytes of $K^2(K^0)$
(0*0000*0000**000)	(*0****0****00***)	[1,6,11,12]
(00*0000**0000*00)	(**0****00****0**)	[2,7,8,13]
(000**0000*0000*0)	(***00****0****0*)	[3,4,9,14]
(*0000*0000*0000*)	(0****0****0****0)	[0,5,10,15]

in Table 2.

Complexity. To recover $K^2[1,6,11,12]$, we must analyze 2^{n-65} pairs. The number of remaining 32-bit key values is $N = 2^{32} \times (1-2^{-32})^{2^{n-65}}$. To ensure that $N \approx 1$, we chose $n = 102$. Then, the data complexity is 2^{166} chosen plaintexts, and the time complexity of the attack is 2^{166} 8-round encryptions. Similarly, to recover all the bits of K^0 and K^2, we repeat the same procedure eight times. Thus, the data complexity is 2^{169} chosen plaintexts and the time complexity is 2^{169} encryption units.

Till now, we recovered all 256-bit values of K^0 and K^2. Next, we mount an attack to recover K^1 and K^3 by adding one round on the bottom of the 7-round distinguisher. The differential trail of the last round is shown in Figure 8.

Phase b:

(1) Construct 2^n structures such that each structure is made up of 2^{32} plaintexts that traverses $P^1[1,6,11,12]$. We expected to obtain 2^{n+63} pairs.

(2) Encrypt the plaintexts and only choose the pairs that satisfy (a) $\Delta C^1 = b$ and (b) $\Delta C^2 = \beta$. This step performs a 64-bit filter; thus, after this step, approximately 2^{n-1} pairs remain.

(3) For each remaining pair, we set $\Delta F(S_8^0) = \Delta C^0$, then $\Delta S_8^1 = 0$, and obtained the output difference of the impossible distinguisher. By using the property of the Super S-box, we obtained a value S_9^3. After XORing this value of S_9^3 with the value of C^3, we obtained the value of K^3 and deleted it.

(4) Repeat step (3) until there is only one value of K^3 remaining. Similar to phase a, we can recover all 256-bit values of K^1 and K^3 by mounting two analogous attacks.

Complexity. To enure that there is only one value of the 128-bit key remaining after the attack process, $N = 2^{128} \times (1-2^{-128})^{2^{n-1}}$ should be approximately equal to 1. We choose $n = 136$; then, the data complexity is approximately 2^{168} chosen plaintexts, and the time complexity is approximately 2^{168} encryption units to encrypt the plaintexts. As we must mount two attack process to recover all 256 bits of K^1 and K^3, the data complexity becomes 2^{169} plaintexts and the time complexity becomes 2^{169} encryption units.

Therefore, we require a final data complexity of 2^{170} chosen plaintexts and a time complexity of 2^{170}

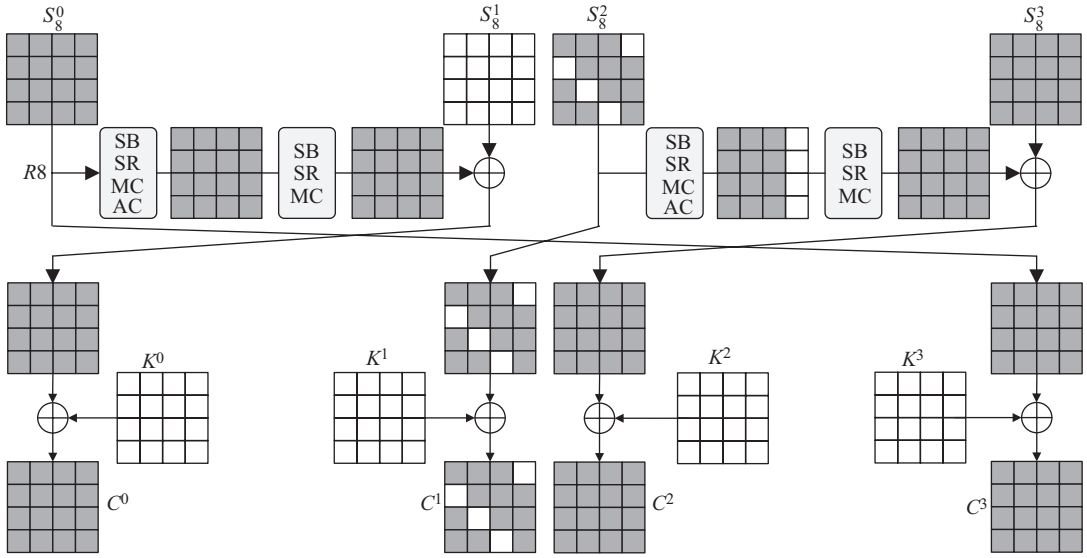

Figure 8 Last round of Phase b.

to recover all 512-bit key values.

4 Conclusion

In this paper, we proposed a 9-round impossible differential on Simpira-4. To the best of our knowledge, this is the first impossible distinguisher of Simpira-4. By using the same contradiction as in the 9-round distinguisher, we proposed a 6-round distinguisher and achieved a 7-round attack on Simpira-4 under the Even-Mansour construction with a data complexity of 2^{57} plaintexts and a time complexity of 2^{57} encryption units to recover a 256-bit key. Next, we presented an attack on an 8-round Simpira-4 under the Even-Mansour construction. By using 2^{170} plaintexts and 2^{170} encryption units, we recovered all 512 bits of the master key. These two attacks aim at contradicting two security claims of the Even-Mansour scheme and the Simpira-4 permutation. As far as we know, this is the first result of impossible differential attacks on Simpira v2.

Acknowledgements This work was supported by National Basic Research Program of China (973 Program) (Grant No. 2013CB834205), National Natural Science Foundation of China (Grant No. 61672019), Fundamental Research Funds of Shandong University (Grant No. 2016JC029), and Foundation of Science and Technology on Information Assurance Laboratory (Grant No. KJ-15-002).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Daemen J, Rijmen V. The Design of Rijndael. Berlin: Springer, 2002
2 Mala H, Dakhilailian M, Rijmen V, et al. Improved impossible differential cryptanalysis of 7-round AES-128. In: Proceedings of International Conference on Cryptology in India. Berlin: Springer-Verlag, 2010. 282–291
3 Lu J Q, Dunkelman O, Keller N, et al. New impossible differential attacks on AES. In: Proceedings of International Conference on Cryptology in India. Berlin: Springer-Verlag, 2008. 279–293
4 Zhang W T, Wu W L, Feng D G. New results on impossible differential cryptanalysis of reduced AES. In: Proceedings of International Conference on Information Security and Cryptology. Berlin: Springer-Verlag, 2007. 239–250
5 Daemen J, Knudsen L, Rijmen V. The block cipher Square. In: Proceedings of International Workshop on Fast Software Encryption. Berlin: Springer-Verlag, 1997. 149–165
6 Gilber H, Minier M. A collision attack on 7 rounds of Rijndael. In: Proceedings of AES Candidate Conference, New York, 2000. 230–241
7 Dunkelman O, Keller N, Shamir A. Improved single-key attacks on 8-round AES-192 and AES-256. In: Advances in Cryptology — ASIACRYPT 2010. Berlin: Springer-Verlag, 2010. 158–176

8 Derbez P, Fouque P, Jean J. Improved key recovery attacks on reduced-round AES in the single-key setting. In: Advances in Cryptology — EUROCRYPT 2013. Berlin: Springer-Verlag, 2013. 371–387

9 Li L B, Jia K T, Wang X Y. Improved single-key attacks on 9-round AES-192/256. In: Fast Software Encryption. Berlin: Springer-Verlag, 2015. 127–146

10 Biryukov A, Khovratovich D. Related-key cryptanalysis of the full AES-192 and AES-256. In: Advances in Cryptology — ASIACRYPT 2009. Berlin: Springer-Verlag, 2009. 1–18

11 Biryukov A, Khovratovich D, Nikolić I. Distingsuiher and related-key attack on the full AES-256. In: Advances in Cryptology — CRYPTO 2009. Berlin: Springer-Verlag, 2009. 231–249

12 Sun B, Liu M C, Guo J, et al. New insights on AES-like SPN ciphers. In: Advances in Cryptology — CRYPTO 2016. Berlin: Springer-Verlag, 2016. 605–624

13 Grassi L, Rechberger C, Rønjom S. Subspace trail cryptanalysis and its applications to AES. IACR Trans Symmetric Cryptol, 2016, 2016: 192–225

14 Gueron S, Mouha N. Simpira v2: a family of efficient permutations using the AES round function. In: Advances in Cryptology — ASIACRYPT 2016. Berlin: Springer-Verlag, 2016. 95–125

15 Dunkelman O, Keller N, Shamir A. Minimalism in cryptography: the Even-Mansour scheme revisited. In: Advances in Cryptology — EUROCRYPT 2012. Berlin: Springer-Verlag, 2012. 336–354

16 Even S, Mansour Y. A construction of a cipher from a single pseudorandom permutation. J Cryptology, 1997, 10: 151–161

17 Dobraunig C, Eichlseder M, Mendel F. Cryptanalysis of Simpira v1. In: Selected Areas in Cryptography, Newfoundland, 2016, in press

18 Rønjom S. Invariant subspaces in Simpira. Cryptology ePrint Archive, Report, 2016. http://eprint.iacr.org/2016/248.pdf

19 Knudsen L R. DEAL — a 128-bit block cipher. Complexity, 1998, 258: 216

20 Biham E, Biryukov A, Shamir A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Advances in Cryptology — EUROCRYPT 1999. Berlin: Springer-Verlag, 1999. 12–23

21 Sun S W, Hu L, Wang M Q, et al. Constructing mixed-integer programming models whose feasible region is exactly the set of all valid differential characteristics of SIMON. Cryptology ePrint Archive, Report, 2015. http://eprint.iacr.org/2015/122.pdf

22 Sun S W, Hu L, Wang M Q, et al. Mixed integer programming models for finite automaton and its application to additive differential patterns of exclusive-or. Cryptology ePrint Archive, Report, 2016. http://eprint.iacr.org/2016/338.pdf

23 Cui T T, Jia K T, Fu K, et al. New automatic search tool for impossible differentials and zero-correlation linear approximations. Cryptology ePrint Archive, Report, 2016. http://eprint.iacr.org/2016/689.pdf

24 Daemen J, Rijmen V. Understanding two-round differentials in aes. In: Proceedings of International Conference on Security and Crytography for Networks. Berlin: Springer-Verlag, 2006. 78–94