Impossible Differential Attack on Simpira v2

Rui Zong[1], Xiaoyang Dong[1], and Xiaoyun Wang[2]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education,
ShandongUniversity, China
{zongrui,dongxiaoyang}@mail.sdu.edu.cn
[2] Institute for Advanced Study, Tsinghua University, China

Abstract. Simpira v2 is a family of cryptographic permutations proposed at ASIACRYPT 2016 which can be used to construct high throughput block ciphers using the Even-Mansour construction, permutation-based hashing and wide-block authenticated encryption. In this paper, we give a 9-round impossible differential of Simpira-4, which turns out to be the first 9-round impossible differential. In order to get some efficient key recovery attacks on its block cipher mode (EM construction with Simpira-4), we use some 6/7-round shrunken impossible differentials. Based on eight different 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 to recover 32-bit of the master key (512-bit) and totally half of the master key bits are recovered. The attacks need 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, which recover the full key space (512-bit) with the data complexity of 2^{170} chosen plaintexts and time complexity of 2^{170} 8-round encryptions. Those are the first attacks on round-reduced Simpira v2 and do not threaten the EM mode with the full 15-round Simpira-4.

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

1 Introduction

Since the block cipher Rijndael [1] designed by Daemen and Rijmen was selected as the Advanced Encryption Standard (AES) in 2001 by NIST, it has been researched worldwide by various cryptanalysis methods, e.g., impossible attack [2,3,4], SQUARE attack [5], collision attack [6] and meet-in-the-middle attack [7,8,9] et al. 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 are proposed [12,13], and extend the long standing 4-round distinguisher by 1 round.

Nowadays, Intel, AMD and ARM all introduce AES instructions to their modern processors to reduce the encryption overheads. To design a permutation based on the AES round function becomes a meaningful project as when used in software implementation, it can introduce the AES instruction directly. As there are proposed cipher suites that allows the message blocks can be processed independently for encryption, the fixed block size of AES becomes a limitation.

To achieve a higher throughput, Shay Gueron and Nicky 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, $x \in \mathbb{N}^+$. Meanwhile, 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 use a Feistel Structure for $x \geq 2$ that operates on x input subblocks of 128 bits each.

One application of Simpira recommended by the designer is to be used as the permutation in the Even-Mansour construction [15,16] to construct 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]. And also, the designer give a security claim about the permutation that Simpira can be used in constructions where a adversary can not query a distinguisher more than 2^{128} times.

There are two related works both focus on Simpira v1. 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 propose full-round collision attacks on the proposed Davies-Meyer hash constructions based on Simpira v1 with $x = 4$. In addition, Sondre Rønjom reported on invariant subspaces in Simpira v1 with $x = 4$ [18]. He showed that the whole coset of dimension 56 over $\mathbb{F}_{2^8}^{64}$, and these invariant subspaces result from the AES based round function together with the particular choice of Feistel configuration.

To solve these problems, the designer give Simpira v2 by ensuring that every subblock will only be operated once. Simpira v2 has more complex round constants and uses a more logical Feistel Structure. Without other statements, we use Simpira to denote Simpira v2 for short in the following.

In this paper, we explore the security of Simpira v2 against impossible differential cryptanalysis. Impossible differential cryptanalysis was independently proposed by Knudsen [19] and Biham [20]. Its main idea is to use impossible differentials that hold with probability zero to discard the wrong keys until only one key is left. Recently, inspired by Sun's work [21,22], a new automatic search tool [23] for searching impossible differentials is proposed.

Our Contribution. In this paper, we will focus on the block cipher mode of Simpira v2 with four branches ($x = 4$), i.e. the Even-Mansour construction with Simpira-4. We first present a 9-round impossible distinguisher which turns out to be the first 9-round impossible differential on Simpira v2 with $x = 4$. In addition, we mount two impossible differential key recovery attacks: one is on 7-round Simpira with $x = 4$ with 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 is on 8-round Simpira with $x = 4$ with a data complexity of 2^{170} plaintexts and a time complexity of 2^{170} encryption units to recover all 512 key bits with a 7-round impossible differentials.

2 Preliminaries

2.1 Notation

$\bigoplus$	bitwise XOR
P	the plaintext
C	the ciphertext
S	the internal state
F	the basic building block of Simpira
ΔS	the difference between S and $S^{'}$
S_h	the input of the h^{th} round, $h \geq 0$
S^i	the i^{th} subblock of S, $i \in \{0,1,2,3\}$
$S^i[j]$	the j^{th} byte of S^i, $j \in \{0, 1, 2, \cdots, 15\}$
Simpira-x	Simpira with x subblocks, $x \in \mathbb{N}^+$
0	nibbles and subblocks with zero difference
*	nibbles and subblocks with nonzero difference
f^{-1}	the 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 permutatioins that supports of $128 \times x$ bits where x is a positive integer. Its design goal is to achieve high throughput on virtually all modern 64-bit processor architectures.

We will only give the detail of Simpira-4 as all attacks are on it. For more about Simpira, we refer to [14].

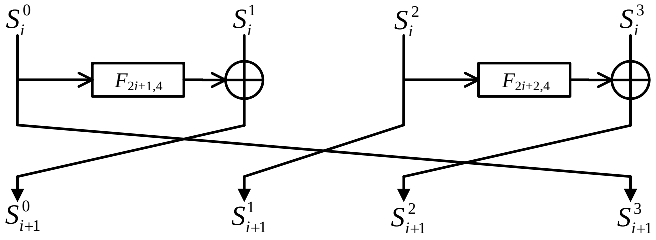

Fig. 1. Round Function of Simpira-4

The round function of Simpira-4 is as shown in Figure 1, so the state update rule will be as follows with $0 \leq i \leq 14$:

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

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, i.e. 4 for Simpira-4 and c is a counter counted from 1. It is made up of two rounds of AES while omitting the second AddRoundKey operation. The only difference is that the specific round constant updating process, i.e. AddRoundKey in AES round function. But beyond that, SubBytes, ShiftRows, MixColumns are identical as AES. For more detail, we refer to [1].

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

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

We also refer to s_i as $S[i]$.

To be convenient when referring to the internal states inside the F function for an input S, we use the same notation as in [17]:

$$S \xrightarrow{SB} S^{SB_1} \xrightarrow{SR} S^{SR_1} \xrightarrow{MC} S^{MC_1} \xrightarrow{AC} S^{AC} \xrightarrow{SB} S^{SB_2} \xrightarrow{SR} S^{SR_2} \xrightarrow{MC} S^{MC_2} = F(S).$$

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.

2.4 Our Attack Assumptions

In this paper, we focus 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 such as the Even-Mansour scheme with 512-bit key.

In 2012 [15], Dunkelman and Shamir gave a security claim that when D plaintext-ciphertexts are available, the secret key K of the Even-Mansour construction can be recovered in $2^n/D$(off-line) 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.

Meanwhile, the designer also gave a security claim about Simpira [14]: Simpira can be used in construction that require a random permutation, however no statements can be made for adversaries that exceed 2^{128} queries. Due to this occasion, the data complexity and the time complexity of the attack both should be less than 2^{128}.

As a result, for different security claims, we mount two attacks of the Simpira-4 basing Even-Mansour construction: one is on 7-round Simpira-4 with a data complexity of 2^{57} plaintexts and a time complexity of 2^{57} encryption units and the other one is on 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 firstly present some useful observations and properties of Simpira-4, and then present the impossible differential distinguisher and the attack procedure.

3.1 Some Observations

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

Definition *(Super Sbox). The AES Super Sbox 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 the sequence of four transformations: Subbytes, MixColumns, AddRoundKey and SubBytes.*

Property*(Differential Property of Super Sbox)Given $\Delta input$ and $\Delta output$ two non-zero difference in F_2^{32}, the equation of Super Sbox:*

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

has one solution in average for each key value.

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

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

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

and swap the order of the first SubBytes operation and the first ShiftRows operation (Figure 2) to obtain an integrated Super Sbox structure.

Then after the ShiftRows operation, the difference pattern will be:

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

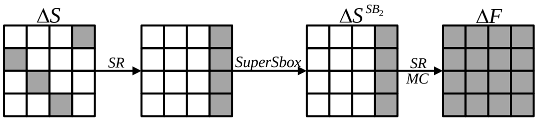

Fig. 2. Super Sbox of AES

The difference pattern of the output of the Super Sbox will be:

$$\Delta S^{SB_2} = (0,0,0,0,0,0,0,0,0,0,0,0,*,*,*,*),$$

so 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) If ΔS_0^1 is the one and 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 pattern of $SR^{-1}\circ MC^{-1}(\Delta F(S_0^1))$ and $SR^{-1}\circ MC^{-1}(\Delta F(S_9^0))$ are not same.

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

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

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

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

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 $SR^{-1}\circ MC^{-1}(\Delta F(S_9^0))$ is

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

In this case, as there exists i that $SR^{-1}\circ MC^{-1}(\Delta F(S_0^1))[i]$ is zero but $SR^{-1}\circ MC^{-1}(\Delta F(S_0^9)[i]$ 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, e.g., $\alpha = F(a)$. 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:

$$\Delta S_0 = (0,\Delta S_0^1,0,0),$$

the first 4-round difference pattern will be as follows:

$$\Delta S_0 = (0,a,0,0) \to (a,0,0,0) \to (\alpha,0,0,a) \to (F(\alpha),0,a,\alpha) \to (F^2(\alpha),a,\alpha,F(\alpha)) = \Delta S_4;$$

and in the backward direction, when the output difference of the 9-round distinguisher:

$$\Delta S_9 = (\Delta S_9^0,0,0,0),$$

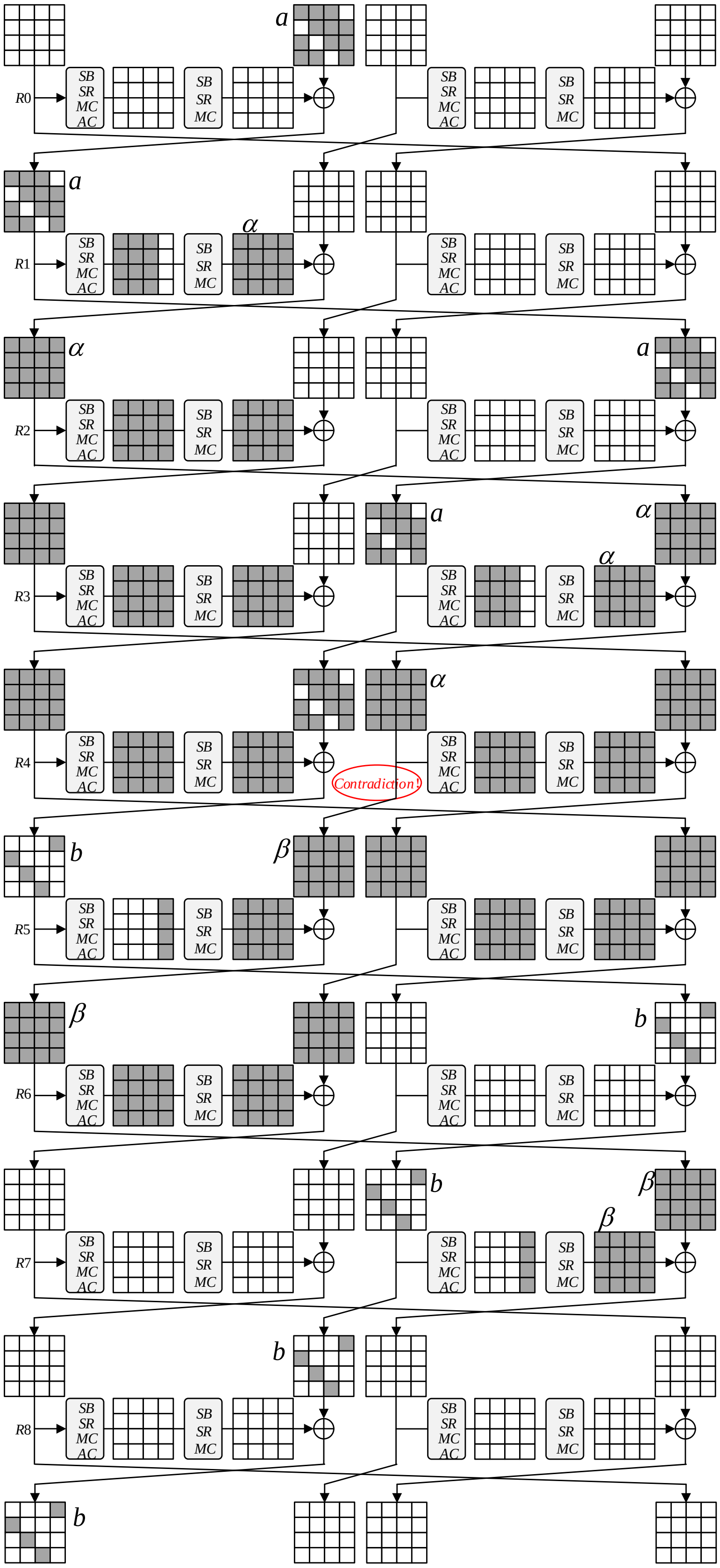

Fig. 3. The 9-Round Impossible Differential

the last 4-round differential 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 $S_4^2 = S_5^1$, the difference pattern of ΔS_4^2 will be same as the difference pattern of ΔS_5^1, that means there exists at least one value of S_4^2 that has the difference pattern α, and S_5^1 that has the difference pattern β, satisfying that $\Delta S_4^2 = \Delta S_5^1$, e.g. $\Delta F(S_0^1) = \Delta F(S_9^0)$.

Since the inverse of ShiftRows and MixColumns are both linear operations, values of the same difference pattern will also share a same difference pattern through these operations, e.g.

$$SR^{-1} \circ MC^{-1}(\Delta F(S_0^1)) = SR^{-1} \circ MC^{-1}(\Delta F(S_9^0)).$$

That is contradict to our assumption, so the observation is proved.

As shown in the above example, when we suppose that the difference pattern of a as:

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

and the difference pattern of b as:

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

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

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

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

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

Obviously, they are different, then

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

is an impossible differential. □

3.2 Attack on 7-round Simpira-4

Because of the security claim about Simpira that an adversary can not query the distinguisher more than 2^{128} times, we can not directly use the 9-round distinguisher to mount an attack. Instead, using the idea of the contradiction in the 9-round distinguisher, we deduce a 6-round impossible distinguisher.

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

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

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

When S_7 satisfies that $\Delta S_7^1 = b$ 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, $\alpha = \beta$, we get the contradiction proved in the 9-round distinguisher, so 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 that each structure is made up of 2^{48} plaintexts. We set $P^0[1, 12]$ and $SR^{-1} \circ MC^{-1}(P^1)[12, 13, 14, 15]$ to be the six active bytes, then each structure will provide 2^{95} pairs.

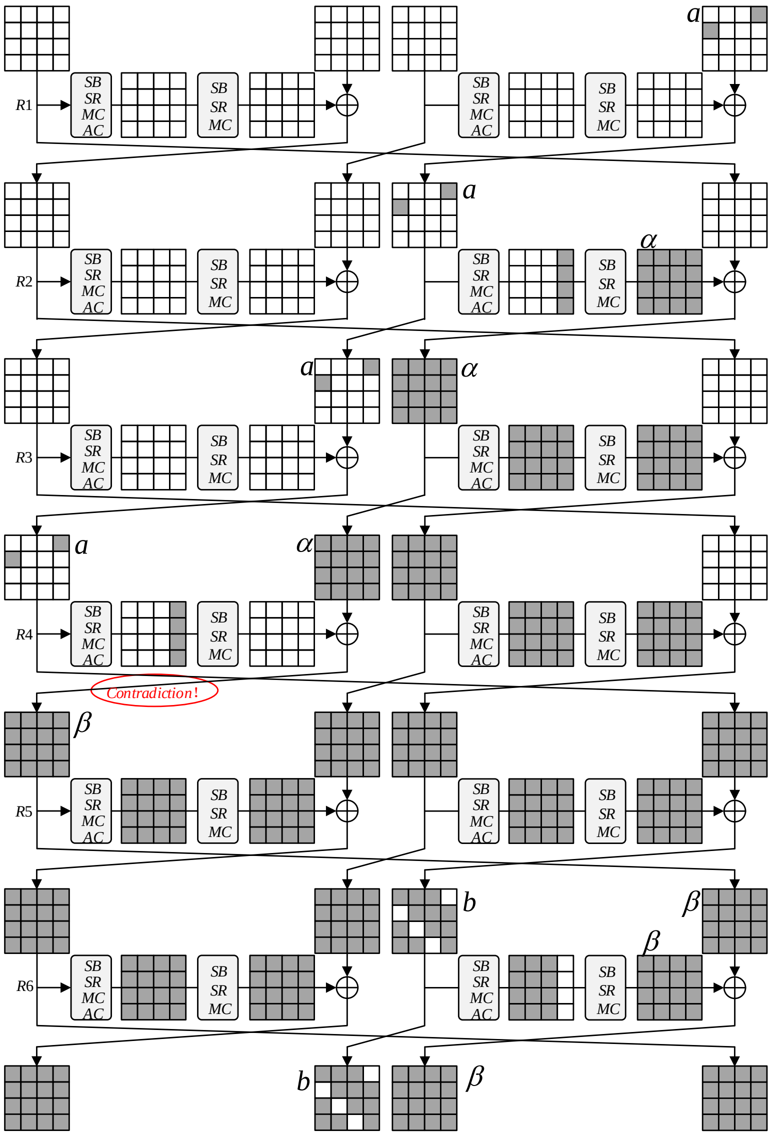

Fig. 4. A 6-Round Impossible Differential

(2) Encrypt the plaintexts and only choose the pairs that satisfy $\Delta C^1 = b$ and $\Delta C^2 = \beta$. This is a 64-bit filter, after this step, about $2^{n+95-64} = 2^{n+31}$ pairs leave in total.

(3) For each remaining pair, ΔS_0^0 is equal to ΔP^0. When $\Delta F(S_0^0)$ is equal to ΔP^1, we get the input difference of the distinguisher. As ShiftRows and MixColumns are both linear operations, with the property of Super S-Box, we get the value of $S_0^0[1,6,11,12]$. 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 the step(3) until there are only one value of the 32-bit key value left and that is the right value of $K^0[1,6,11,12]$.

By changing the positions of active nibbles of the structure, we can get all 256-bit value 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 need to analyze the remaining 2^{n+31} pairs. The number of remaining 32-bit key values is $N = 2^{32} \times (1 - 2^{-32})^{2^{n+31}}$. In order to make sure that $N \approx 1$, we choose $n = 6$. Then the data complexity is 2^{54} chosen plaintexts. The time complexity of the attack is

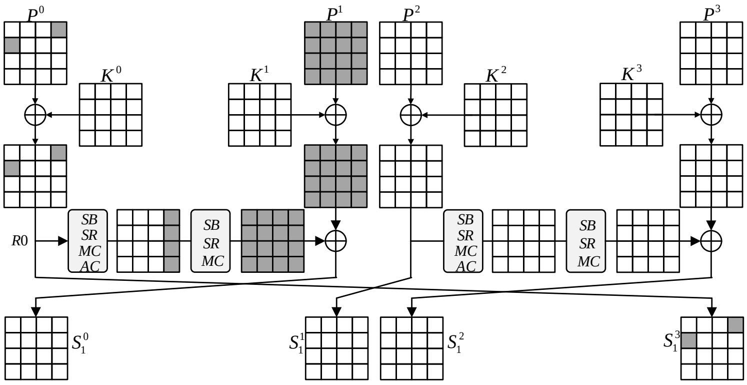

Fig. 5. First Round of the 7-Round Attack

Table 1. Corresponding Key Bytes of the 7-Round Attack

$\varDelta P^0(\varDelta P^2)$	$\varDelta C^1(\varDelta C^3)$	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]

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

3.3 Attack on 8-round Simpira-4

As the security claim of the Even-Mansour scheme, we could not use the 9-round distinguisher to attack Simpira-4 either. To attack 8-round Simpira-4, we propose a 7-round distinguisher. Its key idea is also same as that in the 9-round distinguisher. When the input difference $\varDelta S_1 = (0, a, 0, 0)$ and the output difference $\varDelta S_8 = (*, 0, b, \beta)$, we will get the contradiction.

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

(a) By adding one round on the top of the 7-round impossible differential, we mount an 8-round attack to recover 256-bit key;
(b) Recover the other 256-bit key by adding one round on the bottom of the distinguisher.

The difference characteristic of the first round is depicted in Figure 7. The process of the first phase is as follows:

Phase a:

(1) Construct 2^n structures that plaintexts in each structure traverses 8 bytes: $P^2[1, 6, 11, 12]$ and $SR^{-1} \circ MC^{-1}(P^3)[12, 13, 14, 15]$.
As a result, 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) $\varDelta C^1 = 0$; (b) $\varDelta C^2 = b$; (c) $\varDelta C^3 = \beta$.
This step performs a 192-bit filter, so we expect about $2^{n+127-192} = 2^{n-65}$ pairs left in total.

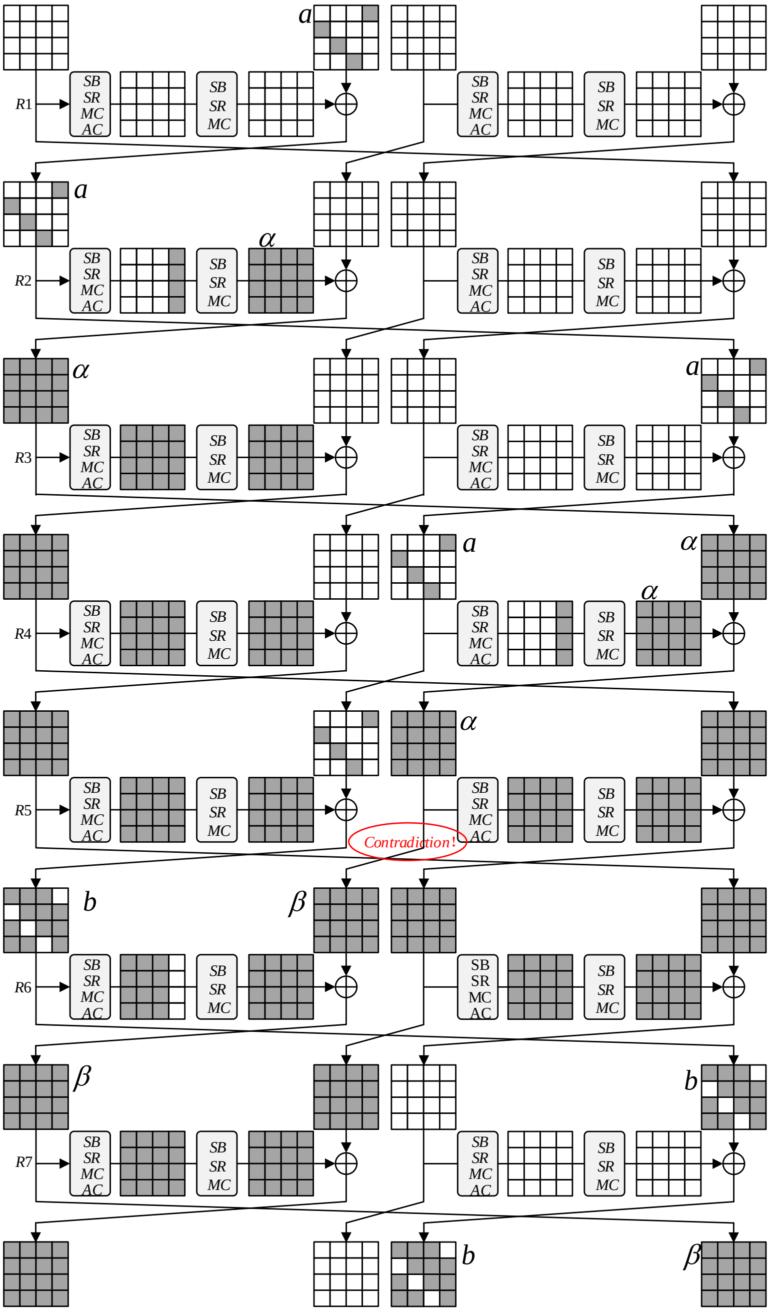

Fig. 6. The 7-Round Distinguisher

(3) For each left pair, we can directly get the value 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 get the input difference of the distinguisher.

By applying the differential property of the Super Sbox, we can easily get the value of $S_0^2[1, 6, 11, 12]$. Combining $S_0^2[1, 6, 11, 12]$ and $P^2[1, 6, 11, 12]$, we get one wrong value of $K^2[1, 6, 11, 12]$.

(4) Repeat the step(3) until there are only one value of $K^2[1, 6, 11, 12]$ remaining, and that is the right value.

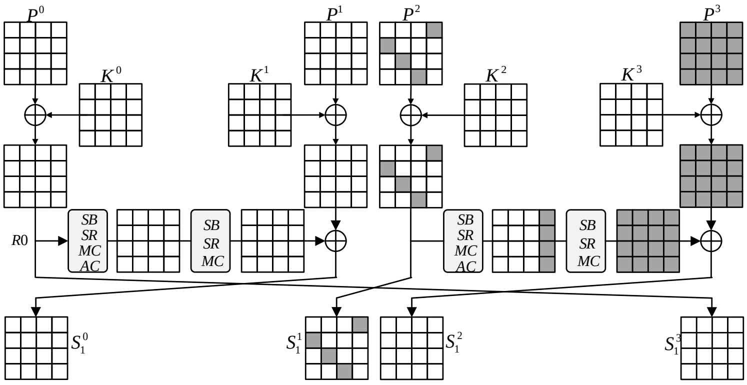

Fig. 7. First Round of Phase a

By changing the position of active bytes of P^2 and C^2, we can recover all 256-bit value of K^2 as shown in Table 2.

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]

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

Till now, we recover all 256-bit value of K^0 and K^2. In the following 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 that each structure is made up of 2^{32} plaintexts that traverses $P^1[1,6,11,12]$. We expected to get 2^{n+63} pairs in total.

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

(3) For each remaining pair, we set $\Delta F(S_8^0) = \Delta C^0$, then $\Delta S_8^1 = 0$, we get the output difference of the impossible distinguisher. By using the property of the Super Sbox, we get the value of S_9^3. Xor the value of S_9^3 with the value of C^3, we get the value of K^3 and delete it.

(4) Repeat the step(3) until there is only one value of K^3 remaining.

Similarly as phase a, we can recover all 256-bit value of K^1 and K^3 by mounting two analogous attacks.

Complexity. To make sure that there is only one value of 128-bit key value 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

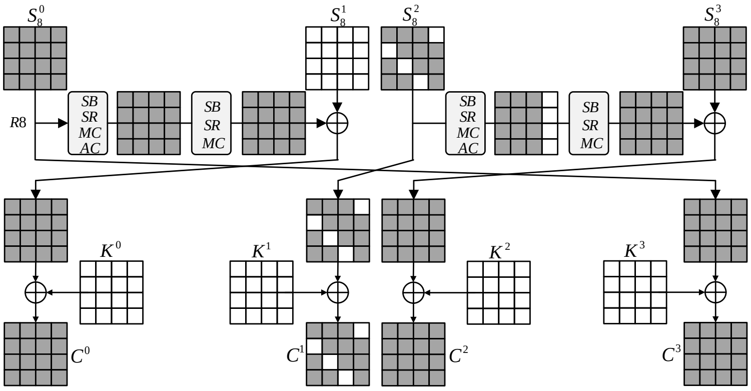

Fig. 8. Last Round of Phase b

the data complexity is about 2^{168} chosen plaintexts, the time complexity is about 2^{168} encryption units to encrypt the plaintexts. As we need to mount two attack process to recover all 256-bit of K^1 and K^3, so the data complexity is 2^{169} plaintexts, the time complexity is 2^{169} encryption units.

So in total, we need a data complexity of 2^{170} chosen plaintexts and a time complexity of 2^{170} to recover all 512-bit key value.

4 Conclusion

In this paper, we propose a 9-round impossible differential on Simpira-4, to our best knowledge, this is the first impossible distinguisher of Simpira-4. By using the same contradiction as in the 9-round distinguisher, we propose a 6-round distinguisher and achieve 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 256 bits key. After that, we present an attack on 8-round Simpira-4 under the Even-Mansour construction. By using 2^{170} plaintexts and 2^{170} encryption units, we recover all 512-bit key. These two attacks aim at two different security claims of the Even-Mansour scheme and the Simpira-4 permutation respectively. As far as we know, this is the first result of impossible differential attacks on Simpira v2.

References

1. Daemen J, Rijmen V. The Design of Rijndael. AES-the Advanced Encryption Standard. Springer, 2002.
2. Mala H, Dakhilailian M, Rijmen V, et al. Improved impossible differential cryptanalysis of 7-round AES-128. In: INDOCRYPT, Berlin/Heidelberg: Springer-Verlag, 2010. 282-291
3. Lu J Q, Dunkelman O, Keller N, et al. New impossible differential attacks on AES. In: INDOCRYPT, Berlin/Heidelberg: Springer-Verlag, 2008, 5365: 279-293
4. Zhang W T, Wu W L, Feng D G. New results on impossible differential cryptanalysis of reduced AES. In: ICISC, Berlin/Heidelberg: Springer-Verlag, 2007, 4817: 239-250
5. Daemen J, Knudsen L, Rijmen V. The block cipher Square. In: FSE, Haifa: Springer-Verlag, 1997, 1267: 149-165
6. Gilber H, Minier M. A collision attack on 7 rounds of Rijndael. In: AES Candidate Conference, 2000. 230-241
7. Dunkelman O, Keller N, Shamir A. Improved single-key attacks on 8-round AES-192 and AES-256. In: ASIACRYPT2010, Berlin/Heidelberg: Springer-Verlag, 2010, 6477: 158-176
8. Derbez P, Fouque P, Jean J. Improved key recovery attacks on reduced-round AES in the single-key setting. In: EUROCRYPT2013, Berlin/Heidelberg: Springer-Verlag, 2013, 7881: 371-387
9. Li L B, Jia K T, Wang X Y. Improved single-key attacks on 9-round AES-192/256. IN: FSE2014, Berlin/Heidelberg: Springer-Verlag, 2015. 127-146

10. Biryukov A, Khovratovich D. Related-key cryptanalysis of the full AES-192 and AES-256. In: ASIACRYPT, Berlin/Heidelberg: Springer-Verlag, 2009, 5912: 1-18
11. Biryukov A, Khovratovich D, Nikolić I. Distingsuiher and related-key attack on the full AES-256. In: CRYPTO, Berlin/Heidelberg: Springer-Verlag, 2009, 5677: 231-249
12. Sun B, Liu M C, Guo J, et al. New Insights on AES-Like SPN Ciphers*. In: CRYPTO2016, Berlin/Heidelberg: Springer-Verlag, 2016. 605-624
13. Grassi L, Rechberger C, Rønjom S. Subspace Trail Cryptanalysis and its Applications to AES - Extended Version. In: FSE2017, in press.
14. Gueron S, Mouha N. Simpira v2: a family of efficient permutations using the AES round function. In ASIACRYPT, Berlin/Heidelberg: Springer-Verlag, 2016, 10031: 95-125
15. Dunkelman O, Keller N, Shamir A. Minimalism in cryptography: the Even-Mansour scheme revisited. In: EUROCRYPT, Berlin/Heidelberg: Springer-Verlag, 2012, 7237: 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. Cryptology ePrint Archieve, Report 2016/244(2016), http://eprint.iacr.org/, in press.
18. Rønjom S. Invariant subspaces in Simpira. Cryptology ePrint Archive, Report 2016/248(2016), http://eprint.iacr.org/2016/248.pdf
19. Knudsen L R. DEAL - a 128-bit block cipher. Complexity, 258(2), 1988.
20. Biham E,Biryukov A, Shamir A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Cryptanalysis Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelbert, 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. 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. 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. http://eprint.iacr.org/2016/689.pdf
24. Daemen J, Rijmen V. Understanding two-round differentials in aes. In: SCN, Springer, 2006. 78-94