Hindawi
Security and Communication Networks
Volume 2018, Article ID 7453572, 11 pages
https://doi.org/10.1155/2018/7453572

Research Article

Multiple Impossible Differentials Cryptanalysis on 7-Round ARIA-192

Zi-Long Jiang and Chen-Hui Jin

Information Science and Technology Institute, Zhengzhou 450000, China

Correspondence should be addressed to Zi-Long Jiang; dracipher@126.com

Received 16 October 2017; Revised 28 December 2017; Accepted 4 January 2018; Published 14 March 2018

Academic Editor: Zhe Liu

This paper studies the security of 7-round ARIA-192 against multiple impossible differentials cryptanalysis. We propose six special 4-round impossible differentials which have the same input difference and different output difference with the maximum number of nonzero common bytes. Based on these differentials, we construct six attack trails including the maximum number of common subkey bytes. Under such circumstances, we utilize an efficient sieving process to improve the efficiency of eliminating common subkeys; therefore, both data and time complexities are reduced. Furthermore, we also present an efficient algorithm to recover the master key via guess-and-determine technique. Taking advantage of the above advances, we have obtained the best result so far for impossible differential cryptanalysis of ARIA-192, with time, data, and memory complexities being $2^{189.8}$ 7-round ARIA encryptions, $2^{116.6}$ chosen plaintexts, and $2^{139.3}$ bytes, respectively.

1. Introduction

Impossible differential attack [1] is a significant method in cryptanalysis for block ciphers. Researchers will first build one or more differentials whose probabilities are zero. Then based on these differentials, they will construct attack trails and obtain the correct subkeys by rejecting all the wrong subkeys. The second phase is called the subkey sieving phase. Actually, the subkey sieving phase is highly technical: in 2008, Lu et al. [2] introduced the early abort technique. They guessed a small quantity of subkeys and selected the useful pairs which can produce the expected difference so as to reduce time complexity. At ASIACRYPT 2014, Boura et al. [3] presented the state-test technique to reduce time complexity by decreasing the quantity of subkey bits during an attack. Li et al. [4] presented the new early abort technique which does not need to check all the remaining pairs, therefore reducing time complexity. As a powerful form of cryptanalysis, impossible differential attack is extensively used to analyze many block ciphers, such as ARIA [5] and AES [6].

In 2008, multiple impossible differentials cryptanalysis was proposed by Tsunoo et al. [7], and Lu et al. also presented the idea that multiple variants of the attack trail can be applied using the same data set [8]. After that, Boura et al. [3] and Li et al. [4, 9] also used multiple impossible differentials to attack CLEFIA, Camellia, FOX, and so on and got good results. They aimed at recovering more subkey bits and increasing the probability of the remaining pairs, thus reducing data complexity. For example, Li et al. [9] presented multiple impossible differentials attacks on FOX with better results than other cryptanalysis of FOX known so far. They constructed four impossible differentials to recover four parts of subkeys. Note that these four differentials play the same role, and the order of differentials to be used does not affect the result.

ARIA, a 128-bit substitution-permutation network block cipher, was proposed as Korean standard block cipher algorithm in 2004. After analyzing its security against liner, differential, impossible differential, and square attacks, the designers declared that ARIA has a better resistance against the above cryptanalysis than AES. Wu et al. [10] constructed a 4-round impossible differential and presented a 6-round attack on ARIA. The cryptanalytic result was further enhanced by Li et al. [11] and Shenhua and Chunyan [12], respectively. Then Du and Chen [13] proposed a 7-round impossible differential attack on ARIA-256. Xie and Chen [14] presented a 7-round impossible differential attack on ARIA-192 (however, we find

Table 1: The comparison of cryptanalytic attacks on ARIA.

Attack type	Rounds	Time	Data	Memory	Reference
Impossible differential	5	$2^{71.6}$	$2^{71.3}$	2^{72}	[12]
Impossible differential	6	2^{112}	2^{121}	2^{121}	[10]
Impossible differential	6	2^{104}	$2^{120.5}$	2^{121}	[12]
Impossible differential	7	2^{238}	2^{125}	-	[13]
Impossible differential^{mk}	7	$2^{189.8}$	$2^{116.6}$	$2^{139.3}$	This paper
Integral attack	7	$2^{225.8}$	$2^{100.6}$	-	[16]
Boomerang attack	7	2^{236}	2^{128}	2^{184}	[17]
Meet-in-the-middle	7	$2^{161.3}$	2^{96}	2^{185}	[18]
Meet-in-the-middle^{mk}	7	$2^{136.1}$	2^{115}	2^{130}	[19]
Meet-in-the-middle	8	$2^{251.6}$	2^{56}	2^{252}	[18]
Meet-in-the-middle^{mk}	8	$2^{245.9}$	2^{113}	2^{138}	[19]

mk: recover the actual master key; -: not given in the related paper.

a flaw in the steps of its cryptanalysis). Despite all these contributions, the previous studies neither recover the actual master key of ARIA nor have a research on the security against multiple impossible differentials cryptanalysis.

At EUROCRYPT 2016, Sun et al. proved that if the details of the S-boxes are not considered, the length of the impossible differential of ARIA could not be improved [15], so we would like to improve the sieving process to obtain better results. Different from preceding studies, our multiple impossible differentials cryptanalysis is expected to reduce the retention rate of wrong subkeys in subkey sieving phase, thus reducing data complexity and time complexity. We also optimize the order of attack trails (i.e., the attack trails with the maximum number of common bytes are priority). If we conclude that a current common subkey is wrong, it is unnecessary for this common subkey to be sieved by other attack trails; therefore, the efficiency of eliminating common subkeys can be improved. Based on this efficient sieving process, we propose the first multiple impossible differentials attack on 7-round ARIA-192, which improves impossible differential attack in two dimensions (i.e., data and time complexities). Table 1 is the comparison of cryptanalytic results on ARIA.

The remainder of the paper is organized as follows. Section 2 briefly describes the ARIA cipher and provides the notations adopted in this paper. Section 3 constructs the 4-round multiple impossible differentials. Section 4 presents our impossible differential attacks on 7-round ARIA-192 combined with various techniques. Section 5 concludes this paper.

2. Preliminaries

2.1. Description of ARIA. The block cipher ARIA is a 128-bit SPN model and the numbers of the round are 12/14/16 corresponding to the keys of 128/192/256 bits, respectively. The plaintext, the ciphertext, and the internal state of ARIA are treated as a 4×4 matrix, as shown in Figure 1.

Three operations are applied in every round as follows.

(1) Round Key Addition (AK). This operation includes an XOR with the round subkeys which are derived from the master key.

(2) Substitution Layer (SL). This operation, based on four types of 8-bit S-boxes S_1, S_2, S_1^{-1}, and S_2^{-1}, has two types of substitution layers SL_1 and SL_2. SL_1 is for the odd rounds, and SL_2 is for the even rounds. The specific layers are as follows.

$$\begin{aligned}\mathrm{SL}_1 &= S_1S_2S_1^{-1}S_2^{-1}S_1S_2S_1^{-1}S_2^{-1}S_1S_2S_1^{-1}S_2^{-1}S_1S_2S_1^{-1}S_2^{-1}\\ \mathrm{SL}_2 &= S_1^{-1}S_2^{-1}S_1S_2S_1^{-1}S_2^{-1}S_1S_2S_1^{-1}S_2^{-1}S_1S_2S_1^{-1}S_2^{-1}S_1S_2.\end{aligned} \tag{1}$$

(3) Diffusion Layer (DL). A linear map DL: $F_{2^8}^{16} \to F_{2^8}^{16}$ is given by $(x_0, x_1, \ldots, x_{15}) \to (y_0, y_1, \ldots, y_{15})$, where

$$\begin{aligned}
y_0 &= x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14}\\
y_1 &= x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15}\\
y_2 &= x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15}\\
y_3 &= x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14}\\
y_4 &= x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15}\\
y_5 &= x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15}\\
y_6 &= x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13}\\
y_7 &= x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13}\\
y_8 &= x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15}\\
y_9 &= x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14}\\
y_{10} &= x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15}\\
y_{11} &= x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14}\\
y_{12} &= x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12}\\
y_{13} &= x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13}\\
y_{14} &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14}\\
y_{15} &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}.
\end{aligned} \tag{2}$$

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

Figure 1: Sixteen cells with every byte numbered.

Note that Diffusion Layer is an involution and therefore $DL^{-1} = DL$. In the last round, the DL is substituted by AK to generate ciphertexts.

The key schedule algorithm can be divided into two parts, that is, Initialization and Round Key Generation. This section focuses on the description of ARIA-192. For more details, please refer to [5].

(1) Initialization. The master key is 192 bits in size which is loaded to 256 bits (KL, KR), and the remaining 64-bit space on KR is filled with zero.

Then, four 128-bit values of (W_0, W_1, W_2, W_3) are generated from (KL, KR) as follows:

$$W_0 = \mathrm{KL} \tag{3}$$

$$W_1 = F_o(W_0, \mathrm{CK}_1) \oplus \mathrm{KR} \tag{4}$$

$$W_2 = F_e(W_1, \mathrm{CK}_2) \oplus W_0 \tag{5}$$

$$W_3 = F_o(W_2, \mathrm{CK}_3) \oplus W_1, \tag{6}$$

where F_e is the even round function and F_o is the odd round function. Three 128-bit values of $(\mathrm{CK}_1, \mathrm{CK}_2, \mathrm{CK}_3)$ are constants.

(2) Round Key Generation. Eight round subkeys k_i are generated as follows $(0 \le i \le 7)$:

$$k_0 = W_0 \oplus (W_1 >>> 19) \tag{7}$$

$$k_1 = W_1 \oplus (W_2 >>> 19) \tag{8}$$

$$k_2 = W_2 \oplus (W_3 >>> 19) \tag{9}$$

$$k_3 = (W_0 >>> 19) \oplus W_3 \tag{10}$$

$$k_4 = W_0 \oplus (W_1 >>> 31) \tag{11}$$

$$k_5 = W_1 \oplus (W_2 >>> 31) \tag{12}$$

$$k_6 = W_2 \oplus (W_3 >>> 31) \tag{13}$$

$$k_7 = (W_0 >>> 31) \oplus W_3. \tag{14}$$

2.2. Notations. Some notations are given as follows:

P: plaintext

C: ciphertext

Δx: the difference of x

Table 2: Six 4-round impossible differential distinguishers of ARIA.

Number	The positions of nonzero input difference	The positions of nonzero output difference
(1)	(7, 13)	(0, 4, 13, 15)
(2)		(0, 1, 4, 13)
(3)		(0, 1, 13, 15)
(4)		(0, 1, 4, 7)
(5)		(0, 4, 7, 15)
(6)		(0, 1, 7, 15)

$k_{i,(p,\ldots,r)}$: pth, …, rth byte in the i-th round subkey

$\Delta a \not\rightarrow_{i\text{-round}} \Delta b$: Δa cannot be Δb after i-round

$x_{i,(p,\ldots,r)}^{\mathrm{AK/SL/DL}}$: the intermediate values of pth, …, rth bytes in the i-th round after the AK/SL/DL

In this paper, we denote the whitening key as k_0.

3. Four-Round Impossible Differentials of ARIA

We find six 4-round impossible differentials of ARIA with the same input difference. As shown in Figure 2, two bytes of the input difference are nonzero, and the others are zero. Four bytes of the output difference are nonzero and equal, and the others are zero. The other five differentials have the same input difference and different output difference with the maximum number of nonzero common bytes. The positions of nonzero difference are shown in Table 2.

Taking the first differential as an example, we describe its property as follows.

Property 1. There is a 4-round impossible differential of ARIA:

$$\begin{aligned}&(0,0,0,0,0,0,0,e_7,0,0,0,0,0,e_{13},0,0)\\ &\not\rightarrow_{4\text{-round}} (m,0,0,0,m,0,0,0,0,0,0,0,0,m,0,m),\end{aligned} \tag{15}$$

where $(e_7, e_{13}) \neq (0,0)$ and m is nonzero difference.

Proof. First, we analyze the first two rounds of the differential. Two nonzero difference bytes f_7, f_{13} can be obtained from the input difference through the AK and SL operations. Then calculate $\Delta x_3^{\mathrm{DL}} = (g_0, g_1, 0, g_3, 0, 0, g_6, g_7, g_8, 0, g_{10}, g_{11}, g_{12}, g_{13}, 0, 0)$, where g_i are nonzero difference bytes. Δx_3^{DL} is preserved after the AK and SL operations, and then $\Delta x_4^{\mathrm{SL}} = (h_0, h_1, 0, h_3, 0, 0, h_6, h_7, h_8, 0, h_{10}, h_{11}, h_{12}, h_{13}, 0, 0)$ is obtained. After DL operation, we can obtain $i_0 = i_{10} = h_3 \oplus h_6 \oplus h_8 \oplus h_{13}$ and $i_0 \oplus i_{10} = 0$. □

Second, we analyze the last two rounds of the differential. We can obtain $\Delta x_6^I = (0, l_1, 0, 0, 0, 0, 0, l_7, 0, l_9, 0, l_{11}, 0, 0, 0, 0)$ from the output difference after the operations DL^{-1}, SL^{-1}, and AK^{-1}. The DL^{-1} of R_5 makes $s_0 = l_9 \neq 0$, $s_{10} = 0$. Then after the operations SL^{-1}, AK^{-1}, we obtain $j_0 \neq 0$, $j_{10} = 0$ and therefore $j_0 \oplus j_{10} \neq 0$, which contradicts $i_0 \oplus i_{10} = 0$.

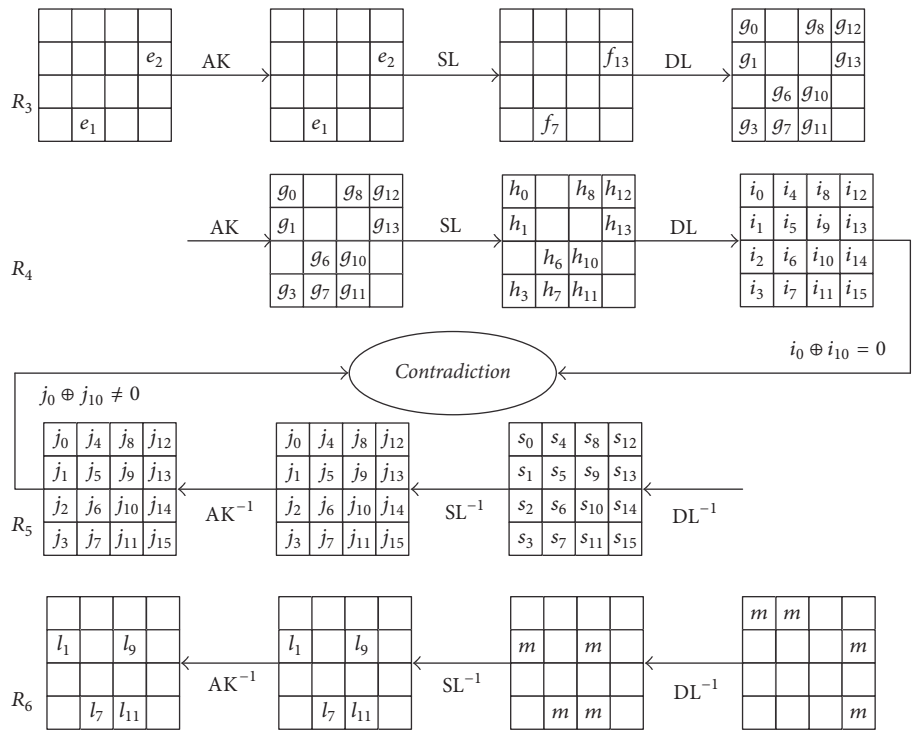

Figure 2: One of 4-round impossible differentials of ARIA.

Figure 2 underlines this contradiction, and the other five impossible differentials can also be proven in a similar way.

4. A Multiple Impossible Differentials Attack on 7-Round ARIA-192

As shown in Figure 3, two rounds at the top and one round at the bottom are added to the 4-round differentials. We first propose the multiple impossible differentials attack on 7-round ARIA-192 combined with a series of techniques.

4.1. Properties of Diffusion Layer. In this section, we first analyze the flaw in [14] and then describe the two linear properties used in this paper.

In attacking 6-byte subkey $k_{1,(0,1,7,10,11,12)}$, 6-byte values of $x^{\mathrm{DL}}_{1,(0,1,7,10,11,12)}$ are needed. From the definition of the Diffusion Layer, we know that the 6-byte values are functions of 14-byte $x^{\mathrm{SL}}_{1,(1,2,3,4,5,6,7,8,9,11,12,13,14,15)}$. However, the attack scenario in [14] did not attack 2-byte whitening keys $k_{0,(5,15)}$ and cannot obtain 2-byte values of $x^{\mathrm{DL}}_{1,(1,10)}$, so the input difference of the differential cannot be obtained and the attack does not work. Section 4 of [14] presented another attack scenario attacking 14-byte $x^{\mathrm{SL}}_{1,(1,2,3,4,5,6,7,8,9,11,12,13,14,15)}$, but the time complexity exceeded 2^{192} 7-round ARIA encryptions. To sum up, the attack scenario of ARIA-192 in [14] does not work. This paper corrects this flaw. Moreover, we present the multiple impossible differentials attack on 7-round ARIA-192.

We provide Properties 2 and 3 of Diffusion Layer. By using Property 2, if the input difference before DL in the first round satisfies eight difference equations of Property 2, we conclude that 10-byte difference $\Delta x^{\mathrm{DL}}_{1,(2,3,4,5,6,8,9,13,14,15)}$ are zero. Similarly, by using Property 3, if the 6-byte difference satisfies $d_0 = d_1 = d_7 = d_{10} = d_{11} = d_{12} = d \neq 0$ and the others are zero, we conclude that 2-byte difference $e_7 = e_{13} = e \neq 0$ and the others are zero. We describe these two properties as follows.

Property 2. Let $(b_0, b_1, \ldots, b_{15})$ and $(c_0, c_1, \ldots, c_{15})$ denote the input difference and output difference of the Diffusion Layer in the first round. Then $c_2 = c_3 = c_4 = c_5 = c_6 = c_8 = c_9 = c_{13} = c_{14} = c_{15} = 0$ if and only if the following eight equations are satisfied simultaneously.

$$b_1 = b_{11} \tag{16}$$

$$b_4 = b_{14} \tag{17}$$

$$b_5 = b_{15} \tag{18}$$

$$b_1 \oplus b_4 \oplus b_5 \oplus b_7 \oplus b_{13} = 0 \tag{19}$$

$$b_4 \oplus b_5 \oplus b_6 \oplus b_{12} = 0 \tag{20}$$

$$b_1 \oplus b_2 \oplus b_4 \oplus b_8 = 0 \tag{21}$$

$$b_2 \oplus b_7 \oplus b_9 \oplus b_{12} \oplus b_{13} = 0 \tag{22}$$

$$b_1 \oplus b_3 \oplus b_5 \oplus b_9 = 0. \tag{23}$$

Property 3. Let $(d_0, d_1, \ldots, d_{15})$ and $(e_0, e_1, \ldots, e_{15})$ denote the input difference and output difference of the Diffusion Layer in the second round. Six-byte difference satisfies $d_0 = d_1 = d_7 = d_{10} = d_{11} = d_{12} = d \neq 0$ and the others are zero;

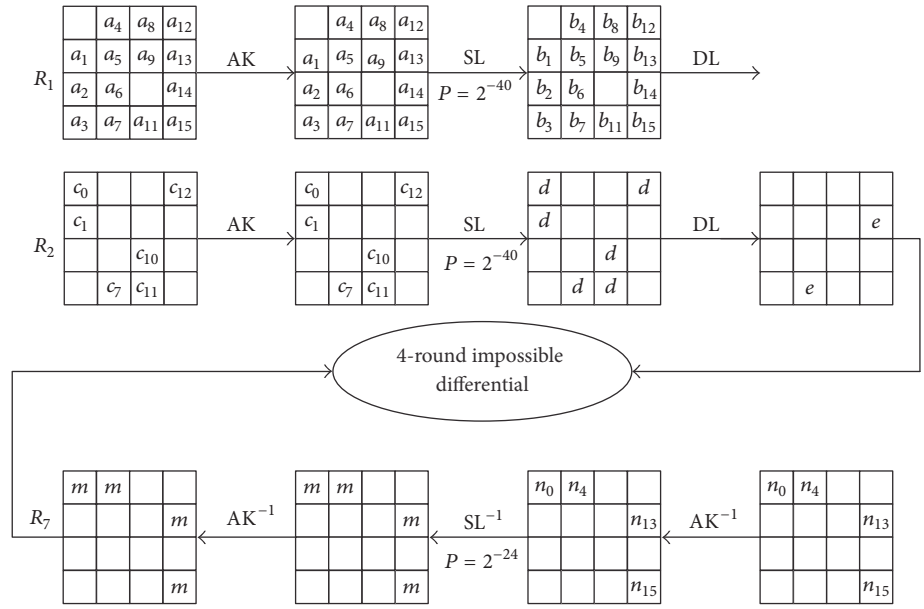

Figure 3: One of 7-round impossible differential attack trails.

we can reach that 2-byte difference satisfies $e_7 = e_{13} = e \neq 0$ and the others are zero.

See Appendix for the proof of these two properties.

4.2. An Efficient Sieving Process. In this section, we introduce an efficient sieving process. For simplicity, we abbreviate the notation in Table 3.

The idea of the efficient sieving process is summarized as below. We would like to construct some special attack trails with the maximum number of common bytes. Then the common subkeys can be repeatedly sieved by multiple attack trails. If we conclude that the current common subkey is wrong, it is unnecessary for this common subkey to be sieved by other attack trails, therefore reducing the retention rate in sieving subkeys and improving the result.

First, we find some impossible differentials which have the same input difference and different output difference with the maximum number of nonzero common bytes. In this paper, only 2-byte extra nonzero output differences are needed, and then five extra differentials can be constructed.

Second, we optimize the order of attack trails to be used (i.e., the attack trails with the maximum number of common bytes are priority). In this paper, although each attack trail discards possible values of 24 subkey bytes, the first two attack trails have 23 common subkey bytes, and only 26 subkey bytes need to be sieved in our attack scenario, which concludes six attack trails. Then these common subkeys can be sieved multiple times and the wrong subkeys will be rejected as soon as possible, therefore reducing the complexity.

In Section 4.3, we use the efficient sieving process to reduce data complexity and time complexity from steps (12) to (16) in online phase. In Section 4.6, we analyze the complexity when attackers only use one of these attack trails with the same techniques. The comparison of the two complexities indicates that this efficient sieving process is practical.

4.3. The Procedure of 7-Round Attack on ARIA-192. In this section, the procedure will be divided into two phases.

Precomputation Phase. Let S denote one of four types of 8-bit S-boxes and Δ_{in} and Δ_{out} denote the input and output difference of S-boxes. When Δ_{in} and Δ_{out} are nonzero bytes, the equation $S(x) \oplus S(x \oplus \Delta_{\text{in}}) = \Delta_{\text{out}}$ has one solution on average.

According to four types of S-boxes S_1, S_2, S_1^{-1}, and S_2^{-1}, we construct four tables Λ_i $(i = 1, 2, 3, 4)$, respectively. Then store the calculated x in Λ_i $(i = 1, 2, 3, 4)$ indexed by 2^{16} possible values of $(\Delta_{\text{in}}, \Delta_{\text{out}})$.

Online Phase. The online phase can be summarized in the following steps. Through the quick sort method [20], steps (1) and (2) select useful plaintext pairs whose ciphertext pairs meet the requirements of the structure. By using Properties 2 and 3, steps (3)–(11) select the plaintext pairs which can obtain the input difference of the distinguisher. According to six special impossible differential attack trails, steps (12)–(16) reject wrong subkeys through the efficient sieving process. Taking advantage of master key recovery algorithm, step (17) rejects wrong subkeys and recovers the master key of ARIA-192.

The specific steps are as follows:

(1) Select 2^{112} plaintexts which are fixed in 2 bytes (0, 10), and take all the values in other 14 bytes. These 2^{112} plaintexts are called a structure. We take 2^n structures and obtain $2^{n+112} \times (2^{n+112} - 1)/2 \approx 2^{n+223}$ plaintext pairs.

(2) By the quick sort method [20], we can choose the pairs whose ciphertext pairs have zero difference in

Table 3: Six 4-round impossible differential distinguishers of ARIA.

Number	The positions of nonzero output difference	Whitening key and 1-st round subkeys	7-th round subkeys
(1)	$A_1 = \{0, 4, 13, 15\}$	$k'_0 = k_{0,(1,2,3,4,5,6,7,8,9,11,12,13,14,15)}$ $k'_1 = k_{1,(0,1,7,10,11,12)}$	$k_7^{(1)} = k_{7,(0,4,13,15)}$
(2)	$A_2 = \{0, 1, 4, 13\}$		$k_7^{(2)} = k_{7,(0,1,4,13)}$
(3)	$A_3 = \{0, 1, 13, 15\}$		$k_7^{(3)} = k_{7,(0,1,13,15)}$
(4)	$A_4 = \{0, 1, 4, 7\}$		$k_7^{(4)} = k_{7,(0,1,4,7)}$
(5)	$A_5 = \{0, 4, 7, 15\}$		$k_7^{(5)} = k_{7,(0,4,7,15)}$
(6)	$A_6 = \{0, 1, 7, 15\}$		$k_7^{(6)} = k_{7,(0,1,7,15)}$

k'_0, k'_1 are 14-byte whitening keys and 6-byte subkeys in the first round during an attack, respectively; $k_7^{(i)}$ are 4-byte subkeys which need to be guessed in the 7-th round when using the i-th distinguisher.

all but the 4 bytes at A_i ($i = 1, \ldots, 6$). Then $2^{n+223} \times 2^{-96} = 2^{n+127}$ pairs remain in 2^n structures. Store the remaining plaintext pairs at 14 bytes $(1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15)$ and the corresponding ciphertext pairs at 4 bytes A_i ($i = 1, \ldots, 6$) in table Ω_i, which are indexed by the serial number of plaintext pairs (hereafter referred to as SN).

(3) Guess $k_{0,(1,11)}$ and partially encrypt plaintext pairs in Ω_1, and then select plaintext pairs whose difference $b_{(1,11)}$ satisfies (16). We store $2^{n+127} \times 2^{-8} = 2^{n+119}$ remaining $(b_{(1,11)}, x^{\mathrm{SL}}_{1,(1,11)})$ and its SN in table $T^{(1)}$ indexed by $k_{0,(1,11)}$.

(4) For current $k_{0,(1,11)}$, guess $k_{0,(4,14)}$ and partially encrypt plaintext pairs in $T^{(1)}$, and then select plaintext pairs whose difference $b_{(4,14)}$ satisfies the equation (17). We store $2^{n+119} \times 2^{-8} = 2^{n+111}$ remaining $(b_{(4,14)}, x^{\mathrm{SL}}_{1,(4,14)})$ and its SN in table $T^{(2)}$ indexed by $k_{0,(4,14)}$.

(5) For current $k_{0,(1,4,11,14)}$, guess $k_{0,(5,15)}$ and partially encrypt plaintext pairs in $T^{(2)}$, and then select plaintext pairs whose difference $b_{(5,15)}$ satisfies the equation (18). We store $2^{n+111} \times 2^{-8} = 2^{n+103}$ remaining $(b_{(5,15)}, x^{\mathrm{SL}}_{1,(5,15)})$ and its SN in table $T^{(3)}$ indexed by $k_{0,(5,15)}$.

(6) For current $k_{0,(1,4,5,11,14,15)}$, guess $k_{0,(7,13)}$ and partially encrypt plaintext pairs in $T^{(3)}$, and then select plaintext pairs whose difference $b_{(7,13)}$ satisfies (19). We store $2^{n+103} \times 2^{-8} = 2^{n+95}$ remaining $(b_{(7,13)}, x^{\mathrm{SL}}_{1,(7,13)})$ and its SN in table $T^{(4)}$ indexed by $k_{0,(7,13)}$.

(7) For current $k_{0,(1,4,5,7,11,13,14,15)}$, guess $k_{0,(6,12)}$ and partially encrypt plaintext pairs in $T^{(4)}$, and then select plaintext pairs whose difference $b_{(6,12)}$ satisfies (20). We store $2^{n+95} \times 2^{-8} = 2^{n+87}$ remaining $(b_{(6,12)}, x^{\mathrm{SL}}_{1,(6,12)})$ and its SN in table $T^{(5)}$ indexed by $k_{0,(6,12)}$.

(8) For current $k_{0,(1,4,5,6,7,11,12,13,14,15)}$, guess $k_{0,(2,8)}$ and partially encrypt plaintext pairs in $T^{(5)}$, and then select plaintext pairs whose difference $b_{(2,8)}$ satisfies (21). We store $2^{n+87} \times 2^{-8} = 2^{n+79}$ remaining $(b_{(2,8)}, x^{\mathrm{SL}}_{1,(2,8)})$ and its SN in table $T^{(6)}$ indexed by $k_{0,(2,8)}$.

(9) For current $k_{0,(1,2,4,5,6,7,8,11,12,13,14,15)}$, we have known $b_{(2)}, b_{(7)}, b_{(12)}, b_{(13)}$, and then $b_{(9)}$ can be obtained from (22). The value of $x^{\mathrm{AK}}_{1,(9)}$ can be obtained by accessing the row in table Λ_2. Compute $P_9 \oplus x^{\mathrm{AK}}_{1,(9)}$, and $k_{0,(9)}$ can be obtained. Store $2^{n+79} \times 2^{-8} = 2^{n+71}$ remaining $(b_9, x^{\mathrm{SL}}_{1,(9)})$ and its SN in table $T^{(7)}$ indexed by $k_{0,(9)}$.

(10) For current $k_{0,(1,2,4,5,6,7,8,9,11,12,13,14,15)}$, we have known $b_{(1)}, b_{(5)}, b_{(9)}$, and then $b_{(3)}$ can be obtained from (23). The value of $x^{\mathrm{AK}}_{1,(3)}$ can be obtained by accessing the row in table Λ_4. Compute $P_3 \oplus x^{\mathrm{AK}}_{1,(3)}$, and $k_{0,(3)}$ can be obtained. Store $2^{n+71} \times 2^{-8} = 2^{n+63}$ remaining $(b_3, x^{\mathrm{SL}}_{1,(3)})$ and its SN in table $T^{(8)}$ indexed by $k_{0,(3)}$.

(11) For current k'_0, $(x^I_{2,(0,1,7,10,11,12)}, \Delta x^I_{2,(0,1,7,10,11,12)})$ can be obtained by computing $\mathrm{DL}(x^{\mathrm{SL}}_{1,(1,2,3,4,5,6,7,8,9,11,12,13,14,15)})$. We choose the pairs whose 6-byte difference $\Delta x^I_{2,(0,1,7,10,11,12)}$ is all nonzero. Guess $\Delta x^{\mathrm{SL}}_{2,(0)}$ and then obtain 6-byte difference $\Delta x^{\mathrm{SL}}_{2,(0,1,7,10,11,12)}$. For each of these 6-byte differences, 6-byte values of $x^{\mathrm{AK}}_{2,(0,1,7,10,11,12)}$ can be obtained by accessing the row in corresponding table Λ_i. Compute $x^I_{2,(0,1,7,10,11,12)} \oplus x^{\mathrm{AK}}_{2,(0,1,7,10,11,12)}$ and obtain k'_1. We store $2^{n+63} \times 2^8 \div 2^{48} = 2^{n+23}$ SN in table $T^{(9)}$ indexed by 2^{48} possible values of k'_1.

(12) For current (k'_0, k'_1), guess 1-byte nonzero difference $\Delta x^I_{7,(0)}$ and then obtain 4-byte difference $\Delta x^I_{7,(0,4,13,15)}$. From the cipher pairs, we can obtain $\Delta x^{\mathrm{SL}}_{7,(0,4,13,15)}$. For each of these 4-byte differences, 4-byte values of $x^{\mathrm{SL}}_{7,(0,4,13,15)}$ can be obtained by accessing the row in corresponding table Λ_i. Compute $x^{\mathrm{SL}}_{7,(0,4,13,15)} \oplus C_{2,(0,4,13,15)}$ and obtain $k_7^{(1)}$. We store candidate subkeys $k_7^{(1)}$ in table $C^{(1)}$ indexed by $k_{7,(0,4,13)}$. For each $k_{7,(0,4,13)}$, if all $k_{7,(15)}$ are discarded, we conclude that the current $(k'_0, k'_1, k_{7,(0,4,13)})$ cannot be right, so it is discarded.

(13) For current (k'_0, k'_1) and the remaining subkeys $k_{7,(0,4,13)}$ in table $C^{(1)}$, we use the second attack trail to sieve wrong subkeys and perform the following substeps.

(13.1) For each plaintext pairs in Ω_2, use the current (k'_0, k'_1) to select useful pairs whose differences are zero in all but 2 bytes at (7, 13) after 2 rounds of ARIA encryptions (i.e., the input difference of distinguishers). Compute $\mathrm{SL}(P_i \oplus k_{0,i})$ to satisfy (16)–(23) step by step. Then compute $\mathrm{SL}(x^{\mathrm{AK}}_{2,(0,1,7,10,11,12)})$ and choose the pairs which satisfy Property 3. The quantity of the remaining pairs is $2^{n+127} \times 2^{-64} \times 2^{-40} = 2^{n+23}$.

(13.2) The procedure of sieving wrong subkeys is similar to step (12). For each $k_{7,(0,4,13)}$, if all $k_{7,(1)}$ are wrong, the current $(k'_0, k'_1, k_{7,(0,4,13)})$ cannot be right and is thus discarded. Otherwise, access the list with index $k_{7,(0,4,13)}$ in table $C^{(1)}$. If $k_{7,(0,4,13)}$ is also a candidate subkey in $C^{(1)}$, we store candidate subkeys $k_{7,(0,1,4,13,15)}$ in table $C^{(2)}$.

(14) For current (k'_0, k'_1) and the remaining subkeys $k_{7,(0,1,4,13,15)}$ in table $C^{(2)}$, we use the third attack trail to sieve wrong subkeys.

(14.1) The procedure of choosing the expected pairs of Ω_3 is similar to step (13.1).

(14.2) For each $k_7^{(3)}$ in table $C^{(2)}$, if a $k_7^{(3)}$ decrypts a ciphertext pairs to the impossible differential, then this corresponding subkey $(k'_0, k'_1, k_{7,(0,1,4,13,15)})$ is wrong and is thus discarded. If all the remaining subkeys in table $C^{(2)}$ are wrong, the current subkey (k'_0, k'_1) is wrong, and then return to step (12) and check the next (k'_0, k'_1). Otherwise, store the remaining subkeys $k_{7,(0,1,4,13,15)}$ in table $C^{(3)}$.

(15) For current (k'_0, k'_1) and the remaining subkeys $k_{7,(0,1,4,13,15)}$ in table $C^{(3)}$, we use the fourth attack trail to sieve wrong subkeys.

(15.1) The procedure of choosing expect pairs of Ω_4 is similar to step (13.1).

(15.2) For each $k_{7,(0,1,4)}$ in table $C^{(3)}$, we need to guess one-byte subkey $k_{7,(7)}$. The procedure of sieving subkeys is similar to step (14.2). If all subkeys in $C^{(3)}$ are wrong, then return to step (12) and check the next (k'_0, k'_1), or store the remaining subkeys $k_{7,(0,1,4,7,13,15)}$ in table $C^{(4)}$.

(16) For current (k'_0, k'_1) and the remaining subkeys $k_{7,(0,1,4,7,13,15)}$ in table $C^{(4)}$, we use the fifth and the sixth attack trails ($i = 5, 6$) to sieve wrong subkeys.

(16.1) The procedure of choosing the expected pairs of Ω_i is similar to step (13.1).

(16.2) For each $k_{7,(0,1,4,7,13,15)}$ in table $C^{(4)}$, the procedure of sieving wrong subkeys is similar to step (14.2). If all subkeys in $C^{(4)}$ are wrong, return to step (12) and check the next (k'_0, k'_1), or store the remaining subkeys $k_{7,(0,1,4,7,13,15)}$ in table $C^{(5)}$.

(17) Taking advantage of master key recovery algorithm, if the candidate subkey $(k'_0, k'_1, k_{7,(0,1,4,7,13,15)})$ passes the check of the algorithm, we conclude that this subkey is right and recover the master key, or return to step (12) and check the next (k'_0, k'_1).

4.4. Complexity Analysis. The complexity in Precomputation Phase can be neglected compared with the complexity in online phase. The complexities of steps (2)–(10) are shown in Table 4.

In step (11), the time complexity is $2^{112} \times 2^{n+63} \times 2^8 \times (1/16) \times (1/7) = 2^{n+176.2}$ 7-round ARIA encryptions, and the memory demands $2^{n+23} \times (n + 127)/8$ bytes.

Step (12) needs $2^{160} \times 2^{n+23} \times 2^8 \times (1/16) \times (1/7) = 2^{n+184.2}$ 7-round ARIA encryptions.

The time complexity of step (13) has the same value as step (12).

Each wrong subkey is rejected by 2^{n+23} pairs with a probability of $P_1 = (1-2^{-24})^{2^{n+23}}$. Step (14) just needs to cheek $2^{40} \times (P_1)^2$ subkeys in table $C^{(2)}$. Taking advantage of new early abort [9], the probability that a wrong subkey can pass $2^{24}d$ tests of pairs while rejecting $2^{24}(d+1)$ tests of pairs is $p_d = (1-2^{-24})^{2^{24}d} - (1-2^{-24})^{2^{24}(d+1)}$, and the mathematical expectation is $E(d) = \sum_{d=1}^{\infty} d p_d \approx \sum_{d=1}^{\infty} d(e^{-d} - e^{-(d+1)}) \approx 2^{-0.8}$. Thus, the time complexity of step (14) is $2^{160} \times 2^{40} \times (P_1)^2 \times 2^{24-0.8} \times (1/16) \times (1/7) = 2^{216.4} \times (P_1)^2$ 7-round ARIA encryptions.

Step (15) just needs to check the remaining subkeys of $C^{(3)}$, so the time complexity is $2^{160} \times 2^{48} \times (P_1)^3 \times 2^{23.2} \times (1/16) \times (1/7) = 2^{224.4} \times (P_1)^3$ 7-round ARIA encryptions.

Step (16) just needs to check the remaining subkeys of $C^{(4)}$, so the time complexity is $2^{160} \times 2^{48} \times (P_1)^4 \times 2^{23.2} \times (1/16) \times (1/7) = 2^{224.4} \times (P_1)^4$ and $2^{224.4} \times (P_1)^5$ 7-round ARIA encryptions, respectively.

The complexity of step (17) is detailed in Section 4.5

4.5. The Procedure of Recovering the Master Key for ARIA-192. In 2015, Akshima et al. [19] presented the master key recovery attacks on ARIA for the first time. Through the guess-and-determine technique, we present an efficient algorithm to recover the master key. Taking step (1) of the master key recovery algorithm as an example, if we guess 2^{128} values of W_0 as proposed in [19], the time complexity is equal to $2^{128} \times (1/7) = 2^{125.2}$ 7-round ARIA encryptions. Based on guess-and-determine technique, the time complexity can be reduced to $2^{76.5}$ 7-round ARIA encryptions in this paper. For better understanding, we first describe the idea of master key recovery algorithm, and then the specific steps are specified.

(1) Attack 2^{128} possible values for W_0:

(a) By $[W_0, k_{0,(1,2,3,4,5,6,7,8,9,11,12,13,14,15)}]$ and (7), we can obtain 112 bits W_1 whose 61 bits belong to the first eight bytes and other 51 bits belong to the last eight bytes.

Table 4: The complexity of online phase.

Step	Time complexity (7-round ARIA encryptions)	Memory complexity (bytes)
(2)	$6 \times 2^n \times 2^{112}\log_2 2^{112} \times \left(\frac{1}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+114.6}$	$6 \times 2^{n+127} \times 2 \times (4+14) \approx 2^{n+134.7}$
(3)	$2 \times 2^{n+127} \times 2^{16} \times \left(\frac{2}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+138.2}$	$2^{n+119} \times \left[4 + \frac{(n+127)}{8}\right]$
(4)	$2 \times 2^{16} \times 2^{n+119} \times 2^{16} \times \left(\frac{2}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+146.2}$	$2^{n+111} \times \left[4 + \frac{(n+127)}{8}\right]$
(5)	$2 \times 2^{32} \times 2^{n+111} \times 2^{16} \times \left(\frac{2}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+154.2}$	$2^{n+103} \times \left[4 + \frac{(n+127)}{8}\right]$
(6)	$2 \times 2^{48} \times 2^{n+103} \times 2^{16} \times \left(\frac{2}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+162.2}$	$2^{n+95} \times \left[4 + \frac{(n+127)}{8}\right]$
(7)	$2 \times 2^{64} \times 2^{n+95} \times 2^{16} \times \left(\frac{2}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+170.2}$	$2^{n+87} \times \left[4 + \frac{(n+127)}{8}\right]$
(8)	$2 \times 2^{80} \times 2^{n+87} \times 2^{16} \times \left(\frac{2}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+178.2}$	$2^{n+79} \times \left[4 + \frac{(n+127)}{8}\right]$
(9)	$2^{96} \times 2^{n+79} \times \left(\frac{1}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+168.2}$	$2^{n+71} \times \left[2 + \frac{(n+127)}{8}\right]$
(10)	$2^{104} \times 2^{n+71} \times \left(\frac{1}{16}\right) \times \left(\frac{1}{7}\right) = 2^{n+168.2}$	$2^{n+63} \times \left[2 + \frac{(n+127)}{8}\right]$

(b) From the key schedule algorithm, we obtain $\mathrm{KR}_{(8,9,10,11,12,13,14,15)} = 0$. According to (4), $W_{1,(8,9,10,11,12,13,14,15)}$ can be obtained by calculating $F_0(W_0, \mathrm{CK}_1)$.

(c) Take advantage of the common 51 bits of W_1 to reject some wrong guesses of W_0. We obtain $2^{128} \times 2^{-51} = 2^{77}$ remaining W_0 and 125 bits of W_1. Note that now we cannot obtain 61-th, 62-th, and 63-th bit of W_1 and denote 125 bits of W_1 as W_1'.

(2) Attack the 61-th, 62-th, and 63-th bits of W_1:

(a) By (8) and $[W_1', k_{1,(0,1,7,10,11,12)}]$, we obtain 48 bits of W_2.

(b) By (5) and $[W_0, W_1]$, we obtain W_2.

(c) Take advantage of the common 48 bits of W_2 to obtain $2^{80} \times 2^{-48} = 2^{32}$ remaining $[W_0, W_1, W_2]$.

(3) For each $[W_0, W_1, W_2]$,

(a) by (6) and $[W_1, W_2]$, we obtain W_3;

(b) by (14) and $[W_0, k_{7,(0,1,4,7,13,15)}]$, obtain 48 bits of W_3,

(c) taking advantage of the common 48 bits of W_3, we can reject some wrong guesses. We obtain the remaining $[W_0, W_1, W_2, W_3]$ and thus can recover the first seven subkeys $[k_0, \ldots, k_7]$ by calculating (7)–(14). Randomly select 2^{10} plaintexts. If these corresponding ciphertexts can be obtained after 7-round ARIA encryptions, we judge that this subkey is right and recover the master key, or reject this candidate subkey and check the next.

The details of step (1) are as follows.

(1.1) Guess 7-byte $W_{0,(0,1,5,6,11,12,14)}$, and $W_{1,(9)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-8} and the quantity of $W_{0,(0,1,5,6,11,12,14)}$ is 2^{48}.

(1.2) Guess 2-byte $W_0^1 = W_{0,(2\oplus3\oplus8\oplus15)}$, $W_{0,(13)}$, and $W_{1,(10)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-8} and the quantity of $[W_{0,(0,1,5,6,11,12,13,14)}, W_0^1]$ is 2^{56}.

(1.3) Guess 1-byte $W_0^2 = W_{0,(2\oplus3\oplus4\oplus7\oplus9)}$, and $W_{1,(11)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-8} and the quantity of $[W_{0,(0,1,5,6,11,12,13,14)}, W_0^1, W_0^2]$ is 2^{56}.

(1.4) Guess 2-byte $W_0^3 = W_{0,(4\oplus8\oplus10\oplus15)}$, $W_{0,(2)}$, and $W_{1,(15)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-8} and the quantity of $[W_{0,(0,1,2,5,6,11,12,13,14)}, W_0^1, W_0^2, W_0^3]$ is 2^{64}.

(1.5) Guess 2-byte $W_0^4 = W_{0,(7\oplus9)}$, $W_{0,(15)}$, and $W_{1,(12)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-8} and the quantity of $[W_{0,(0,1,2,5,6,10,11,12,13,14,15)}, W_0^1, W_0^2, W_0^3, W_0^4]$ is 2^{72} (note that $W_{0,(10)} = W_0^1 \oplus W_0^2 \oplus W_0^3 \oplus W_0^4$).

(1.6) Guess 1-byte $W_0^5 = W_{0,(4\oplus7)}$, and 5-bit of $W_{1,(8)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-5} and the quantity of $[W_{0,(0,1,2,5,6,10,11,12,13,14,15)}, W_0^1, W_0^2, W_0^3, W_0^4, W_0^5]$ is 2^{75}.

(1.7) Guess 1-byte $W_{0,(3)}$.

By $[W_{0,(0,1,2,3,5,6,10,11,12,13,14,15)}, W_0^1, W_0^2, W_0^3, W_0^4, W_0^5]$, we can obtain 2^{83} W_0. Six-bit $W_{1,(13,14)}$ can be obtained by calculating (4) and (7). So the retention rate is 2^{-6} and the quantity of W_0 is 2^{77}.

To sum up, the time complexity of steps (1.1)–(1.7) is $2^{56}, 2^{64}, 2^{64}, 2^{72}, 2^{72}, 2^{80}, 2^{80}$, and 2^{83}, respectively. So the time complexity of step (1) is $(2^{83} + 2 \times 2^{80}) \times (1/16) \times (1/7) \approx 2^{76.5}$ 7-round ARIA encryptions.

The details of step (2) are as follows.

(2.1) We first sieve the 2^{77} remaining W_0. According to $[W_{1,(0,10,11,12)}, k_{1,(0,10,11,12)}]$ and (8), we can obtain

Table 5: The comparison of two complexities of ARIA-192.

Attack type	Time	Data	Memory
Impossible differential attack with one attack trail	$2^{191.3}$	$2^{119.1}$	$2^{139.3}$
Impossible differential attack in Section 4.3	$2^{189.8}$	$2^{116.6}$	$2^{139.3}$

$W_{2,(9,14)}$. Then $W_{2,(9,14)}$ can be obtained by calculating (5). So the retention rate is 2^{-16} and the quantity of $[W_0, W_1']$ is 2^{61}.

(2.2) Guess 61-th, 62-th, and 63-th bits of W_1 and we can obtain W_1. Then other 32-bit common W_2 can be obtained, so the retention rate is 2^{-32} and the quantity of $[W_0, W_1, W_2]$ is $2^{61+3} \times 2^{-32} = 2^{32}$.

To sum up, the time complexity of step (2) is $(2^{77} + 2^{64}) \times (1/16) \times (1/7) \approx 2^{70.2}$ 7-round ARIA encryptions.

The time complexity of step (3) is 2^{32} 1-round ARIA encryptions. Consequently, for a subkey $(k_0', k_1', k_{7,(0,1,4,7,13,15)})$, our algorithm needs $2^{76.5}$ 7-round ARIA encryptions to recover the master key.

4.6. The Summary of Complexity. By the analysis in Sections 4.4 and 4.5, we take $n = 4.6$ and obtain $P_1 = 2^{-17.5}$.

In step (16) of Section 4.4, the number of the remaining subkeys is $\xi = 2^{208} \times (2^{-17.5})^6 = 2^{103}$, and this step demands $\xi \times 2^{76.5} = 2^{179.5}$ 7-round ARIA encryptions.

Consequently, our cryptanalysis demands $2^{n+112} = 2^{116.6}$ chosen plaintexts. The memory complexity is dominated by step (2), which is $2^{n+134.7} = 2^{139.3}$ bytes. The time complexity is dominated by steps (12) and (13), which is $2 \times 2^{n+184.2} = 2^{189.8}$ 7-round ARIA encryptions.

By attacking as many common subkey bytes as possible when using different attack trails, the efficient sieving process can improve the efficiency of sieving wrong subkey, so it helps to reduce the retention rate of wrong subkeys to $2^{-17.5\times6} = 2^{-105}$. Furthermore, based on one of these attack trails, we analyze the complexity with the same attack scenario, which need to take $n = 7.1$, and the time, data, and memory complexities are $2^{191.3}$, $2^{119.1}$, and $2^{189.8}$, respectively. The comparison of the two complexities is shown in Table 5.

It is known from the comparison that the efficient sieving process can increase the efficiency of sieving subkeys, so the results are improved to an extent.

5. Conclusion

In this paper, we utilize an efficient sieving process, which can be applied to multiple impossible differentials attack. The efficient sieving process can reduce the retention rate of wrong subkeys; thus, both data complexity and time complexity can be reduced. Taking advantage of a series of techniques, we present multiple impossible differentials cryptanalysis on 7-round ARIA-192 and recover the master key, with the best result so far for impossible differential cryptanalysis of ARIA-192.

Appendix

Proof of Property 2. By the definition of DL and $b_0 = b_{10} = 0$, we know that

$$\begin{aligned}
c_0 &= b_3 \oplus b_4 \oplus b_6 \oplus b_8 \oplus b_9 \oplus b_{13} \oplus b_{14}\\
c_1 &= b_2 \oplus b_5 \oplus b_7 \oplus b_8 \oplus b_9 \oplus b_{12} \oplus b_{15}\\
c_2 &= b_1 \oplus b_4 \oplus b_6 \oplus b_{11} \oplus b_{12} \oplus b_{15}\\
c_3 &= b_5 \oplus b_7 \oplus b_{11} \oplus b_{13} \oplus b_{14} \oplus b_{15}\\
c_4 &= b_2 \oplus b_5 \oplus b_8 \oplus b_{11} \oplus b_{14} \oplus b_{15}\\
c_5 &= b_1 \oplus b_3 \oplus b_4 \oplus b_9 \oplus b_{14} \oplus b_{15}\\
c_6 &= b_2 \oplus b_7 \oplus b_9 \oplus b_{12} \oplus b_{13}\\
c_7 &= b_1 \oplus b_3 \oplus b_6 \oplus b_8 \oplus b_{11} \oplus b_{12} \oplus b_{13}\\
c_8 &= b_1 \oplus b_4 \oplus b_7 \oplus b_{13} \oplus b_{15}\\
c_9 &= b_1 \oplus b_5 \oplus b_6 \oplus b_{11} \oplus b_{12} \oplus b_{14}\\
c_{10} &= b_2 \oplus b_3 \oplus b_5 \oplus b_6 \oplus b_8 \oplus b_{13} \oplus b_{15}\\
c_{11} &= b_2 \oplus b_3 \oplus b_4 \oplus b_7 \oplus b_9 \oplus b_{12} \oplus b_{14}\\
c_{12} &= b_1 \oplus b_2 \oplus b_6 \oplus b_7 \oplus b_9 \oplus b_{11} \oplus b_{12}\\
c_{13} &= b_3 \oplus b_6 \oplus b_7 \oplus b_8 \oplus b_{13}\\
c_{14} &= b_3 \oplus b_4 \oplus b_5 \oplus b_9 \oplus b_{11} \oplus b_{14}\\
c_{15} &= b_1 \oplus b_2 \oplus b_4 \oplus b_5 \oplus b_8 \oplus b_{15}
\end{aligned} \tag{A.1}$$

and then by eight equations (16)–(23), we know that

$$c_2 = c_3 = c_4 = c_5 = c_6 = c_8 = c_9 = c_{13} = c_{14} = c_{15} = 0. \tag{A.2}$$

To prove the property in the opposite direction, by the definition of DL^{-1} and $c_2 = c_3 = c_4 = c_5 = c_6 = c_8 = c_9 = c_{13} = c_{14} = c_{15} = 0$, we know that

$$\begin{aligned}
b_0 &= 0\\
b_1 &= c_7 \oplus c_{12}\\
b_2 &= c_1 \oplus c_{10} \oplus c_{11} \oplus c_{12}\\
b_3 &= c_0 \oplus c_7 \oplus c_{10} \oplus c_{11}\\
b_4 &= c_0 \oplus c_{11}\\
b_5 &= c_1 \oplus c_{10}\\
b_6 &= c_0 \oplus c_7 \oplus c_{10} \oplus c_{12}\\
b_7 &= c_1 \oplus c_{11} \oplus c_{12}\\
b_8 &= c_0 \oplus c_1 \oplus c_7 \oplus c_{10}\\
b_9 &= c_0 \oplus c_1 \oplus c_{11} \oplus c_{12}\\
b_{10} &= 0\\
b_{11} &= c_7 \oplus c_{12}
\end{aligned}$$

$$b_{12} = c_1 \oplus c_7 \oplus c_{11} \oplus c_{12}$$
$$b_{13} = c_0 \oplus c_7 \oplus c_{10}$$
$$b_{14} = c_0 \oplus c_{11}$$
$$b_{15} = c_1 \oplus c_{10}. \tag{A.3}$$

by the above equation, we can obtain eight equations (16)–(23).

To sum up, the necessary and sufficient condition of $c_2 = c_3 = c_4 = c_5 = c_6 = c_8 = c_9 = c_{13} = c_{14} = c_{15} = 0$ is that the eight equations (16)–(23) are satisfied simultaneously. □

Proof of Property 3. Based on the condition that 6-byte difference satisfies $d_0 = d_1 = d_7 = d_{10} = d_{11} = d_{12} = d \neq 0$ and the others are zero, we know from the definition of DL that

$$e_0 = 0$$
$$e_1 = 0$$
$$e_2 = 0$$
$$e_3 = 0$$
$$e_4 = 0$$
$$e_5 = 0$$
$$e_6 = 0$$
$$e_7 = d$$
$$e_8 = 0$$
$$e_9 = 0$$
$$e_{10} = 0$$
$$e_{11} = 0$$
$$e_{12} = 0$$
$$e_{13} = d$$
$$e_{14} = 0$$
$$e_{15} = 0 \tag{A.4}$$

so we can reach that 2-byte difference $e_7 = e_{13} \neq 0$ and the others are zero.

To prove the property in the opposite direction, based on the condition that $e_7 = e_{13} \neq 0$ and the others are zero, we know from the definition of DL^{-1} that

$$d_0 = e$$
$$d_1 = e$$
$$d_2 = 0$$
$$d_3 = 0$$
$$d_4 = 0$$
$$d_5 = 0$$
$$d_6 = 0$$
$$d_7 = e$$
$$d_8 = 0$$
$$d_9 = 0$$
$$d_{10} = e$$
$$d_{11} = e$$
$$d_{12} = e$$
$$d_{13} = 0$$
$$d_{14} = 0$$
$$d_{15} = 0 \tag{A.5}$$

so we can reach that 6-byte difference $d_0 = d_1 = d_7 = d_{10} = d_{11} = d_{12} \neq 0$ and the others are zero.

To sum up, the necessary and sufficient condition that 6-byte difference satisfies equation $d_0 = d_1 = d_7 = d_{10} = d_{11} = d_{12} = d \neq 0$ and the others are zero is that 2-byte difference satisfied equation $e_7 = e_{13} \neq 0$ and the others are zero. □

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Ting Cui, Lin Ding, and XinRan Li for their useful help. The work in this paper is supported by the Natural Science Foundation of China (Grants nos. 61772547, 61402523, and 61272488).

References

[1] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," in *Advances in Cryptology — EUROCRYPT '99*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 12–23, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[2] J. Lu, J. Kim, N. Keller et al., "Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1," *CT-RSA*, vol. 4964, 2008.

[3] C. Boura, M. Naya-Plasencia, and V. Suder, "Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and SIMON," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8873, pp. 179–199, 2014.

[4] X. Li, F.-W. Fu, and X. Guang, "Multiple impossible differential cryptanalysis on reduced FOX," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98A, no. 3, pp. 906–911, 2015.

[5] D. Kwon, J. Kim, S. Park et al., "New Block Cipher: ARIA," in *Information Security and Cryptology - ICISC 2003*, vol. 2971 of

Lecture Notes in Computer Science, pp. 432–445, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[6] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*, Springer Science & Business Media, 2013.

[7] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Suzaki, and T. Kawabata, "Cryptanalysis of CLEFIA using multiple impossible differentials," in *Proceedings of the 2008 International Symposium on Information Theory and its Applications, ISITA2008*, New Zealand, December 2008.

[8] J. Lu, O. Dunkelman, N. Keller, and J. Kim, "New impossible differential attacks on AES," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5365, pp. 279–293, 2008.

[9] X. Li, C.-H. Jin, and F.-W. Fu, "Improved results of impossible differential cryptanalysis on reduced FOX," *The Computer Journal*, vol. 59, no. 4, pp. 541–548, 2016.

[10] W.-L. Wu, W.-T. Zhang, and D.-G. Feng, "Impossible differential cryptanalysis of reduced-round ARIA and Camellia," *Journal of Computer Science and Technology*, vol. 22, no. 3, pp. 449–456, 2007.

[11] R. Li, B. Sun, P. Zhang et al., "New Impossible Differential Cryptanalysis of ARIA," http://eprint.iacr.org/2008/227.pdf.

[12] L. Shenhua and S. Chunyan, "Improved impossible differential cryptanalysis of ARIA," in *Proceedings of the 2nd International Conference on Information Security and Assurance, ISA 2008*, pp. 129–132, Republic of Korea, April 2008.

[13] C. Du and J. Chen, "Impossible differential cryptanalysis of ARIA reduced to 7 rounds," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6467, pp. 20–30, 2010.

[14] Z.-M. Xie and S.-Z. Chen, "Impossible differential cryptanalysis of 7-round ARIR-192," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 35, no. 10, pp. 2301–2306, 2013.

[15] B. Sun, M. Liu, J. Guo, V. Rijmen, and R. Li, "Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9665, pp. 196–213, 2016.

[16] Y. Li, W. Wu, and L. Zhang, "Integral attacks on reduced-round ARIA block cipher," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6047, pp. 19–29, 2010.

[17] E. Fleischmann, C. Forler, M. Gorski, and S. Lucks, "New boomerang attacks on ARIA," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6498, pp. 163–175, 2010.

[18] X. Tang, B. Sun, R. Li, C. Li, and J. Yin, "A meet-in-the-middle attack on reduced-round ARIA," *The Journal of Systems and Software*, vol. 84, no. 10, pp. 1685–1692, 2011.

[19] Akshima, D. Chang, M. Ghosh, A. Goel, and S. K. Sanadhya, "Improved meet-in-the-middle attacks on 7 and 8-round ARIA-192 and ARIA-256," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9462, pp. 198–217, 2015.

[20] Q. Zhang, "Plaintext pair sieve methods in impossible differential attack," *Computer Engineering*, vol. 2, p. 46, 2010.