Improved Meet-in-the-Middle Attacks on 7 and 8-Round ARIA-192 and ARIA-256

Akshima, Donghoon Chang, Mohona Ghosh(✉), Aarushi Goel, and Somitra Kumar Sanadhya

Indraprastha Institute of Information Technology, Delhi (IIIT-D), New Delhi, India
{akshima12014,donghoon,mohonag,aarushi12003,somitra}@iiitd.ac.in

Abstract. The ARIA block cipher has been established as a Korean encryption standard by Korean government since 2004. In this work, we re-evaluate the security bound of reduced round ARIA-192 and ARIA-256 against meet-in-the-middle (MITM) key recovery attacks in the single key model. We present a new 4-round distinguisher to demonstrate the best 7 & 8 round MITM attacks on ARIA-192/256. Our 7-round attack on ARIA-192 has data, time and memory complexity of 2^{113}, $2^{135.1}$ and 2^{130} respectively. For our 7-round attack on ARIA-256, the data/time/memory complexities are 2^{115}, $2^{136.1}$ and 2^{130} respectively. These attacks improve upon the previous best MITM attack on the same in all the three dimensions. Our 8-round attack on ARIA-256 requires 2^{113} cipher calls and has time and memory complexity of $2^{245.9}$ and 2^{138} respectively. This improves upon the previous best MITM attack on ARIA-256 in terms of time as well as memory complexity. Further, in our attacks, we are able to recover the actual secret key unlike the previous cryptanalytic attacks existing on ARIA-192/256. To the best of our knowledge, this is the first actual key recovery attack on ARIA so far. We apply multiset attack - a variant of meet-in-the-middle attack to achieve these results.

Keywords: Block cipher · ARIA · Key Recovery · Differential characteristic · Multiset attack

1 Introduction

The block cipher ARIA, proposed by Kwon et al. in ICISC 2003 [12], is a 128-bit block cipher that adopts substitution-permutation network (SPN) structure, similar to AES [3], and supports three key sizes -128-bit, 192-bit and 256-bit. The first version of ARIA (version 0.8) had 10/12/14 rounds for key sizes of 128/192/256 respectively and only two kinds of S-boxes were employed in its substitution layer [2,19]. Later ARIA version 0.9 was announced at ICISC 2003 in which four kinds of S-boxes were used. This was later upgraded to ARIA version 1.0 [9], the current version, which was standardized by Korean Agency for Technology and Standards (KATS) - the government standards organization of South Korea as the 128-bit block encryption algorithm (KS X 1213)

A. Biryukov and V. Goyal (Eds.): INDOCRYPT 2015, LNCS 9462, pp. 198–217, 2015.
DOI: 10.1007/978-3-319-26617-6_11

in December 2004. In this version, the number of rounds was increased to 12/14/16 and some modifications in the key scheduling algorithm were introduced. ARIA has also been adopted by several standard protocols such as IETF (RFC 5794 [11]), SSL/TLS (RFC 6209 [10]) and PKCS #11 [13].

ARIA block cipher has been subjected to reasonable cryptanalysis in the past 12 years since its advent. In [1], Biryukov et al. analyzed the first version (version 0.8) of ARIA and presented several attacks such as truncated differential cryptanalysis, dedicated linear attack, square attack etc. against reduced round variants of ARIA. In the official specification document of the standardized ARIA (version 1.0) [12], the ARIA developers analyzed the security of ARIA against many classical cryptanalyses such as differential and linear cryptanalysis, impossible and higher order differential cryptanalysis, slide attack, interpolation attack etc. and claimed that ARIA has a better resistance against these attacks as compared to AES. In [18], Wu et al. presented a 6-round impossible differential attack against ARIA which was improved in terms of attack complexities by Li et al. in [15]. In [16], Li et al. presented a 6-round integral attack on ARIA followed by Fleischmann et al. [8] who demonstrated boomerang attacks on 5 and 6 rounds of ARIA. Du et al. in [6], extended the number of rounds by one and demonstrated a 7-round impossible differential attack on ARIA-256. In [17], Tang et al., applied meet-in-the-middle (MITM) attack to break 7 and 8-rounds of ARIA-192/256. In Table 1, we summarize all the existing attacks on ARIA version 1.0.

In this work, we improve the attack complexities of the 7 and 8-round MITM attack on ARIA-192/256. Our work is inspired from the multiset attack demonstrated by Dunkelman et al. on AES in [7]. Multiset attack is a variant of meet-in-the-middle attack presented by Demirci et al. on AES in [4]. Demirci et al.'s attack involves constructing a set of functions which map one active byte in the first round to another active byte after 4-rounds of AES. This set of functions depend on 'P' parameters and can be described using a table of 2^P ordered 256-byte sequence of entries. This table is precomputed and stored, thus allowing building a 4-round distinguisher and attacking upto 8 rounds of AES. Due to structural similarities between ARIA and AES, a similar attack was applied to 7 & 8-rounds of ARIA by Tang et al. in [17]. The bottleneck of this attack is a very high memory complexity which is evident in the attacks on ARIA as well as shown in Table 1. To reduce the memory complexity of Demirci's attacks on AES, Dunkelman et al. in [7], proposed multiset attack which replaces the idea of storing 256 ordered byte sequences with 256 unordered byte sequences (with multiplicity). This reduced both memory and time complexity of MITM attack on AES by reducing the parameters to 'Q' (where, Q<P). They also introduced the novel idea of differential enumeration technique to significantly lower the number of parameters required to construct the multiset from 'Q' to 'R' (where, R<Q<P), thus further decreasing the attack complexities on AES. Derbez et al. in [5] improved Dunkelman et al.'s attack by refining the differential enumeration technique. By using rebound-like techniques [14], they showed that the number of reachable multisets are much lower than those counted in Dunkelman et al.'s

Table 1. Comparison of cryptanalytic attacks on ARIA version 1.0. The entries are arranged in terms of decreasing time complexities for each category of attacked rounds.

Rounds attacked	Attack type	Time complexity	Data complexity	Memory complexity	Reference
5	Boomerang Attack	2^{110}	2^{109}	2^{57}	[8]
	Integral Attack	$2^{76.7}$	$2^{27.5}$	$2^{27.5}$	[16]
	Impossible Differential	$2^{71.6}$	$2^{71.3}$	2^{72}	[15]
	Meet-in-the-middle	$2^{65.4}$	2^{25}	$2^{122.5}$	[17]
6	Integral Attack	$2^{172.4}$	$2^{124.4}$	$2^{124.4}$	[16]
	Meet-in-the-middle	$2^{121.5}$	2^{56}	$2^{122.5}$	[17]
	Impossible Differential	2^{112}	2^{121}	2^{121}	[18]
	Boomerang Attack	2^{108}	2^{128}	2^{56}	[8]
	Impossible Differential	$2^{104.5}$	$2^{120.5}$	2^{121}	[15]
7	Impossible Differential	2^{238}	2^{125}	2^{125}	[6]
	Boomerang Attack	2^{236}	2^{128}	2^{184}	[8]
	Meet-in-the-middle	$2^{185.3}$	2^{120}	2^{187}	[17]
	Meet-in-the-middle (ARIA-192)	$2^{135.1}$	2^{113}	2^{130}	This work, Sect. 4
	Meet-in-the-middle (ARIA-256)	$2^{136.1}$	2^{115}	2^{130}	This work, Sect. 4
8	Meet-in-the-middle (ARIA-256)	$2^{251.6}$	2^{56}	2^{252}	[17]
	Meet-in-the-middle (ARIA-256)	$2^{245.9}$	2^{113}	2^{138}	This work, Sect. 5

attack. This improvement allowed mounting of comparatively efficient attacks on AES and also enabled extension of number of rounds attacked. Though the results of this line of work are quite interesting, yet they have not been explored further. Coupled with the fact that the security of ARIA has not been analyzed much after Fleischmann et al.'s attack in Indocrypt 2010 [8], motivated us to investigate the effectiveness of multiset attack on ARIA.

In our attacks, we construct a new 4-round distinguisher for ARIA. As a result, our attacks significantly reduce the data/time/memory complexities of the previous 7-round MITM attack on ARIA-192/256 shown in [17]. Our 8-round attack also improves upon the time and memory complexities of the previous best 8-round MITM attack on ARIA-256 [17] but at the expense of increase in the data complexity. The key schedule algorithm of ARIA does not allow recovery of master key from a subkey unlike AES [3]. This is likely the reason why none of the previous attacks have shown the actual key retrieval on any ARIA variant. However, depending upon the key expansion of ARIA, recovery of specific subkeys allows extracting the actual secret key. In our 7 and 8-round attack on ARIA-192/256, we exploit this key scheduling property to demonstrate the actual secret key recovery in ARIA. To the best of our knowledge, we are the first to demonstrate actual key recovery on ARIA.

Our Contribution. The main contributions of this work are as follows:

- We present the best 7-round MITM based key recovery attack on ARIA 192/256 and 8-round attack on ARIA-256.
- We apply multiset attack to construct a new 4-round distinguisher on ARIA-192 and ARIA-256.
- Our 7-round attack on ARIA-192 has data/time/memory complexity of 2^{113}, $2^{135.1}$ and 2^{130} respectively.
- Our 7-round attack on ARIA-256 has data/time/memory complexity of 2^{115}, $2^{136.1}$ and 2^{130} respectively.
- Our 8-round attack on ARIA-192/256 has data/time/memory complexity of 2^{113}, $2^{245.6}$ and 2^{138} respectively.
- We present the first actual master key recovery on our attacks on ARIA-192/256.

Our results are summarized in Table 1.

Organization. The paper is organized as follows. In Sect. 2, we provide a brief description of ARIA followed by important notations adopted throughout the work. In Sect. 3, we give details of our distinguisher so constructed on 4-rounds of ARIA. In Sect. 4, we present our 7-round attack followed by Sect. 5, where we demonstrate our 8-round attack on ARIA and show actual key recovery. Finally in Sect. 6, we summarize and conclude our paper.

2 Preliminaries

In this section, we first describe ARIA and then mention the key notations and definitions used in our cryptanalysis technique to facilitate better understanding.

2.1 Description of ARIA

The block cipher ARIA adopts substitution-permutation network in its design and is structurally similar to Advanced Encryption Standard (AES). The ARIA

specification defines 3 key sizes - 128-bit, 192-bit and 256-bit with block size limited to a fixed 128-bit size for all the three alternatives. Each ARIA variant has different number of rounds per full encryption, i.e., 12, 14 and 16 rounds for ARIA-128, ARIA-192 and ARIA-256 respectively. The 128-bit internal state and key state are treated as a byte matrix of 4 × 4 size, where the bytes are numbered from 0 to 15 column wise (as shown in Fig. 1). Each round consists of 3 basic operations (as shown in Fig. 2):

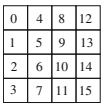

Fig. 1. Byte numbering in a state of ARIA

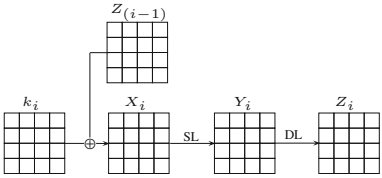

Fig. 2. i^{th} round of ARIA.

1. *Add Round Key (ARK)* - This step involves an exclusive-or operation with the round subkey. The key schedule of ARIA consists of two phases:
   - A nonlinear expansion phase, in which the 128-bit, 192-bit or 256-bit master key is expanded into four 128-bit words W_0, W_1, W_2, W_3 by using a 3-round 256-bit Feistel cipher.
   - A linear key schedule phase in which the subkeys are generated via simple XORs and rotation of W_0, W_1, W_2, W_3 each.
2. *Substitution Layer (SL)* - It uses four types of 8-bit S-boxes S_1, S_2 and their inverses S_1^{-1} and S_2^{-1}. Each S-Box is defined to be an affine transformation of the inversion function over GF(2^8). The S_1 S-box is the same as used in AES. ARIA has two types of substitution layers for even and odd rounds respectively. In each odd round, the substitution layer is (LS, LS, LS, LS) where $LS = (S_1, S_2, S_1^{-1}, S_2^{-1})$ operates one column and in each even round, the substitution layer is (LS^{-1}, LS^{-1}, LS^{-1}, LS^{-1}) where $LS^{-1} = (S_1^{-1}, S_2^{-1}, S_1, S_2)$ operates on one column as well.
3. *Diffusion Layer (DL)* - This layer consists of a 16 × 16 involutional binary matrix with branch number 8. Given an input state y and output state z, the diffusion layer is defined as:

$$
\begin{aligned}
z[0] &= y[3] \oplus y[4] \oplus y[6] \oplus y[8] \oplus y[9] \oplus y[13] \oplus y[14] \\
z[1] &= y[2] \oplus y[5] \oplus y[7] \oplus y[8] \oplus y[9] \oplus y[12] \oplus y[15] \\
z[2] &= y[1] \oplus y[4] \oplus y[6] \oplus y[10] \oplus y[11] \oplus y[12] \oplus y[15] \\
z[3] &= y[0] \oplus y[5] \oplus y[7] \oplus y[10] \oplus y[11] \oplus y[13] \oplus y[14] \\
z[4] &= y[0] \oplus y[2] \oplus y[5] \oplus y[8] \oplus y[11] \oplus y[14] \oplus y[15] \\
z[5] &= y[1] \oplus y[3] \oplus y[4] \oplus y[9] \oplus y[10] \oplus y[14] \oplus y[15] \\
z[6] &= y[0] \oplus y[2] \oplus y[7] \oplus y[9] \oplus y[10] \oplus y[12] \oplus y[13]
\end{aligned}
$$

$$
\begin{aligned}
z[7] &= y[1] \oplus y[3] \oplus y[6] \oplus y[8] \oplus y[11] \oplus y[12] \oplus y[13] \\
z[8] &= y[0] \oplus y[1] \oplus y[4] \oplus y[7] \oplus y[10] \oplus y[13] \oplus y[15] \\
z[9] &= y[0] \oplus y[1] \oplus y[5] \oplus y[6] \oplus y[11] \oplus y[12] \oplus y[14] \\
z[10] &= y[2] \oplus y[3] \oplus y[5] \oplus y[6] \oplus y[8] \oplus y[13] \oplus y[15] \\
z[11] &= y[2] \oplus y[3] \oplus y[4] \oplus y[7] \oplus y[9] \oplus y[12] \oplus y[14] \\
z[12] &= y[1] \oplus y[2] \oplus y[6] \oplus y[7] \oplus y[9] \oplus y[11] \oplus y[12] \\
z[13] &= y[0] \oplus y[3] \oplus y[6] \oplus y[7] \oplus y[8] \oplus y[10] \oplus y[13] \\
z[14] &= y[0] \oplus y[3] \oplus y[4] \oplus y[5] \oplus y[9] \oplus y[11] \oplus y[14] \\
z[15] &= y[1] \oplus y[2] \oplus y[4] \oplus y[5] \oplus y[8] \oplus y[10] \oplus y[15]
\end{aligned}
$$

In the last round, diffusion layer is replaced by key xoring to generate the ciphertext. The key schedule algorithm of ARIA [11] is divided into two phases - *Initialization phase* and *Round Key Generation phase*. In the initialization phase, for ARIA-256, first we compute KL and KR for the master key K as follows:

$$KL \mid\mid KR = K \mid\mid 0...0$$

where, $\mid KL \mid = \mid KR \mid = 128$-bits and number of zeroes padded to K equals 128, 64 and 0 for $\mid K \mid$ equal to 128, 192 and 256 respectively.

Then, four 128-bit values W_0, W_1, W_2 and W_3 are set as:

$$W_0 = KL \tag{1}$$
$$W_1 = F_o(W_0, CK_1) \oplus KR \tag{2}$$
$$W_2 = F_e(W_1, CK_2) \oplus W_0 \tag{3}$$
$$W_3 = F_o(W_2, CK_3) \oplus W_1 \tag{4}$$

where, F_o and F_e are ARIA odd and even round functions and CK_1, CK_2 and CK_3 are pre-defined constants. In the round key generation phase, the following round subkeys are generated as follows:

$$K_1 = W_0 \oplus (W_1 >>> 19) \tag{5}$$
$$K_2 = W_1 \oplus (W_2 >>> 19) \tag{6}$$
$$K_3 = W_2 \oplus (W_3 >>> 31) \tag{7}$$
$$K_4 = (W_0 >>> 19) \oplus W_3 \tag{8}$$
$$K_5 = W_0 \oplus (W_1 >>> 31) \tag{9}$$
$$K_6 = W_1 \oplus (W_2 >>> 31) \tag{10}$$
$$K_7 = W_2 \oplus (W_3 >>> 31) \tag{11}$$
$$K_8 = (W_0 >>> 31) \oplus W_3 \tag{12}$$
$$K_9 = W_0 \oplus (W1 <<< 61) \tag{13}$$

For further details, one can refer [11].

2.2 Notations and Definitions

The following notations are followed throughout the rest of the paper.

$\mathbf{P}$:	Plaintext
$\mathbf{C}$:	Ciphertext
$\mathbf{k_i}$:	Subkey of round i
$\mathbf{k_i^*}$:	$DL^{-1}(k_i)$, where, DL^{-1} is the inverse diffusion layer
$\mathbf{X_i}$:	State obtained after ARK in round i
$\mathbf{Y_i}$:	State obtained after SL in round i
$\mathbf{Z_i}$:	State obtained after DL in round i
$\mathbf{\Delta s}$:	Difference in a state s
$\mathbf{s_i[m]}$:	m^{th} byte of a state s in round i, where $0 \leq m \leq 15$
$\mathbf{s_i[p, \ldots, r]}$:	p^{th} byte, $\ldots$, r^{th} byte of state s in round i, where $0 \leq p, r \leq 15$

In our attacks, rounds are numbered from 1 to R, where R = 7 or 8. A *full* round consists of all the three round operations, i.e., ARK, SL and DL whereas a *half* round denotes a round in which the DL operation is omitted.

We utilize the following definitions for our attacks.

Definition 1 (δ-list). We define the δ-list as an ordered list of 256 16-byte distinct elements that are equal in 15 bytes. Each of the 15 equal bytes is called as passive byte whereas the one byte that takes all possible 256 values is called the active byte [3]. We denote the δ-list as (x^0, x^1, $x^2, \ldots, x^{255}$) where x^j indicates the j^{th} 128-bit member of the δ-list. As mentioned in the notations section, x_i^j [m] represents the m^{th} byte of x^j in round i.

Definition 2 (Multiset). A multiset is a set of elements in which multiple instances of the same element can appear. A multiset of 256 bytes, where each byte can take any one of the 256 possible values, can have $\binom{2^8+2^8-1}{2^8} \approx 2^{506.17}$ different values.

Two crucial properties that will be used in our attacks are as follows:

Property 1. For a given input-output difference (denoted as (ΔY, ΔZ)) state over a diffusion layer operation (as shown in Fig. 3), if the 7-bytes of ΔY [3, 4, 6, 8, 9, 13, 14] have equal differences, say y, then it will lead to non-zero difference only at byte 0 of ΔZ (instead of full state diffusion) after the diffusion layer operation. Rest all bytes of ΔZ will be passive. Thus, under the given constraints, probability of the differential trail ΔY $\rightarrow$ Δ Z is 1.

Proof. As per the diffusion layer specification of ARIA, each output byte of state Z is a xored sum of 7 input bytes of state Y. The same property is preserved in case of differences as well, i.e., each output byte difference of Z is a xored sum of 7 input byte difference of Y. In lieu of this, for each output byte, if even number

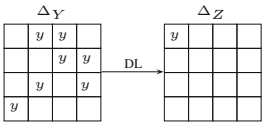

Fig. 3. Differential property of diffusion layer

of corresponding input bytes (i.e., 2, 4 or 6) have equal differences, then they cancel out each other. In the above trail, 7 bytes of Y, i.e., Y [3, 4, 6, 8, 9, 13, 14] have equal differences 'y', whereas the rest of the bytes have zero differences. Hence, all output bytes except ΔZ [0] have zero differences since their xored sum have either 2 or 4 equal input byte differences. E.g.,

$$\begin{aligned}
\Delta Z[0] &= \Delta Y[3] \oplus \Delta Y[4] \oplus \Delta Y[6] \oplus \Delta Y[8] \oplus \Delta Y[9] \oplus \Delta Y[13] \oplus \Delta Y[14] \\
&= \not{y} \oplus \not{y} \oplus \not{y} \oplus \not{y} \oplus \not{y} \oplus \not{y} \oplus y = y \\
\Delta Z[1] &= \Delta Y[2] \oplus \Delta Y[5] \oplus \Delta Y[7] \oplus \Delta Y[8] \oplus \Delta Y[9] \oplus \Delta Y[12] \oplus \Delta Y[15] \\
&= 0 \oplus 0 \oplus 0 \oplus \not{y} \oplus \not{y} \oplus 0 \oplus 0 = 0 \\
\Delta Z[11] &= \Delta Y[2] \oplus \Delta Y[3] \oplus \Delta Y[4] \oplus \Delta Y[7] \oplus \Delta Y[9] \oplus \Delta Y[12] \oplus \Delta Y[14] \\
&= 0 \oplus \not{y} \oplus \not{y} \oplus 0 \oplus \not{y} \oplus 0 \oplus \not{y} = 0
\end{aligned}$$

Similar equations can be constructed for other output bytes of Z as well. Thus, Property 1 holds true.

Property 2. For a given ARIA S-box, say S_1 and any non-zero input - output difference pair, say (Δ_i - Δ_o) in F_{256}, there exists one solution in average, say y, for which the equation, $S_1(y) \oplus S_1(y \oplus \Delta_i) = \Delta_o$, holds true (since ARIA uses AES S-box as S_1 [5]). This property is also applicable to other ARIA S-boxes, i.e., S_2, S_1^{-1} and S_2^{-1}.

The time complexity of the attack is measured in terms of number of full round (7 or 8) ARIA encryptions required. The memory complexity is measured in units of 128-bit ARIA blocks required.

3 Distinguishing Property of 4-round ARIA

Given a list of 256 distinct bytes (M^0, M^1, ..., M^{255}), a function $f : \{0,1\}^{128} \mapsto \{0,1\}^{128}$ and a 120-bit constant U, we define a multiset v as follows:

$$\begin{aligned}
C^i &= f(M^i \,||\, U), \text{where } (0 \le i \le 255) \\
v &= \{C^0[0] \oplus C^0[0], C^1[0] \oplus C^0[0], \ldots, C^{255}[0] \oplus C^0[0]\}
\end{aligned}$$

Note that, $(M^0 \,||\, U, M^1 \,||\, U, \ldots, M^{255} \,||\, U)$ forms a δ-list and atleast one element of the multiset is always zero.

Distinguishing Property. Let us consider $\mathcal{F}$ to be a family of permutations on 128-bit. Then, given any list of 256 distinct bytes $(M^0, M^1, \ldots, M^{255})$, the aim is to find how many multisets v are possible when, $f \xleftarrow{\$} \mathcal{F}$ and $U \xleftarrow{\$} \{0,1\}^{120}$.

In case, when $\mathcal{F}$ = family of all permutations on 128-bit and $f \xleftarrow{\$} \mathcal{F}$. Under such setting, since in the multiset v, we have 255 values that are chosen uniformly and independently from the set $\{0, 1, \ldots, 255\}$ (as one element, say $C^0[0] \oplus C^0[0]$, is always 0), the total possible multisets v are $\binom{2^8-1+2^8-1}{2^8-1} \approx 2^{505.17}$.

In case, when $\mathcal{F}$= 4-full rounds of ARIA and $f \xleftarrow{\$} \mathcal{F}$. Here, $f \xleftarrow{\$} \mathcal{F} \Leftrightarrow K \xleftarrow{\$} \{0,1\}^k$ and $f = E_K$, where k = 128 (for ARIA-128), 192 (for ARIA-192) or 256 (for ARIA-256). Let us consider, 4-full rounds of ARIA as shown in Fig. 4 where, multiset v is defined as $v = \{Z_4^0[0] \oplus Z_4^0[0], Z_4^1[0] \oplus Z_4^0[0], \ldots, Z_4^{255}[0] \oplus Z_4^0[0]\}$. Then, we state the following *Observation 1.*

Observation 1. The multiset v is determined by the following 30 single byte parameters only:

- $X_2^0[3, 4, 6, 8, 9, 13, 14]$ (7-bytes)
- $X_3^0[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$ (full 16-byte state)
- $X_4^0[3, 4, 6, 8, 9, 13, 14]$ (7-bytes)

Thus, the total number of multisets possible is $2^{30\times 8} = \mathbf{2^{240}}$ since, each 30-bytes defines one multiset.

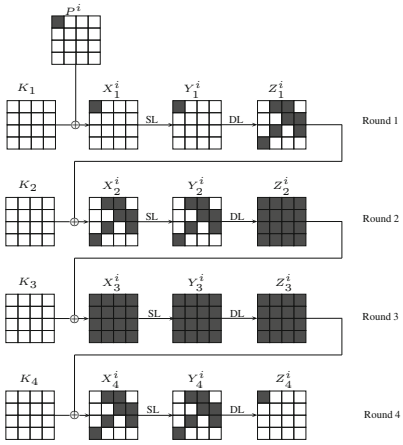

Fig. 4. 4-Round distinguisher in ARIA. Here, P^i denotes $(M^i \,||\, U)$ and X_j^i, Y_j^i, Z_j^i denote intermediate states corresponding to P^i in round j. The round subkeys K_i, where, $1 \leq i \leq 4$ are generated from the master key K.

Proof. In round 1, the set of differences $\{X_1^0[0] \oplus X_1^0[0], X_1^1[0] \oplus X_1^0[0], \ldots, X_1^{255}[0] \oplus X_1^0[0]\}$ (or, equivalently, set of differences at $X_1[0]$) are known as there are exactly 256 differences possible. Since S-box S_1 is injective, exactly 256 values exist in the set $\{Y_1^0[0] \oplus Y_1^0[0], Y_1^1[0] \oplus Y_1^0[0], \ldots\ldots, Y_1^{255}[0] \oplus Y_1^0[0]\}$ as well. Due to DL and ARK operations being linear, the set of differences at X_2[3, 4, 6, 8, 9, 13, 14] are known (according to diffusion layer (DL) definition discussed in Sect. 2). Owing to the non-linearity of the substitution layer, the set of differences at Y_2[3, 4, 6, 8, 9, 13, 14] cannot be known and one cannot move forward. To alleviate this problem, it is sufficient to know X_2^0[3, 4, 6, 8, 9, 13, 14], i.e., values of the active bytes of the first state (out of 256 states) at X_2 as it enables calculating the active bytes of the other X_2^i states (where, $1 \leq i \leq 255$) and cross SL in round 2. Again, since DL and ARK operations are linear, the set of differences $\{X_3^0 \oplus X_3^0, X_3^1 \oplus X_3^0, \ldots, X_3^{255} \oplus X_3^0\}$ is known. In order to know the set of values $\{X_3^0, X_3^1, \ldots, X_3^{255}\}$ for crossing the SL in round 3, it is sufficient to know the value of the full state X_3^0 which is given as a parameter.

By similar logic, as explained above, the set of differences $\{X_4^0 \oplus X_4^0, X_4^1 \oplus X_4^0, \ldots, X_4^{255} \oplus X_4^0\}$ are known. Now, at this stage, if only X_4^0[3, 4, 6, 8, 9, 13, 14] bytes are known, the SL layer in round 4 can be crossed and the set of 256 values $\{Z_4^0[0], Z_4^1[0], \ldots, Z_4^{255}[0]\}$ at Z_4 can be computed. Then the value of multiset $v = \{Z_4^0[0] \oplus Z_4^0[0], Z_4^1[0] \oplus Z_4^0[0], \ldots, Z_4^{255}[0] \oplus Z_4^0[0]\}$ can be determined easily as well. This shows that the multiset v depends on 30 parameters and can take 2^{240} possible values. □

Since, there are 2^{240} possible multisets at $Z_4[0]$, if we precompute and store these values in a hash table, then the precomputation complexity goes higher than brute force for ARIA-192. In order to reduce the number of multisets, we apply the Differential Enumeration technique suggested by Dunkelman et al. in [7] and improved by Derbez et al. in [5]. We call the improved version proposed in [5] as *Refined Differential Enumeration.*

Refined Differential Enumeration. The basic idea behind this technique is to choose a list of 256 distinct bytes (M^0, M^1, …, M^{255}) such that several of the parameters that are required to construct the multiset equal some predetermined constants.

To achieve so, let us construct a truncated differential for four full rounds of ARIA, in which the input and output differences are non-zero at byte 0 only (as shown in Fig. 5).

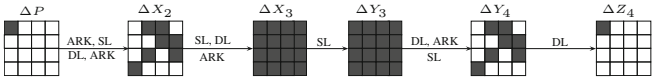

Fig. 5. 4-Round truncated differential in ARIA

The probability of this trail is 2^{-120} as follows: the one byte difference at $\Delta P[0]$ propagates to 7-byte difference in ΔX_2 and 16-byte difference in ΔY_3 with proba-

bility 1. Next, the probability that full state difference in ΔY_3 leads to 7-byte difference in ΔY_4 is 2^{-72} (since 9 bytes of ΔY_4, i.e., $\Delta Y_4[0, 1, 2, 5, 7, 10, 11, 12, 15]$) have zero difference). Further, the probability that random differences in ΔY_3 yield equal differences in the active bytes of ΔY_4 i.e., $\Delta Y_4[3, 4, 6, 8, 9, 12, 13]$ is 2^{-48}.[1] Therefore, the total probability of $\Delta Y_3 \rightarrow \Delta Y_4$ is $2^{-(72+48)} = 2^{-120}$. Then, by the virtue of *Property 1* (mentioned in Sect. 2), 7-byte difference in ΔY_4 yields a single byte difference in $\Delta Z_4[0]$ with probability 1. Thus, the overall probability of the differential from $\Delta P \rightarrow \Delta Z_4$ is 2^{-120}.

In other words, we require 2^{120} plaintext pairs to get a right pair. Once, we get a right pair, say (P^0, P^1), we state the following *Observation 2*:

Observation 2. Given a right pair (P^0, P^1) that follows the truncated differential trail shown in Fig. 5, then the 30 parameters corresponding to P^0 mentioned in *Observation 1* can take one of at most 2^{128} fixed 30-byte values (out of the total 2^{240} possible values) where, each of these 2^{128} 30-byte values are defined by each of the 2^{128} values of the 16 following parameters:

- $\Delta Y_1[0]$
- $X_2^0[3, 4, 6, 8, 9, 13, 14]$
- $Y_4^0[3, 4, 6, 8, 9, 13, 14]$
- $\Delta Z_4[0]$

Proof. Given a right pair (P^0, P^1), the knowledge of these 16 new parameters allows us to compute all the differences shown in Fig. 4. This is so because, knowledge of $\Delta Y_1[0]$ allows computation of $\Delta Z_1[3, 4, 6, 8, 9, 13, 14]$ and $\Delta X_2[3, 4, 6, 8, 9, 13, 14]$. Then, if the values of $X_2^0[3, 4, 6, 8, 9, 13, 14]$ are known, one can compute the corresponding $X_2^1[3, 4, 6, 8, 9, 13, 14]$, cross the SL layer in round 2 and calculate the full state difference ΔX_3. Similarly, from the bottom side, knowledge of $\Delta Z_4[0]$ allows computation of $\Delta Y_4[3, 4, 6, 8, 9, 13, 14]$. Then, if the values of $Y_4^0[3, 4, 6, 8, 9, 13, 14]$ are known, one can easily determine $Y_4^1[3, 4, 6, 8, 9, 13, 14]$, compute the corresponding $X_4^0[3, 4, 6, 8, 9, 13, 14]$ and $X_4^1[3, 4, 6, 8, 9, 13, 14]$ respectively and subsequently full state ΔY_3. Then, using the differential property of ARIA S-boxes (*property 2* mentioned in Sect. 2), the possible values of X_3^0 and X_3^1 can be computed. □

Thus, the knowledge of these 16 bytes given in *Observation 2* allows computation of the corresponding 30 parameters described in *Observation 1*. Hence, total possible values of these 30 single byte parameters are at most $2^{16\times 8} = 2^{128}$. Moreover, since these computations do not require the knowledge of key bytes, they can be easily precomputed.

Using *Observations 1* and *2*, we state the following third *Observation 3*:

Observation 3. Given $(M^0, M^1, \ldots, M^{255})$ and $f \xleftarrow{\$} \mathcal{F}$ and $U \xleftarrow{\$} \{0,1\}^{120}$, such that $M^0 \,||\, U$ and $M^j \,||\, U$, (where, $j \in \{0, 1, \ldots, 255\}$) is a right pair

[1] Random differences in 16-bytes of ΔY_3 yield random differences in the 7 active bytes of ΔX_4 which in turn lead to random differences in the active bytes of ΔY_4. The probability that these random differences in the 7-bytes of ΔY_4 are equal is 2^{-48}.

that follows differential trail shown in Fig. 5, then at most 2^{128} multisets v are possible at $Z_4[0]$.

Proof. From *Observation 1*, we know that each 30-byte parameter defines one multiset and *Observation 2* restricts the possible values of these 30-byte parameters to 2^{128}. Thus, at most 2^{128} multisets are only possible for ARIA. □

As the number of multisets in case of 128-bit random permutation ($= 2^{505.17}$) is much higher than 4-round ARIA ($= 2^{128}$), a valid distinguisher is constructed.

4 Key Recovery Attack on 7-Round ARIA-192/256

In this section, we use our *Observation 3* to launch a meet-in-the-middle attack on 7-round ARIA-192/256 to recover the key. The distinguisher is placed from round 2 to round 5, i.e., δ-list is constructed in state X_2 with byte 0 being the active byte and multiset is checked in $Z_5[0]$ (as shown in Fig. 6). One round at the top and two rounds at the bottom are added to the 4-round distinguisher. The attack consists of the following two phases:

Precomputation Phase. Compute and store the 2^{128} possible multisets at $\Delta Z_5[0]$ in a hash table based on *Observation 2*.

Online Phase. If we extend the differential trail (shown in Fig. 5) by one round backwards, such that 7-bytes (3, 4, 6, 8, 9, 13 and 14) are active in the plaintext, then with a probability of 2^{-48}, these 7 active bytes will induce a non-zero difference of one byte in $X_2[0]$. Thus, we require $2^{120+48} = 2^{168}$ plaintext pairs to start our online phase. For each of these pairs, we will guess the subkey candidates for which the pair becomes a right pair and construct the corresponding δ-list. The steps of the online phase are:

1. Encrypt 2^{57} structures of 2^{56} plaintexts each, where bytes 3, 4, 6, 8, 9, 13 and 14 take all possible values and rest of the bytes are constants.[2]
2. For each structure, store the ciphertexts in a hash table and look for pairs in which the difference in bytes 0, 1, 2, 5, 7, 10, 11, 12, 15 of the ciphertext is zero. Out of the total 2^{168} pairs, only 2^{96} pairs are expected to remain.
3. For each of the remaining 2^{96} plaintext pairs do the following:
   (a) Guess 7 bytes of $K_8[3, 4, 6, 8, 9, 13, 14]$ and check whether ΔY_6 has non zero difference only in byte 0 or not. Out of the 2^{56} possible values for K_8, only 2^8 key guesses are expected to remain (since with probability 2^{-48}, each will yield equal differences in the active bytes of ΔZ_6). Since we are only interested in checking the difference at $\Delta Y_6[0]$, $K_7[0]$ is not required to be guessed at this stage.
   (b) Guess 7 bytes of $K_1[3, 4, 6, 8, 9, 13, 14]$ and check whether ΔZ_1 has non zero difference only in byte 0 or not. Out of the 2^{56} possible values for K_1, only 2^8 key guesses are expected to remain.
   (c) For each of the $2^8 \times 2^8 = 2^{16}$ remaining guesses of 14 active bytes of K_1 and K_8:

[2] One structure has $2^{56} \times 2^{55} = 2^{111}$ plaintext pairs. Therefore, 2^{57} structures have $2^{57+111} = 2^{168}$ plaintext pairs.

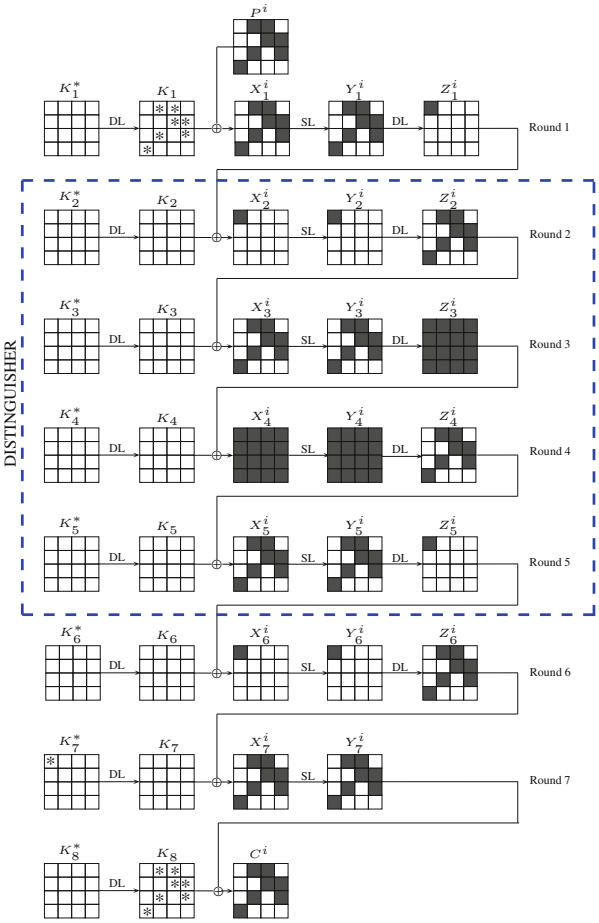

Fig. 6. 7-round attack on ARIA-192/256. The subkey bytes derived are star marked.

- Take one of the members of the pair and find its δ-list at $Z_1[0]$ using the knowledge of 7 active bytes of K_1.[3]
- Get the corresponding ciphertexts of the resulting plaintext set of the δ-list from the hash table. Guess byte $K_7^*[0] = DL^{-1}(K_7[0]) = K_7[3] \oplus K_7[4] \oplus K_7[6] \oplus K_7[8] \oplus K_7[9] \oplus K_7[13] \oplus K_7[14]$ and using the knowledge of $K_8[3, 4, 6, 8, 9, 13, 14]$, partially decrypt the ciphertexts of the δ-list to obtain the multiset at $\Delta Z_5[0]$ (which is same as that constructed in $\Delta X_6[0]$).
- Check whether this multiset exists in the precomputed table or not. If not, then discard the corresponding key guess.

[3] Encrypt the chosen right pair message to one full round using $k_1[3, 4, 6, 8, 9, 13, 14]$ and compute $Z_1[0]$. Xor other $Z_1[0]$ byte with 255 other values and decrypt them back to obtain the other plaintexts.

The probability for a wrong guess to pass the test is $2^{128} \times 2^{-467.6} = 2^{-339.6}$.[4] Since we try only $2^{96+16+8} = 2^{120}$ multisets, only the right subkey should verify the test with a probability close to 1.

Complexities. The time complexity of the precomputation phase is $2^{128} \times 2^8 \times 2^{-1.9} = 2^{134.1}$.[5] ARIA encryptions. The time complexity of the online phase is dominated by step 3(c) which is $2^{96} \times 2^{16} \times 2^8 \times 2^8 \times 2^{-2.9} = 2^{125.1}$ ARIA encryptions. Clearly the time complexity of this attack is dominated by the precomputation phase. It was shown in [5] that each 256-byte multiset requires 512-bit space. Hence, the memory complexity of the attack is $2^{128} \times 2^2 = 2^{130}$ 128-bit ARIA Blocks. The data complexity of the attack is 2^{113} plaintexts.

4.1 Recovering the Actual Master Key for 7-Round ARIA-192

In the above attack, 7-bytes of subkeys K_1 and 7-bytes of K_8 as well as 1 byte of K_7^* were recovered. In order to recover the master key do the following:

1. Guess 16-bytes of W_0.
   (a) Using the guessed value of W_0 and 7-bytes of K_1 recovered in the attack, we can deduce 56-bit of W_1 from Eq. 5. It is observed that 16-bit of this 56-bit of W_1 deduced, are part of 11^{th}, 12^{th} and 13^{th} bytes and rest 40-bits are part of first 8 bytes.
   (b) Calculate $F_o(W_0, CK_1)$. We already know that for ARIA-192, KR [8, 9,...,15] = 0. Thus, $W_1[8, 9, ..., 15]$ equals corresponding bytes of $F_o(W_0, CK_1)$ following from Eq. 2.
   (c) Discard the guesses of W_0 for which the common 16-bit of W_1 computed in (a) and (b) do not match. 2^{112} guesses of W_0 are expected to remain.
2. For each of the remaining guesses of W_0, guess 24-bits of $W_1[0, 1, ...7]$ other than the 40-bits deduced in 1(a) to know the 2^{24} possible values of W_1 corresponding to each of W_0.
3. For each remaining guesses of W_0 and corresponding guesses of W_1, deduce W_2 and W_3 from Eqs. 3 and 4.
   (a) Following Eq. 12, deduce K_8 and compare its bytes 3, 4, 6, 8, 9, 13 and 14 with the values of the same 7-bytes of K_8 recovered from the attack. Discard the guesses of W_0 and W_1 in case of mismatch of these 7-bytes of K_8. Repeat the same process for 1-byte of K_7^*. This is a 8-byte and 64-bit filtering. Out of 2^{136}, 2^{72} guesses of W_0 and W_1 are expected to remain which can be tested by brute force to obtain the correct master key.

The time complexity of the recovering process of step 3 is maximum. It is equal to $2^{136} \times (2/7) = 2^{134.2}$ 7-round ARIA encryptions as we need to compute

[4] Note that the probability of randomly having a match is $2^{-467.6}$ and not $2^{-505.17}$ since the number of ordered sequences associated with a multiset is not constant [7].

[5] The normalization factor of $2^{-1.9}$ is calculated by calculating the ratio of number of S-Box operations required in the precomputation phase to the total number of S-Box operations performed in 7-Round ARIA encryption. Similarly all other normalization factors have been calculated.

2 rounds of ARIA to deduce W_2 and W_3 and all other operations have negligible complexity as they are simple linear operations.

Therefore, the final time complexity of the attack is $2^{134.2} + 2^{134} = 2^{135.1}$. Other complexities remain the same.

4.2 Recovering the Actual Master Key for 7-Round ARIA-256

In the above attack, 7-byte of subkey K_1 and 7-byte of subkey K_8 as well as 1 byte of K_7^* were recovered. As shown in Fig. 6, we have obtained a trail such that 1^{st} byte is active at X_2. In order to recover all 16-bytes of subkey K_1, we can repeat the attack 4 times by modifying the trail such that we get a different byte active at X_2:

- bytes 3,4,6,8,9,13,14 to obtain byte 0 active at X_2
- bytes 2,5,7,8,9,12,15 to obtain byte 1 active at X_2
- bytes 1,4,6,10,11,12,15 to obtain byte 2 active at X_2
- bytes 0,5,7,10,11,13,14 to obtain byte 3 active at X_2

The time and data complexity of the attack will become 4 times of the time and data complexties mentioned in the 7-round attack in Sect. 4 respectively. Then we do the following to recover the master key:

1. Guess 16-bytes of W_0
2. For each guess of W_0, using the value of K_1 recovered from the attack, we obtain W_1 from Eq. 2. Then we follow the step 3 as mentioned in Sect. 4.1.

The time complexity of recovering the master key is $2^{128} \times (2/7) = 2^{126.2}$ 7-round ARIA encryptions.

Therefore, the final time complexity of the attack is $(4 \times 2^{134}) + 2^{126.2} = 2^{136}$. The data complexity of the attack becomes 2^{115} while the memory complexity remains same.

5 Key Recovery Attack on 8-Round ARIA-256

In this section, we describe our meet-in-the-middle attack on 8-round ARIA-256.

5.1 Construction of 4.5-Round Distinguisher

For the 8-round attack, the distinguisher constructed in Fig. 4 is extended by half round forwards upto Y_5 (DL operation is omitted). The distinguisher for 8-round attack is shown in Appendix A. Similar to *Observation 1*, we state the following *Observation 4*:

Observation 4. Given $(M^0, M^1, \ldots, M^{255})$ and $f \xleftarrow{\$} \mathcal{F}$ and $U \xleftarrow{\$} \{0,1\}^{120}$, where, f represents 4.5 rounds of ARIA, the multiset $v = \{Y_5^0[0] \oplus Y_5^0[0], Y_5^0[0] \oplus Y_5^1[0], \ldots\ldots, Y_5^0[0] \oplus Y_5^{255}[0]\}$ is determined by the following 31 1-byte parameters:

- $X_2^0[3, 4, 6, 8, 9, 13, 14]$
- $X_3^0[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$ (full 16-byte state)
- $X_4^0[3, 4, 6, 8, 9, 13, 14]$
- $X_5^0[0]$

The number of possible multisets is $2^{31\times 8} = 2^{248}$. The proof for this is similar to that described for *Observation 1* in Sect. 3.

Number of Admissible Multisets. The differential trail shown in Fig. 5 can be extended 0.5 round forwards to ΔY_5 in which only byte 0 is active with probability 1, i.e., the probability of differential trail: $\Delta P \rightarrow \Delta Y_5$ remains 2^{-120}. Then, similar to *Observation 2*, we state the following *Observation 5*.

Observation 5. Given a right pair (P^0, P^1) that follows the truncated differential trail $(\Delta P \rightarrow \Delta Y_5)$, then the 31 parameters corresponding to P^0 mentioned in *Observation 4* can take one of at most 2^{136} fixed 31-byte values (out of the total 2^{248} possible values) where, each of these 2^{136} 31-byte values are defined by each of the 2^{136} values of the 17 following parameters:

- $\Delta Y_1[0]$
- $X_2^0[3, 4, 6, 8, 9, 13, 14]$
- $Y_4^0[3, 4, 6, 8, 9, 13, 14]$
- $\Delta Z_4[0]$
- $X_5^0[0]$

The proof of this *Observation* is similar to the proof of *Observation 2* described in Sect. 3. From, *Observations 4 and 5*, we can say that the total number of admissible multisets is $2^{17\times 8} = \mathbf{2^{136}}$.

5.2 Key Recovery Attack

In this section, we discuss our 8-round attack. The distinguisher is placed from round 2 to round 5.5, i.e., δ-list is constructed in state X_2 with byte 0 being the active byte and multiset is checked in $Y_6[0]$ (as shown in Fig. 7). One round at the top and three rounds at the bottom are added to the 4.5-round distinguisher. The attack consists of the following two phases:

Precomputation Phase. Compute and store the 2^{136} possible multisets at $\Delta Y_6[0]$ in a hash table based on *Observation 5*.

Online Phase. The steps of the online phase are:

1. Encrypt 2^{57} structures of 2^{56} plaintexts each, where bytes 3, 4, 6, 8, 9, 13 and 14 take all possible values and rest of the bytes are constants. Store the ciphertexts in a hash table.
2. For each of the 2^{168} plaintext pairs do the following:
   (a) For each 2^8 guesses of ΔZ_1 [0], resolve input-output differences at SL layer of round 1 (using *Property 2*) and deduce the corresponding value of $K_1[3, 4, 6, 8, 9, 13, 14]$.

(b) For each $2^8 \times 2^{56} = 2^{64}$ guesses of $\Delta Y_6[0]$ and ΔY_7 [3, 4, 6, 8, 9, 13, 14], resolve input-output differences at SL layers in round 7 and round 8 respectively and deduce corresponding K_8^* [3, 4, 6, 8, 9, 13, 14] and full subkey K_9.

(c) For each of the $2^{64+8} = 2^{72}$ guesses of 30 bytes of K_1, K_8^* and K_9:
- Take one of the members of the pair and find its δ-list using the knowledge of 7 active bytes of K_1.
- Get the corresponding ciphertexts of the resulting plaintext set of the δ-list from the hash table. Using the knowledge of K_9 and K_8^* [3, 4, 6,

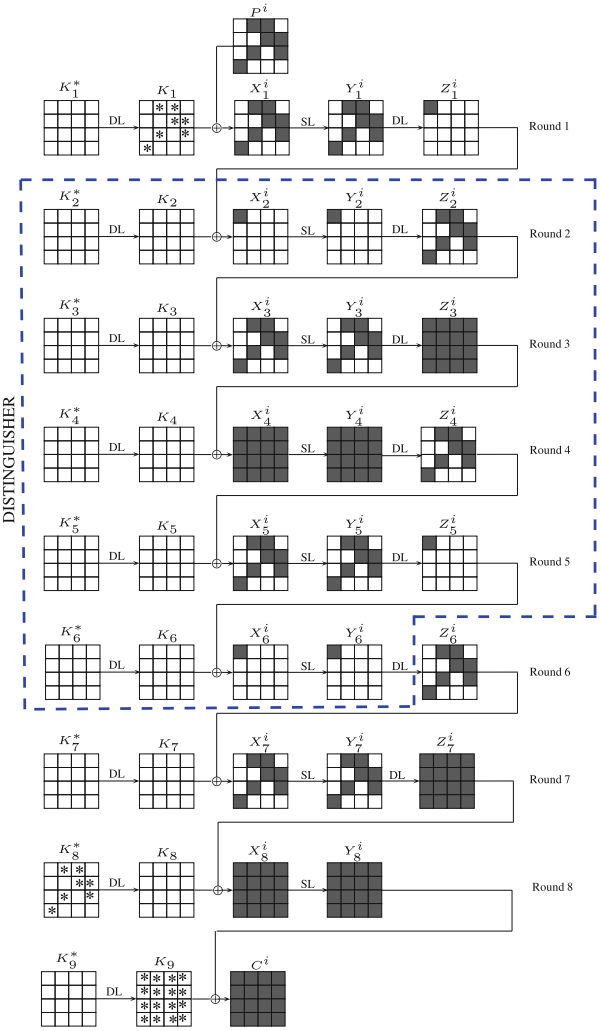

Fig. 7. 8-round attack on ARIA-256. The subkey bytes derived are star marked.

8, 9, 13, 14], partially decrypt the ciphertexts of the δ-list to compute the multiset at $\Delta Y_6[0]$.
- Check whether this multiset exists in the precomputed table or not. If not, then discard the corresponding key guess.

The probability for a wrong guess to pass the test is $2^{136} \times 2^{-467.6} = 2^{-331.6}$. Since, we try only $2^{168+72} = 2^{240}$ multisets, only the right subkey should verify the test with a probability close to 1.

Complexities. The time complexity of the precomputation phase is $2^{136} \times 2^8 \times 2^{-2} = 2^{142}$ ARIA encryptions. The time complexity of the online phase is dominated by step 2(c) which is $2^{168} \times 2^{72} \times 2^8 \times 2^{-2.1} = 2^{245.9}$ ARIA encryptions. Clearly the time complexity of this attack is dominated by the online phase. The memory complexity of the attack is $2^{136} \times 2^2 = 2^{138}$ 128-bit ARIA Blocks. The data complexity of the attack is 2^{113} plaintexts.

5.3 Recovering the Actual Master Key

In the above attack, 7-bytes of subkeys k_1 and k_8 as well as full subkey k_9 were recovered. Once these bytes are known, the remaining bytes in k_1 and k_8 can be found by exhaustive search without affecting the overall complexity of the 8-round attack. When full subkeys k_1 and k_9 are known then the master key K can be recovered as follows. Since, Eqs. 5 and 6 are two equations in two variables, they can be solved through standard matrix method by constructing a (256×256) binary matrix. We found the rank of this matrix to be 240 suggesting 2^{16} solutions for the tuple (W_0 and W_1). Once, values of W_0 and W_1 are known, KL and KR can be obtained through Eqs. 1 and 2 respectively. Thus, we get 2^{16} solutions for the master key K. Then through brute-force, the original key can be easily recovered.

6 Conclusions

In this work, we explore the space of multiset attacks as applied to key recovery attack on ARIA-192 and ARIA-256. We improve the previous 7-round and 8-round attacks on these structures and show the best attacks on them. We achieve these results by constructing a new 4-round distinguisher on ARIA and applying MITM attacks on the rest of the rounds. We also show recovery of the actual master key through our 8-round attack on ARIA-256. To our best knowledge, this is the first attempt in this direction. Currently, the number of attacked rounds remains 8 and it would be an interesting problem to try applying multiset attacks to break more rounds of ARIA.

References

1. Biryukov, A., De Canniere, C., Lano, J., Ors, S.B., Preneel, B.: Security and performance analysis of ARIA, version 1.2. Technical report, Katholieke Universiteit Leuven, Belgium (2004). http://www.cosic.esat.kuleuven.be/publications/article-500.pdf
2. De Cannière, C.: Analysis and Design of Symmetric Encryption Algorithms. PhD thesis, Katholieke Universiteit Leuven, Belgium, May 2007
3. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer, Heidelberg (2002)
4. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round AES. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 116–126. Springer, Heidelberg (2008)
5. Derbez, P., Fouque, P.-A., Jean, J.: Improved key recovery attacks on reduced-round `AES` in the single-key setting. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 371–387. Springer, Heidelberg (2013)
6. Du, C., Chen, J.: Impossible differential cryptanalysis of ARIA reduced to 7 Rounds. In: Heng, S.-H., Wright, R.N., Goi, B.-M. (eds.) CANS 2010. LNCS, vol. 6467, pp. 20–30. Springer, Heidelberg (2010)
7. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round AES-192 and AES-256. J. Cryptology **28**(3), 397–422 (2015)
8. Fleischmann, E., Forler, C., Gorski, M., Lucks, S.: New boomerang attacks on ARIA. In: Gong, G., Gupta, K.C. (eds.) INDOCRYPT 2010. LNCS, vol. 6498, pp. 163–175. Springer, Heidelberg (2010)
9. Korean Agency for Technology and Standards. 128 bit block encryption algorithm ARIA - Part 1: General (in Korean). KS X 1213-1:2009, December 2009
10. Kim, W.-H., Lee, J., Park, J.-H., Kwon, D.: Addition of the ARIA Cipher Suites to Transport Layer Security (TLS). RFC 6209, April 2011. https://tools.ietf.org/html/rfc6209
11. Kwon, D., Kim, J., Lee, J., Lee, J., Kim, C.: A Description of the ARIA Encryption Algorithm. RFC 5794, March 2010. https://tools.ietf.org/html/rfc5794
12. Kwon, D., et al.: New block cipher: ARIA. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971. Springer, Heidelberg (2004)
13. RSA Laboratories. Additional PKCS #11 Mechanisms. PKCS #11 v2.20 Amendment 3 Revision 1, January 2007
14. Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schläffer, M.: Rebound distinguishers: results on the full whirlpool compression function. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 126–143. Springer, Heidelberg (2009)
15. Li, R., Sun, B., Zhang, P., Li, C.: New impossible differential cryptanalysis of ARIA. IACR Cryptology ePrint Archive, 2008:227 (2008). http://eprint.iacr.org/2008/227
16. Li, Y., Wu, W., Zhang, L.: Integral attacks on reduced-round ARIA block cipher. In: Kwak, J., Deng, R.H., Won, Y., Wang, G. (eds.) ISPEC 2010. LNCS, vol. 6047, pp. 19–29. Springer, Heidelberg (2010)
17. Tang, X., Sun, B., Li, R., Li, C., Yin, J.: A meet-in-the-middle attack on reduced-round ARIA. J. Syst. Softw. **84**(10), 1685–1692 (2011)
18. Wenling, W., Zhang, W., Feng, D.: Impossible differential cryptanalysis of reduced-round ARIA and camellia. J. Comput. Sci. Technol. **22**(3), 449–456 (2007)
19. Z'aba, M.R.: Analysis of linear relationships in block ciphers. Master's thesis, Queensland University of Technology, May 2010

A 4.5 Round Distinguisher on ARIA-256

In Fig. 8, we show the 4.5 round distinguisher require for the 8-round attack on ARIA-256 demonstrated in Sect. 5.

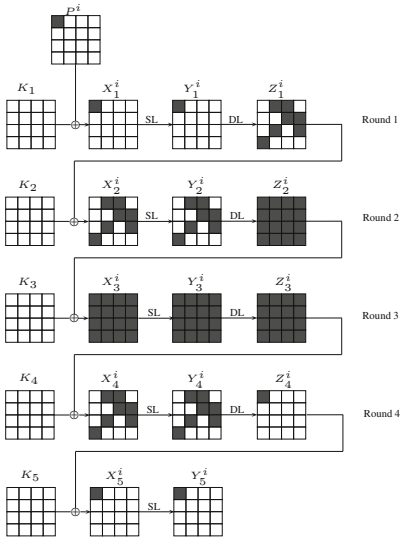

Fig. 8. 4.5-Round distinguisher in ARIA