Attacking Reduced Rounds of the ARIA Block Cipher

Ewan Fleischmann, Michael Gorski, and Stefan Lucks

Bauhaus-University Weimar, Germany
{Ewan.Fleischmann, Michael.Gorski, Stefan.Lucks}@uni-weimar.de

Abstract. ARIA [4] is a block cipher proposed at ICISC'03. Its design is very similar to the advanced encryption standard (AES). The authors propose that on 32-bit processors, the encryption speed is at least 70% of that of the AES. They claim to offer a higher security level than AES. In this paper we present two attacks of reduced round ARIA which shows some weaknesses of the cipher. Moreover, our attacks have the lowest memory requirements compared to existing attacks on ARIA with an increase in the time complexity.

Keywords: block ciphers, differential cryptanalysis, ARIA.

1 Introduction

The ARIA block cipher [4] was presented at ICISC'03. Its design is very similar to the advanced encryption standard (AES/Rijndael) [3]. The block size is 128-bit and the key size is either 128, 192 or 256 bits. It uses the same number of rounds as the AES, which are 10, 12 and 14 respectively. ARIA employs two kinds of S-Boxes and two types of substitution layers which are different between even and odd rounds. They skip using a MixComuns operation and use an 16×16 binary matrix with branch number 8 in their diffusion layer. The authors propose that ARIA can increase the efficiency in 8-bit and 32-bit software implementations in comparison to AES. Moreover, they claim to have better security against all existing attacks on block ciphers.

Wu et al. [7] showed that there exist good impossible differentials to break up to 6 rounds of ARIA. Later Li et al. [5] presented also some impossible differential attacks of up to 6 rounds of ARIA. In this paper we apply another technique on ARIA which is called the boomerang attack. We show that our attack can also break up to 6 rounds of ARIA but with the lowest data complexity compared to previous attacks. Our results should introduce a new technique for cryptanalysis on ARIA and should therefore leave some space for further research.

The boomerang attack [6] is a strong extension to differential cryptanalysis [1] in order to break more rounds than plain differential attacks can, since the cipher is treated as a cascade of two sub-ciphers, using short differentials in each sub-cipher. These differentials are combined in an adaptive chosen plaintext and ciphertext attack to exploit properties of the cipher that have a high probability. Biryukov [2] proposed a similar boomerang attack on the AES-128 which can break up to 5 and 6 out of 10 rounds.
The paper is organized as follows: In Section 2 we give a brief description of the ARIA block cipher. In Section 3 we describe the boomerang attack. In Section 4, we present a boomerang attack on 5-rounds which works on each instance of ARIA. In Section 5 we extend the attack

Table 1. Comparison of attacks on ARIA

Attack	# Rounds	Data	Time	Source
Impossible Differential	5	$2^{71.3}$	$2^{71.6}$	[5]
Boomerang Attack	5	2^{57}	$2^{115.5}$	Section 4
Impossible Differential	6	2^{121}	2^{112}	[7]
Impossible Differential	6	$2^{120.5}$	$2^{104.5}$	[5]
Impossible Differential	6	2^{113}	$2^{121.6}$	[5]
Boomerang Attack	6	2^{57}	$2^{171.2}$	Section 5

on a 6-round attack that is applicable to ARIA-192 and ARIA-256 only. We conclude the paper in Section 6.

2 Description of ARIA

ARIA [4] uses data blocks of 128 bits with an 128, 192 or 256-bit cipher key. A different number of rounds is used depending on the length of the cipher key. ARIA has 10, 12 and 14 rounds when a 128, 192 or 256-bit cipher key is used respectively. The plaintexts are treated as a 4 x 4 byte matrix, which is called state. A round applies three operations to the state:

- **Substitution layer (SL)** is a non-linear byte-wise substitution applied on every byte of the state matrix in parallel, where two different substitution layer exist.
- **Diffusion layer (DL)** is a linear matrix multiplication of the state matrix with a 16×16 involution binary matrix.
- **Round key addition (RK)** is a XORing of the state and a 128-bit subkey which is derived from the cipher key.

Before the first round, an initial RK operation is applied and the DL operation is omitted in the last round. The bytes coordinates of a 4 x 4 state matrix are labeled as:

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

Substitution Layer (SL). ARIA uses two S-Boxes S_1 and S_2 and also their inverse S_1^{-1}, S_2^{-1}. Each S-Box is defined to be an affine transformation of the inversion function over GF(2^8).

$$S_1, S_2 : GF(2^8) \rightarrow GF(2^8)$$

$$S_1 : x \mapsto A \cdot x^{-1} \oplus a,$$

where

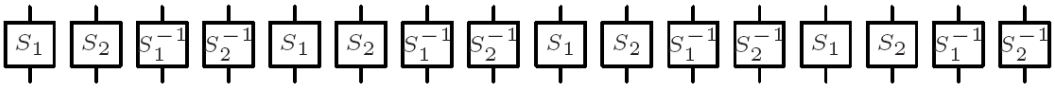
$$A = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \quad \text{and} \quad a = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

$$S_2 : x \mapsto B \cdot x^{-247} \oplus b,$$

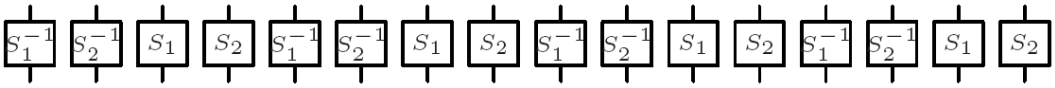
where

$$B = \begin{pmatrix} 0&1&0&1&1&1&1&0 \\ 0&0&1&1&1&1&0&1 \\ 1&1&0&1&0&1&1&1 \\ 1&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&0&0 \\ 1&0&0&0&0&0&0&1 \\ 0&1&0&1&1&1&0&1 \\ 1&1&0&1&0&0&1&1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

ARIA has two types of S-Box layers for even and odd rounds as shown in Figure 1. Type 1 is used in the odd rounds and type 2 is used in the even rounds.



S-Box layer type 1



S-Box layer type 2

Fig. 1. The two types of S-Box layers

Diffusion Layer (DL). The function $A : \mathrm{GF}(2^8)^{16} \rightarrow \mathrm{GF}(2^8)^{16}$ is given by

$$(x_0, x_1, \ldots, x_{15}) \mapsto (y_0, y_1, \ldots, y_{15}),$$

where

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

Round Key Addition (RK). The round keys are derived from the cipher key using the key schedule which uses a 3-round 256-bit Feistel cipher. We skip its introduction since we do not need it for our attack. We refer to [4] for more details.

3 The Boomerang Attack

We now describe the boomerang attack [6] in more detail. But first, we give some definitions.

Definition 1. *Let P, P' be two bit strings of the same length. The bit-wise xor of P and P', $P \oplus P'$, is called the* difference *of P, P'. Let $'a'$ be a known and $'*'$ an unknown non-zero byte difference.*

Definition 2. *$\alpha \to \beta$ is called a* differential *if α is the plaintext difference $P \oplus P'$ before some non-linear operation $f(\cdot)$ and β is the difference after applying these operation, i.e, $f(P) \oplus f(P')$. The probability p is linked on a differential saying that an α difference turns into a β difference with probability p. The backward direction, i.e., $\alpha \leftarrow \beta$ has probability $\hat{p}$.*

Two texts (P, P') are called a *pair*, while two pairs (P, P', O, O') are called a *quartet*. Regularly, the differential probability decreases the more rounds are included. Therefore two short differential covering only a few rounds each will be used instead of a long one covering the whole cipher. We split the boomerang attack into two steps: The *boomerang distinguisher step* and the *key recovery step*. The boomerang distinguisher is used to find all plaintexts sharing a desired difference that depends on the choice of the differential. These plaintexts are used in the key recovery step afterwards to recover subkey bits for the initial round key.

Distinguisher Step. During the *distinguisher step* we treat the cipher as a cascade of two sub-ciphers $E_K(P) = E_K^1(P) \circ E_K^0(P)$, where K is the key used for encryption and decryption. Since we always use the same key we omit the key K and write $E(P) = E^1(P) \circ E^0(P)$ instead. We assume that the differential $\alpha \to \beta$ for E^0 occurs with probability p, while the differential $\gamma \to \delta$ for E^1 occurs with probability q, where α, β, γ and δ are differences of texts. The backward direction $E^{0^{-1}}$ and $E^{1^{-1}}$ of the differential for E^0 and E^1 are denoted by $\alpha \leftarrow \beta$ and $\gamma \leftarrow \delta$ and occur with probability $\hat{p}$ and $\hat{q}$ respectively. The attack works as follows:

1. Choose a pool of s plaintexts P_i, $i \in \{1, \ldots, s\}$ uniformly at random and compute a pool $P_i' = P_i \oplus \alpha$.
2. Ask for the encryption of P_i, i.e., $C_i = E(P_i)$ and ask for the encryption of P_i', i.e., $C' = E(P_i')$.
3. Compute the new ciphertexts $D_i = C_i \oplus \delta$ and $D_i' = C_i' \oplus \delta$.
4. Ask for the decryption of D_i, i.e., $O_i = E^{-1}(D_i)$ and ask for the decryption of D_i', i.e., $O_i' = E^{-1}(D_i')$.

– For each pair (O_i, O_j'), $i, j \in \{1, \ldots, s\}$

5. If $O_i \oplus O_j'$ equals α store the quartet (P_i, P_j', O_i, O_j') into a set M.

A pair (P_i, P_j'), $i, j \in \{1, \ldots, s\}$ with the difference α satisfies the differential $\alpha \to \beta$ with probability p. The output of E_0 is A_i and A_j', i.e., $E^0(P_i) = A_i$ and $E^0(P_j') = A_j'$ have a certain difference $\beta = A_i \oplus A_j'$ with probability p. Using the ciphertexts C_i and C_j' we can compute the new ciphertexts $D_i = C_i \oplus \delta$ and $D_j' = C_j' \oplus \delta$. Let $B_i = E^{1^{-1}}(D_i)$ and $B_j' = E^{1^{-1}}(D_j')$ are the decryption of D_i and D_j' with $E^{1^{-1}}$ $i \in \{1, \ldots s\}$. A difference δ turns into a difference γ after passing $E^{1^{-1}}$ with probability $\hat{q}$. Since $\delta = C_i \oplus D_i$ and $\delta = C_j' \oplus D_j'$ we know that $\gamma = A_i \oplus B_i$ and $\gamma = A_j' \oplus B_j'$ with probability $\hat{q}^2$. Since we also know, that $A_i \oplus A_j' = \beta$ with probability p, it follows that $(A_i \oplus B_i) \oplus (A_i \oplus A_j') \oplus (A_j' \oplus B_j') = \gamma \oplus \beta \oplus \gamma = \beta = (B_i \oplus B_j')$ holds with probability $p \cdot \hat{q}^2$. A β difference turns into an α difference after passing the differential $E^{0^{-1}}$ with probability $\hat{p}$. Thus, a pair of plaintexts (P_i, P_j') with $P_i \oplus P_j' = \alpha$ generates a new pair of plaintexts (O_i, O_j') where $O_i \oplus O_j' = \alpha$ with probability $p \cdot \hat{p} \cdot \hat{q}^2$. A quartet containing these two pairs is defined as:

Definition 3. *A quartet (P_i, P_j', O_i, O_j') which satisfies*

$$
\begin{aligned}
&P_i \oplus P_j' = \alpha = O_i \oplus O_j',\\
&A_i \oplus A_j' = \beta = B_i \oplus B_j',\\
&A_i \oplus B_i = \gamma = A_j' \oplus B_j',\\
&C_i \oplus D_i = \delta = C_j' \oplus D_j',
\end{aligned}
$$

is called a ***correct boomerang quartet*** *which occurs with probability $Pr_c = p \cdot \hat{p} \cdot \hat{q}^2$. A quartet (P_i, P_j', O_i, O_j') which only satisfies the condition $P \oplus P_j' = \alpha = O_i \oplus O_j'$ is called a* ***false boomerang quartet****.*

Figure 2 displays the structure of the boomerang distinguisher step. Any attacker who applies a boomerang distinguisher does not know the internal states A_i, A_j', B_i, B_j', since he can only

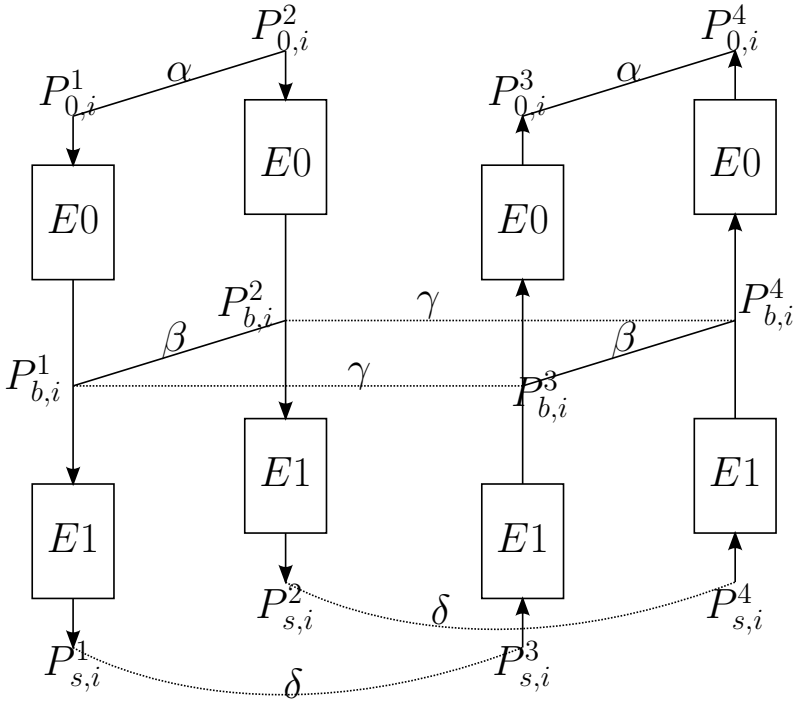


Fig. 2. The boomerang distinguisher

apply a chosen plaintext and ciphertext attack on the cipher. The set M which is the output of the boomerang distinguisher, therefore contains correct and false boomerang quartets. It is impossible to form another distinguisher which separates the correct and the false boomerang quartets, since the interior differences β and γ cannot be computed.

Key Recovery Step. The second step of the boomerang attack is the *key recovery step*. From now on, an attacker operates on the set M that was stored by the boomerang distinguisher. Let k be some key bits of the last round keys derived from the cipher keys K. Let $d_k(C)$ be the one round partial decryption of C under the key k. The key recovery step works as follows:

- For each key-bit combination of k
  1. Initialize a counter for each key-bit combination with zero.
  - For all quartets (P, P', O, O') stored in M
    2. Ask for the encryption of P, P', O, O' and obtain the ciphertext quartet C, C', D, D' respectively. Decrypt the ciphertexts C, C', D, D', i.e., $\bar{C} = d_k(C), \bar{C'} = d_k(C'), \bar{D} = d_k(D)$ and $\bar{D'} = d_k(D')$.
    3. Test whether the differences $\bar{C} \oplus \bar{D}$ and $\bar{C'} \oplus \bar{D'}$ have a desired difference an attacker would expect depending on the differential being used. Increase a counter for the used key-bits if the difference is fulfilled in both pairs.
4. Output the key-bits k with the highest counter as the correct one.

Four cases can be differentiated in Step 3, since M contains correct and false boomerang quartets and the key-bit combination k can either be correct or false. A correct boomerang

quartet encrypted with the correct key bits will have the desired difference needed to pass the test in Step 3 with probability 1. Hence, the counter for the correct key bits is increased. The three other cases are: a correct boomerang quartet is used with false key bits (Pr_{cK_f}), a false boomerang quartet is used with the correct key-bits (Pr_{fK_c}) or a false boomerang quartet is used with a false key-bit combination (Pr_{fK_f}). We assume that the cipher acts like a random permutation. In these cases we assume that

$$Pr_{cK_f} = Pr_{fK_c} = Pr_{fK_f} =: Pr_{filter}.$$

The probability that a quartet in one of the three undesirable cases is counted for a certain key bit combination is Pr_{filter}. The differentials have to be chosen such that the counter of the correct key bits is significantly higher than the counter of each false key bit combination. If the differentials have a high probability the key recovery step outputs the correct key-bits in Step 4 with a high probability much faster than exhaustive search.

4 Boomerang Attack on 5-Round ARIA

In this section we mount a boomerang attack 5-round ARIA-128. The cipher is represented as $E = E^1 \circ E^0$. E^0 is a differential containing rounds 1 to 3. E^1 is a differential covering rounds 4 to 5. After applying the boomerang distinguisher for $E^1 \circ E^0$ using the differentials E^0 and E^1 we apply it to recover 56 key-bits of the initial round-keys. We assumed, that the S-Box acts like a random permutation. Thus, all S-Box output differences will have the same probability for a given input difference. The notation used in our attack will be defined as:

- P_i, P'_j, O_i, O'_j plaintexts.
- C_i, C'_j, D_i, D'_j ciphertexts.
- $E^0_{K_i}(\cdot)$ 3-round ARIA encryption from round 1 to 3 under key K_i, $i \in \{a, b, c, d\}$.
- $E^{1^{-1}}_{K_i}(\cdot)$ 2-round ARIA decryption from round 5 to 4 under key K_i, $i \in \{a, b, c, d\}$.
- a is a known non-zero byte difference.
- $*$ is a variable unknown non-zero byte differences.

4.1 The Differential E^0

The input difference α of E^0 has a non-zero difference in bytes 3, 4, 6, 8, 9, 13 and 14. A non-zero difference $*$ transforms into an a difference through SL with probability 2^{-8}. Thus, we have an a difference in bytes 3, 4, 6, 8, 9, 13 and 14 with probability 2^{-56}. DL_1 then leaves an a difference in byte 0, while the remaining bytes become zero. Since RK is linear it will not alter this difference. SL_2 produces a non-zero difference in byte 0 and DL_2 spreads this difference in bytes 3, 4, 6, 8, 9, 13 and 14. At the end of the differential we obtain a difference called β_{out} where all the 16 bytes of the state difference are non-zero. The probability of the differential E^0, i.e., the transformation of an α difference into a β_{out} difference is given by

$$Pr(\alpha \rightarrow \beta_{out}) = 2^{-56}.$$

The differential E^0 is shown in Figure 3.

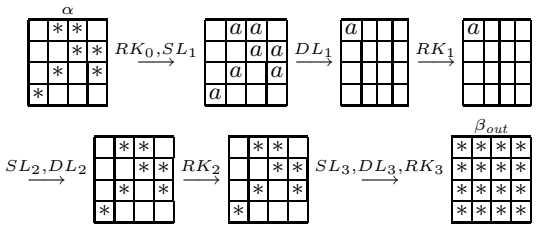


Fig. 3. The differential E^0

4.2 The Differential $E^{1^{-1}}$

The input difference δ consists of one non-zero difference in byte 0 and and a zero difference in the remaining bytes. The non-zero difference remains after the the inverse of round 5. The DL_4 operation spreads this non-zero difference to the bytes 3, 4, 6, 8, 9, 13 and 14. We call the remaining difference of the state γ. The probability of $E^{1^{-1}}$ is $\Pr(\gamma \leftarrow \delta) = 1$. The differential $E^{1^{-1}}$ is shown in Figure 4.

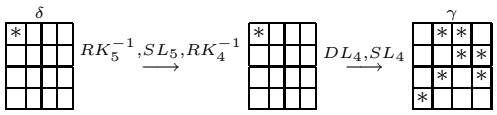


Fig. 4. The differential $E^{1^{-1}}$

4.3 The Differential $E^{0^{-1}}$

For the following steps we need that the output difference β_{out} of the differential E^0 is equal to the input difference β_{in} for the differential $E^{0^{-1}}$. Note that β_{in} and β_{out} are not only equal in the same positions of non-zero differences but are also equal in each byte. We will shown how to construct such a case. From the boomerang condition inside the cipher for two differences γ_1 and γ_2 we know that

$$\beta_{out} \oplus \gamma_1 \oplus \gamma_2 = \beta_{in}$$

holds with some probability. Since γ_1 and γ_2 are equal in each byte, we simply write γ. Thus the above condition reduces to:

$$\beta_{out} \oplus \gamma \oplus \gamma = \beta_{out} = \beta_{in} \tag{1}$$

Using the differentials above, the differences β_{in} and β_{out} are equal with probability one. Note that these difference occur only with some probability, which will be described more detailed later.

Let A, A', B, B' be the internal state after SL_3 in forward direction when encrypting P, P', O, O' respectively. The notation from Figure 2 is used. Since DL is linear γ can be expressed as

$$\gamma = K_3 \oplus DL_3(A) \oplus K_3 \oplus DL_3(B) = DL_3(A \oplus B) \tag{2}$$

and as

$$\gamma = K_3 \oplus DL_3(A') \oplus K_3 \oplus DL_3(B') = DL_3(A' \oplus B'). \tag{3}$$

Equation (2) and (3) can be combined, which leaves $A \oplus A' = B \oplus B'$. In other words, DL_3 can be undone with the probability 1 due to the boomerang condition (1). This means that we know exactly that after DL_3 in backward direction the bytes 3, 4, 6, 8, 9, 13 and 14 are non-zero while the remaining bytes are zero. SL_3 an a difference in bytes 3, 4, 6, 8, 9, 13 and 14 with probability 2^{-56}. DL_2 outputs an a difference in byte 0 and a zero difference in the remaining bytes. SL_2 then transforms the a difference in byte 0 into a non-zero difference, which is spread into the bytes 3, 4, 6, 8, 9, 13 and 14 after DL_1. The output difference α of the differential $E^{0^{-1}}$ contains these non-zero and zero differences. The differential $E^{0^{-1}}$ has the probability $\Pr(\alpha \leftarrow \beta_{in}) = 2^{-56}$ to occur. It is shown in Figure 5.

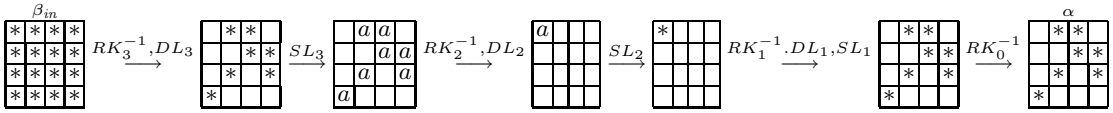


Fig. 5. The differential $E^{0^{-1}}$

4.4 The Attack

The attack first mounts a boomerang distinguisher to obtain all correct and false boomerang quartets which are stored in M. A key-search is then applied to M in order to find 56 bits of K_0. Let k_0 be an 56-bit subkey in the position of bytes 3, 4, 6, 8, 9, 13 and 14. Let $e_{0,k}(X)$ be the partially encryption of X under the subkey k before DL_1 is applied. The attack is as follows:

1. Choose $2^{58.5}$ structures $S_1, S_2, \ldots, S_{2^{58.5}}$ of 2^{56} plaintexts P_i, $i \in \{1, 2, \ldots, 2^{58}\}$ which have all possible values in seven bytes (3, 4, 6, 8, 9, 13 and 14). With a chosen plaintext scenario ask for encryption of P_i to obtain the ciphertexts C_i, i.e., $C_i = E(P_i)$.
2. For each ciphertext C_i compute a new ciphertext $D_i = C_i \oplus \delta$, where δ is a fixed 128-bit value with a non-zero value in byte 0 and zero in the remaining bytes.
3. With a chosen ciphertext scenario ask for the decryption of D_i and obtain the new ciphertexts O_i, i.e. $O_i = E^{-1}(D_i)$.

4. Store only those quartets (P_i, P'_j, O_i, O'_j) in the set M where $O_i \oplus O'_j$ have a non-zero difference in bytes 3, 4, 6, 8, 9, 13 and 14 and a zero difference in the remaining bytes.
5. For each 56-bit key k

   For each quartet passing the test in Step 5:

   5.1. Partially encrypt a plaintext quartet (P_i, P_j, O_i, O_j), i.e., $\bar{P}_i = e_{0,k}(P_i)$, $\bar{P}_j = e_{0,k}(P_j)$, $\bar{O}_i = e_{0,k}(O_i)$ and $\bar{O}_j = e_{0,k}(O_j)$.

   5.2. Increase a counter for the used 56-bit subkey k by one if $\bar{P}_i \oplus \bar{P}_j$ and $\bar{O}_i \oplus \bar{O}_j$ have an a-difference in bytes 3, 4, 6, 8, 9, 13 and 14.
6. Output the 56-bit subkey k which counts at least two quartets as the correct one.

4.5 Analysis of the Attack

Two pools of 2^{56} plaintexts can be combined to approximately $\frac{(2^{56})^2}{2} = 2^{111}$ quartets. Using $2^{58.5}$ structures we obtain $\#PP \approx 2^{58.5} \cdot 2^{58.5} = 2^{169.5}$ quartets in total. A correct boomerang quartet occurs with probability

$$\begin{aligned} Pr_c &= \Pr(\alpha \rightarrow \beta_{out}) \cdot (\Pr(\gamma \leftarrow \delta))^2 \cdot \Pr(\gamma_1 = \gamma_2) \cdot \Pr(\alpha \leftarrow \beta_{in}) \\ &= 2^{-56} \cdot 1 \cdot 2^{-56} \cdot 2^{-56} = 2^{-168}, \end{aligned}$$

since all differential conditions are fulfilled. A random permutation of a difference $O_i \oplus O_j$ has 9 zero byte difference with probability $Pr_f = 2^{-72}$. Thus, after Step 4 we have about $\#C = \#PP \cdot Pr_c = 2^{169.5} \cdot 2^{-168} = 2^{1.5}$ correct and $\#F = \#PP \cdot Pr_f = 2^{169.5} \cdot 2^{-72} = 2^{97.5}$ false boomerang quartets. A false boomerang quartet passes the test in Step 5.2 with probability $Pr_{filter} = 2^{-112}$, since we have a 56-bit filtering condition on both pairs of a quartet. Thus $\#CK_c = 2^{1.5}$ correct boomerang quartets and $\#FK_c = \#F \cdot Pr_{filter} = 2^{97.5} \cdot 2^{-112} = 2^{-14.5}$ false boomerang quartets are counted with the correct key bits. About $\#CK_c + \#FK_c = 2^{1.5} + 2^{-14.5} \approx 3$ quartets are counted in Step 5.2 for the correct key bits.

About $\#CK_f = \#C \cdot Pr_{filter} = 2^{1.5} \cdot 2^{-112} = 2^{-110.5}$ correct boomerang quartets and $\#FK_f = \#F \cdot Pr_{filter} = 2^{97.5} \cdot 2^{-112} = 2^{-14.5}$ false boomerang quartets are counted with the false key bits, which are in total $\#CK_f + \#FK_f = 2^{-110.5} + 2^{-14.5} = 2^{-14.5}$ counts for each false key bit combination.

Using the Poisson distribution we can compute the success rate of our attack. The probability that the number of remaining quartets for each false key bit combination is larger than 1 is $Y \sim Poisson(\mu = 2^{-14.5})$, $\Pr(Y \geq 2) \approx 0$. Therefore the probability that our attack outputs false key bits as the correct one is very low. We expect to have a count of 3 quartets for the correct key bits. The probability that the number of quartets counted for the correct key bits is larger than 1 is $Z \sim Poisson(\mu = 3)$, $\Pr(Z \geq 2) \approx 0.8$.

Each structure can be analyzed sequentially. Thus the data complexity is determined by Step 1 to 3, which is about $2 \cdot 2^{56} = 2^{57}$ chosen plaintexts and ciphertexts. The data complexity of Step 4, 5.1 and 5.2 is negligible compared to the data complexity of the first two steps. The time complexity of Step 1 to 3 is $2 \cdot 2^{56} = 2^{57}$ encryptions. Since we have to run these steps for each structure the overall time complexity is about $2^{58.5} \cdot 2^{57} = 2^{115.5}$ five round ARIA-128 encryptions. Note that due to the complexity of this attack it can be applied to each instance of ARIA.

5 Boomerang Attack on 6-Round ARIA-192 and ARIA-256

The attack of the previous section can be easily extend to a 6-round attack on ARIA-192 and ARIA-256. Therefore we need the following property of ARIA.

Property 1. The round key addition (RK) and the diffusion layer (DL) can be interchanged, due to its linearity.

Using this property we can change the order of DL_5 and RK_5. Thus we can use our 5-round boomerang distinguisher to apply a 6-round attack in the following way. We add one round after the boomerang distinguisher as shown in Figure 6. We can guess 7 byte of K_6 at bytes 3,

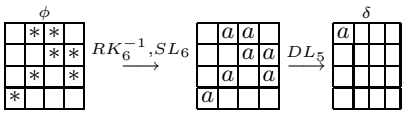


Fig. 6. The round after the distinguisher

4, 6, 8, 9, 13 and 14. This will allow us to choose the desired difference ϕ such that after SL_6 in backward direction a known difference a occurs in each of these bytes. The DL_5 operation then outputs an a difference in byte 0 while the remaining bytes become zero. From this point the 5-round boomerang distinguisher works as explained above. The data complexity of our 6-round attack remains the same as the 5-round attack which are 2^{57} chosen plaintexts and ciphertexts. The overall time complexity increases to $2^{56} \cdot (5/6) \cdot 2^{115.5} \approx 2^{171.2}$ six round ARIA-192 encryptions. This attack is also applicable to ARIA-256 but not on the 128 bit version.

6 Conclusion

In this paper we have shown how to attack ARIA using the boomerang attack, which is a strong extension of differential cryptanalysis. Our 5-round boomerang attack on ARIA-128 has a data complexity of 2^{57} chosen plaintexts and ciphertexts. Its time complexity is of about $2^{115.5}$ five round encryptions. We extended this attack to mount a 6-round boomerang attack on ARIA-192 which has the same data complexity as our 5-round attack. The time complexity of our 6-round boomerang attack is about $2^{171.2}$ six round encryptions.

To the best of our knowledge there are no better attacks on ARIA in terms of data complexity or number of attacked rounds than the 6-round attack presented in this paper.

References

[1] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

[2] Alex Biryukov. The Boomerang Attack on 5 and 6-Round Reduced AES. In Hans Dobbertin, Vincent Rijmen, and Aleksandra Sowa, editors, *AES Conference*, volume 3373 of *Lecture Notes in Computer Science*, pages 11–15. Springer, 2004.

[3] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002.

[4] Daesung Kwon, Jaesung Kim, Sangwoo Park, Soo Hak Sung, Yaekwon Sohn, Jung Hwan Song, Yongjin Yeom, E-Joong Yoon, Sangjin Lee, Jaewon Lee, Seongtaek Chee, Daewan Han, and Jin Hong. New Block Cipher: ARIA. In Jong In Lim and Dong Hoon Lee, editors, *ICISC*, volume 2971 of *Lecture Notes in Computer Science*, pages 432–445. Springer, 2003.

[5] Peng Zhang Ruilin Li, Bing Sun and Chao Li. New Impossible Differential Cryptanalysis of ARIA. Cryptology ePrint Archive, Report 2008/227, 2008. http://eprint.iacr.org/.

[6] David Wagner. The Boomerang Attack. In Lars R. Knudsen, editor, *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.

[7] Wenling Wu, Wentao Zhang, and Dengguo Feng. Impossible Differential Cryptanalysis of Reduced-Round ARIA and Camellia. *J. Comput. Sci. Technol.*, 22(3):449–456, 2007.