Revised Unified Impossible Differential Cryptanalysis of ARIA

Yixin Zhong[1,2], Weijia Xue[1], Daoguang Mu[2], Xuejia Lai[1]
[1]Institute of Cryptology and Information Security, Dept of Computer Science and Technology
Shanghai Jiaotong University
Shanghai, China
[2]Science and Technology on Communication Security Laboratory
Chengdu, China
Corresponding author: lai-xj@cs.sjtu.edu.cn

Abstract—Unified Impossible Differential (UID) crypt-analysis is a systematic method to find impossible differentials for block ciphers and there are large amount of cryptanalysis results coming out by using it. ARIA is a Korean block cipher expecting no impossible differential chains on four or more rounds. In this paper, we apply UID to ARIA and 89136 four rounds impossible differential chains are found. With the optimization of the conflict searching algorithms, UID gets better results compared with former cryptanalysis results. Moreover, we conclude that no impossible differential chains with number of rounds larger than four can be found by the UID method.

Keywords—impossible differential cryptanalysis, UID, ARIA, block cipher

I. INTRODUCTION

Impossible differential cryptanalysis [1] is an effective analysis method in symmetric cryptography, which has already been applied in many cryptanalysis and got optimistic results. Technically, impossible differential cryptanalysis uses impossible differential chains with 0-possiblilty to rule out all the wrong keys and find the correct one.

Unified Impossible Differential cryptanalysis (UID) [2] is inspired by the previous U-method, which is used to find the impossible differential characteristics for block cipher structures. Since UID makes a great performance in the problem of retrieving the impossible differential characteristics of block ciphers, we apply the UID method to ARIA.

ARIA [3] has a structure similar to AES (Advanced Encryption Standard): it is designed by Korean scientists and is a 128-bit substitution-permutation network block cipher. Each round of the cipher consists of three parts, round key addition (RKA), substitution layer (SL) and diffusion layer (DL). The impossible differential characteristic does not change through RKA. As a result, we can focus on the transformation of SL and DL.

The previous cryptanalysis of ARIA includes a dedicated linear attack, truncated differential cryptanalysis and impossible differential cryptanalysis. Wu and so on [4] find four 4-round impossible differential chains in the same form while Li and so on [5] find five in another form. Xue and so on [6] use the UID method and find 6721 4-round impossible differential chains which are more varied in form.

We use the UID method and find 89136 impossible differential chains with the length of four rounds. We failed to find impossible differential chains with number of rounds larger than four and we conclude that the UID method cannot find impossible differential chains with number of rounds larger than four.

II. DESCRIPTION OF ARIA

A. Round Transformation

ARIA is an involution SPN block cipher and each round of the cipher consists of the following three parts:

a) Round key addition: XOR the round key, which is 128-bit long and generated by round key generation.

b) Substitution layer: There're two types of substitution layers, one for odd round and the other for even rounds.

c) Diffusion layer: It's a linear map which is an involution.

Note that in the last round, the diffusion layer is replaced by a round key addition.

Substitution Layer. We use two 8-bit S-boxes S_1, S_2 and their inverses S_1^{-1}, S_2^{-1}. In odd rounds, the substitution layer is $(S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1})$; In even rounds, the substitution layer is $(S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2, S_1^{-1}, S_2^{-1}, S_1, S_2)$.

Diffusion Layer. The diffusion layer $\mathrm{DL}: \mathrm{GF}(2^8)^{16} \to \mathrm{GF}(2^8)^{16}$ is given by

$$(x_1, x_2, \cdots, x_{15}) \mapsto (y_1, y_2, \cdots, y_{15}).$$

And

$$y_0 = x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14},$$

$$y_1 = x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15},$$

$$y_2 = x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15},$$

$$y_3 = x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14},$$

$$y_4 = x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15},$$

$$y_5 = x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15},$$

$$y_6 = x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13},$$

$$y_7 = x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13},$$

$$y_8 = x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15},$$

$$y_9 = x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14},$$

$$y_{10} = x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15},$$

$$y_{11} = x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14},$$

$$y_{12} = x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12},$$

$$y_{13} = x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13},$$

$$y_{14} = x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14},$$

$$y_{15} = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}. \quad [1]$$

B. Key Schedule

Key schedule consists two parts, initialization and round key generation.

In the initialization part, four 128-bit values W_0, W_1, W_2, W_3 are generated from the master key (denoted as MK).

Note that MK can be 128, 192 or 256 bit size. We first fill out the 128-bit value KL (key left) with bits from MK and use the remaining of MK as KR (key right). Both KL and KR are 128-bit value and the space remaining on KR (if any) is filled with zero. Thus, we have

$$KL \parallel KR = MK \parallel 000\cdots0. \quad [2]$$

Then we set

$$W_0 = KL,$$

$$W_1 = F_o(W_0, CK_1) \oplus KR,$$

$$W_2 = F_e(W_1, CK_2) \oplus W_0,$$

$$W_3 = F_o(W_2, CK_3) \oplus W_1. \quad [3]$$

Here, F_o and F_e are odd and even round functions. The 128-bit keys CK_i(i=1, 2, 3, 4) of round functions are fixed to be the rational part of π^{-1}.

The round key generation part combines the four values W_i to obtain an encryption round key ek_i and the decryption round key dk_i, which is given as follow:

$$ek_1 = (W_0^{\ggg 7}) \oplus (W_1^{\lll 11}),$$

$$ek_2 = (W_1^{\lll 22}) \oplus (W_2),$$

$$ek_3 = (W_2^{\ggg 17}) \oplus (W_3^{\lll 16}),$$

$$ek_4 = (W_0^{\ggg 14}) \oplus (W_3^{\lll 32}),$$

$$ek_5 = (W_0^{\ggg 21}) \oplus (W_2^{\ggg 34}),$$

$$ek_6 = (W_1^{\lll 33}) \oplus (W_3^{\lll 48}),$$

$$ek_7 = (W_1^{\lll 44}) \oplus (W_2^{\ggg 51}),$$

$$ek_8 = (W_0^{\ggg 28}) \oplus (W_3^{\lll 64}),$$

$$ek_9 = (W_1^{\lll 55}) \oplus (W_3^{\lll 80}),$$

$$ek_{10} = (W_0^{\ggg 35}) \oplus (W_2^{\ggg 68}),$$

$$ek_{11} = (W_0^{\ggg 42}) \oplus (W_1^{\lll 66}),$$

$$ek_{12} = (W_1^{\lll 77}) \oplus (W_2^{\ggg 85}) \oplus (W_3^{\lll 96}),$$

$$ek_{13} = (W_0^{\ggg 49}) \oplus (W_2^{\ggg 102}),$$

$$ek_{14} = (W_2^{\ggg 119}) \oplus (W_3^{\lll 112}) \oplus (KR^{\lll 64}),$$

$$ek_{15} = (W_0^{\ggg 56}) \oplus (W_1^{\lll 88}) \oplus (KR). \quad [4]$$

Note that the number of rounds are 10, 12 or 14 respect of the size 128, 192 or 256 of the MK size. Since there is one more key addition layer, the 128-bit round keys are needed in the 11th, 13th, or 15th round.

The decryption round keys are derived from the encryption round keys:

$$dk_i = ek_{n+2-i}, i = 1 \text{ or } n+1, \quad [5]$$

$$dk_j = DL(ek_{n+2-j}), j = 2,3,4,\cdots,n. \quad [6]$$

Note that DL represents for the diffusion layer here.

III. Description of the Analysis Process

This section briefly describes the analysis process, including the UID identity, the characteristic matrix of the round function and the way to find contradictions.

A. UID Identity

UID identity is a triple <*L*, *M*, *R*>, where *L*, *M*, *R* represent the XOR results of nonzero fixed differences (l_i), nonzero unknown differences (m_i) and unknown differences (r_i) respectively.

First, we define that the addition operation of two UID identity is actually XOR operation, which means if a certain variable appears even number of times in the same UID identity, it has no affects about the value of such UID identity.

Other than addition, we define three more operations for UID identity, shown as Table 1.

TABLE I. THREE OPERATIONS OF THE UID IDENTITY

Operation	INPUT	OUTPUT
$\mathbf{0}(\mu)$	$< L, M, R >$	$< 0, 0, 0 >$
$\mathbf{1}(\mu)$	$< L, M, R >$	$< L, M, R >$
$\varphi(\mu)$	$< 0, 0, 0 >$	$< 0, 0, 0 >$
	$< L, 0, 0 >$	$< 0, M_{new}, 0 >$
	$< 0, M, 0 >, \lvert M \rvert = 1$	$< 0, M_{new}, 0 >$
	otherwise	$< 0, 0, R_{new} >$

In which m_j denotes a new nonzero unknown difference and r_j denotes a new unknown difference.

B. Characteristic Matrix of ARIA Round Function

Because the cipher structure has 16 data subblocks in every round, the encryption and decryption characteristic matrices are 16×16 matrices.

Note that the S-box is a random transformation or a nonlinear transformation. Hence, in the substitution layer, each subblock would go through a nonlinear transformation. Therefore, the characteristic matrix of substitution layer is a diagonal matrix (denoted as ε_1) as follow:

$$\begin{pmatrix} \varphi & 0 & 0 & \cdots & 0 \\ 0 & \varphi & 0 & \cdots & 0 \\ 0 & 0 & \varphi & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \varphi \end{pmatrix}.$$

And according to chapter 2.1, the characteristic matrix of diffusion layer is a matrix with only 0 and 1 (denoted as ε_2) as follow:

$$\begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix}$$

In each column and row, there are 7 position of 1 and others are 0. Here 1 means identity mapping while 0 means zero mapping given in Table 1.

The characteristic matrix of encryption round function is $\varepsilon_1 \cdot \varepsilon_2$. Since the inverse of ε_2 is itself, the characteristic matrix of decryption round function is $\varepsilon_2 \cdot \varepsilon_1$.

C. Selection of Plaintext and Ciphertext Difference

In encryption, because the plaintext difference goes through the substitution layer first, there's no need to determine whether each block of the differences are the same fixed difference or not. We can merely focus on whether the input block has a non-zero difference or not. Each block has two possible values, 0 or 1 and 0 indicate zero difference and 1 indicate a fixed difference.

In decryption, the ciphertext first goes through the diffusion layer. Hence, whether the fixed differences are equal or not will affect the output. However, according to the conflict searching algorithms, we can leave out the diffusion layer in the last round. Thus, we only consider two possible values, 0 or 1 of the ciphertext difference, the same as encryption.

D. Searching of Confliction

The dimension of the UID vector for the intermediate difference is 16. Our algorithm computes two rounds from the encryption and decryption direction separately, and then takes two UID vectors one from the encryption and the other from the decryption direction and judges whether they are inconsistent. Specifically, we need to search all over nonempty subsets of {1, 2, 3, …, 16}, 65535 subsets in total. For each subset, compare the two XOR results of UID differences in it and see whether there is a contradiction, which means the

condition that two fixed differences are not equal or a nonzero difference is equal to a zero difference. When any inconsistency is found, we stop the traversal. [6]

IV. Improvement on Algorithms

Each block of the plaintext and ciphertext difference has two possible values, 0 or 1. To exhaust all the difference pairs, the input data complication is $2^{16} \cdot 2^{16} = 2^{32}$ and the judging process has a complication of 2^{16} at the worst, under the circumstance that all the subsets are judged and no inconsistency is found. Without any optimization or distributed computing, we cannot obtain the result within a tolerable time. Hence, we have done several improvements on the searching algorithms.

a) In the substitution layer, since the characteristic matrix is a diagonal matrix, instead of multiplying the UID vector with matrix ε_1, we do the nonlinear transformation φ to each UID identity of the UID vector. The time complexity of multiplying an $n \times n$ matrix and an $n \times 1$ vector is $O(n^2)$, while the time complexity of doing the nonlinear transformation φ to each item of the UID vector is $O(n)$.

b) When searching all over the nonempty subsets of {1, 2, 3,…, 16}, we need to do XOR operation to the UID identities. We first list all the UID identities together, sort all the difference of them and then use a for-loop to detect whether each difference should remain in the final UID identity after the XOR operation.

c) Before the conflict searching, we do a pre-judging on the UID vector generated by the ciphertext difference. Each UID vector has 16 UID identities. We do not take the identities with unknown differences in to account in the traversal. For example, if in the UID vector there is only 3 identities (the 1st, 4th, 10th) that do not have unknown differences, then we need to search all over nonempty subsets of {1, 4, 10}, 7 subsets instead of 65535 subsets. Because the last transformation from the encryption direction is a linear function and the UID identity goes through a diffusion progress, we cannot do such pre-judging on the UID vector generated by the plaintext difference.

V. Detailed Results

We get 89136 impossible differential chains in total and their length are all four rounds, two rounds from the encryption direction and two rounds from the decryption direction.

Table 2 provides some impossible differential chains found by the UID method which never mentioned in the former research.

TABLE II. Examples of impossible differential chains of ARIA

{0,0,1,0,1,0,0,0,1,1,0,0,1,0,0,0} → {0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0}
{0,0,0,0,1,0,0,1,1,0,1,0,0,1,0,1} → {0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0}
{0,0,1,1,0,0,0,1,1,0,1,0,0,1,0,1} → {1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0}
{0,1,0,0,1,0,0,0,1,1,0,0,1,0,0,1} → {0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0}
{1,0,0,0,1,0,0,1,0,0,1,0,0,0,1,1} → {0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0}
{1,0,0,1,0,0,0,1,0,1,0,1,1,0,1,0} → {0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0}
{0,0,0,0,0,0,0,1,0,1,1,0,0,0,1,0} → {0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0}
{1,0,1,0,0,0,1,0,0,0,0,0,0,1,0,0} → {0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0}

And Fig. 1 shows the distribution of all the impossible differential chains.

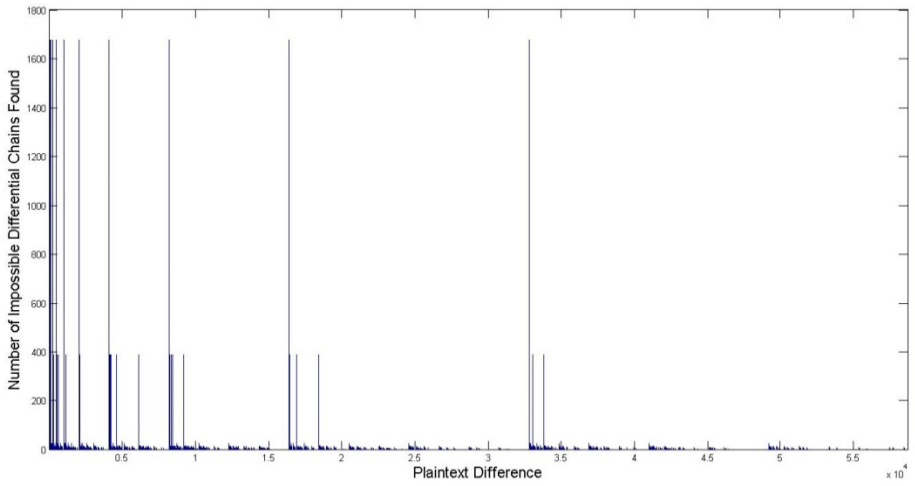

Fig. 1. The distribution of all the impossible differential chain

We analyze the weight of the results found by the UID method. Here, the 'weight' means the number of 1 in the plaintext difference or the ciphertext difference. For example, the weight of '0100000010000000' is 2. And here are some characteristics of the results:

a) Among all the impossible differential chains, the weights of both the plaintext and ciphertext differences are not more than 7.

b) Among all the impossible differential chains, when the weight of the plaintext difference is larger than 2, the weight of the corresponding ciphertext difference is either 1 or 2.

c) Among all the 120 impossible differential chains whose plaintext difference with weight 2, 48 of them have a fixed number of ciphertext differences whose weight are larger than 2, and the fixed number is 390.

We have not found any impossible differential chains with the number of rounds larger than four. However, it doesn't mean such impossible differential chains don't exist. We consider it as the result of the limitation of the UID method. UID needs to extend the impossible differential chains, which is not simple. Specifically, when applying UID to ARIA, if we encrypt or decrypt all the 2^{16} possible differences for three rounds and analyze the intermediate differences respectively, all the 65536 XOR results for the subset of {1, 2, 3,…, 16} will be unknown. Thus we could not find any inconsistency.

VI. Conclusion

In this paper, we apply the UID method to ARIA. We improve the algorithm and traverse all over the plaintext difference and the ciphertext difference pairs (2^{32} *in total*). We find 89136 four rounds impossible differential chains and we also prove that using the UID method cannot find impossible differential chains with number of rounds larger than four.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61272440, 61472251), and China Postdoctoral Science Foundation (2013M531174, 2014T70417).

References

[1] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials", Advances in Cryptology - EUROCRYPT'99, Springer-Verlag, LNCS1592, 1999, pp.12-23.

[2] Y. Luo, Z. Wu, X. Lai, and G. Gong, "A Unified Method for Finding Impossible Differentials of Block Cipher Structures", Cryptology ePrint Archive: Report 2009/627, available at: http://eprint.iacr.org/2009/627.pdf.

[3] D. Kwon, J. Kim, S. Park et al., "New Block Cipher: ARIA", Information Security and Cryptology - ICISC 2003, Springer-Verlag, LNCS 2971, 2004, pp.432-445.

[4] 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 22.3 (2007): 449-456.

[5] S. Li, and C. Song, "Improved impossible differential cryptanalysis of ARIA." Information Security and Assurance, 2008. ISA 2008. International Conference on. IEEE, 2008.

[6] W., Xue, and X. Lai, "Unified Impossible Differential Cryptanalysis of ARIA." CHINA COMMUNICATIONS 9.8(2012): 129-134.83.

[7] C. Du, and J. Chen, "Impossible differential cryptanalysis of ARIA reduced to 7 rounds." Cryptology and Network Security. Springer Berlin Heidelberg, 2010. 20-30.

[8] L. ZHANG, and J. S. GUO, "Best impossible differential cryptanalysis of 6-round ARIA." Journal of the Graduate School of the Chinese Academy of Sciences 2 (2011): 021.

[9] R. Li, B. Sun, P. Zhang, and C. Li, "New Impossible Differential Cryptanalysis of ARIA." IACR Cryptology ePrint Archive 2008 (2008): 227.

[10] R. Li, B. Sun, and C. Li, "Impossible differential cryptanalysis of SPN ciphers." Information Security, IET 5.2 (2011): 111-120.