Automatic Search of Truncated Impossible Differentials and Applications

Shengbao Wu[1,2], Mingsheng Wang[3]

1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, PO Box 8718, China
2. Graduate School of Chinese Academy of Sciences, Beijing 100190, China
3. State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

`wushengbao@is.iscas.ac.cn`
`mingsheng_wang@yahoo.com.cn`

Abstract. Finding the longest impossible differentials is an essential assignment in proceeding impossible differential cryptanalysis. In this paper, we introduce a novel tool to search the longest truncated impossible differentials for word-oriented block ciphers with bijective S-boxes. It costs polynomial time to return a flag indicating whether a truncated differential is impossible under several filter conditions. To demonstrate the strength of our tool, we show that it allows to automatically find the longest truncated impossible differentials for many word-oriented block ciphers. It independently rediscovers all known truncated impossible differentials on nine round CLEFIA. What's more, it finds new and longest truncated impossible differentials for the AES, ARIA, Camellia without FL and FL^{-1} layers, E2, MIBS, LBlock and Piccolo. Finally, we give an impossible differential of 14-round LBlock to illustrate that our tool is more powerful than the $\mathcal{U}$-method and UID-method. We expect that the tool proposed in this paper will be useful for evaluating the security of block ciphers against impossible differentials, especially when one tries to design a word-oriented block cipher with bijective S-boxes.

Key words: word-oriented block ciphers, truncated impossible differentials, difference propagation system, $\mathcal{U}$-method, UID-method

1 Introduction

Impossible differential cryptanalysis is one of the most popular cryptanalytic tools for block ciphers. It was firstly proposed by Knudsen to analyze DEAL [2] in 1998 and then extended by Biham *et al.* to attack IDEA [3] and Skipjack [4]. As a variant of differential cryptanalysis, impossible differential cryptanalysis follows the idea of difference but uses differentials with probability zero to achieve the right key by discarding all wrong keys. Until now, impossible differential cryptanalysis has shown its superiority against differential cryptanalysis in many block ciphers such as IDEA, Skipjack, CLEFIA [5] and AES [1].

The success of impossible differential cryptanalysis is mainly based on the length of impossible differentials. The longer the impossible differential is, the better the attack will be. Another important factor is the pattern of impossible differentials, because the attack may be improved using new impossible differentials[11–13]. Therefore, it's meaningful to find them as many as possible.

Impossible differentials are usually constructed by a miss-in-the-middle approach, whose basic idea is, first, to construct two probability-one differentials from encryption direction and decryption direction respectively, and then to demonstrate some contradictions when combining them together. Once the existence of these impossible combinations is proven, attackers obtain an impossible differential, which can be used to retrieve subkeys from the outer rounds of a block cipher. In this key recovering step, many techniques have developed to improve the attack, including precomputation tables [22], the *early abort technique* [19], using multiple impossible differentials [18] and reduplicate key bits in the key schedule.

However, there are few techniques for finding impossible differentials of practical block ciphers. Until now, most of them are constructed manually by experience and intuition, or by carefully observing some properties of inner primitives [20, 21]. We may miss some better impossible differentials in these cases because manual methods are inefficient and unsystematic.

In this paper, we focus on automatic approaches of finding the longest truncated impossible differentials for word-oriented block ciphers with bijective S-boxes. This class of block ciphers are worth to be considered because they are so popular that many famous ciphers are included in it, such as the AES, CLEFIA and E2 [16].

Related Work. In [4], Biham *et al.* described an automatic technique named *shrinking* for finding global impossible differentials. Instead of the original block cipher, *Shrinking* technique considers a shrunken variant of this cipher but preserves its global structure. Then, for a given plaintext difference, *Shrinking* technique encrypts many pairs of plaintexts under various keys and selects the cipher differences that never occur to construct impossible differentials. The problem is that impossible differentials obtained by this technique may fail with some probabilities.

Other works considered the block cipher structures. In [9], Kim *et al.* proposed the $\mathcal{U}$-method to find the longest impossible differentials for various block cipher structures with bijective round functions. In the $\mathcal{U}$-method, the propagation of differences in a structure is translated to simple matrix operations. And some inconsistent conditions are adopted to identify impossible differentials. Luo *et al.* [10] developed the idea of $\mathcal{U}$-method and proposed a more general method — UID-method. The UID-method releases some limitations in the $\mathcal{U}$-method and harnesses more inconsistent conditions to judge impossible differentials.

Although the $\mathcal{U}$-method and the UID-method are determinant, the results achieved by them may not be used directly in analyzing practical ciphers. Because the length of impossible differentials for the underlying structure may not be the longest. For example, Wu *et al.* [11] found that Camellia without FL and FL^{-1} layers has 8 round impossible differentials while its underlying structure—Feistel structure — only contributes a 5 round impossible differential. Another problem is that they may fail to detect some kinds of impossible differentials when they are directly applied to the practical block ciphers because much information is chucked by them in the process of finding impossible differentials.

Our Tool and Contributions. In this work, we show a powerful tool for automatically searching the longest truncated impossible differentials on word-oriented block ciphers. Unlike the miss-in-the-middle approach, which splits a block cipher into two parts, we treat it as an entirety. The input of our tool is a system of equations that describes the differential propagation of a block cipher and some constraints on

Table 1. Summary of new truncated impossible differentials (ID) obtained by our tool. Camellia* is the Camellia without FL and FL^{-1} layers.

Block Cipher	Word unit	Previous results		In this paper		
		Round	No. of IDs	Round	No. of IDs	New IDs
AES	byte	4([22–26])	269,554	4	3,608,100	3,338,546
ARIA	byte	4([11–13, 21])	156	4	94,416	94,260
Camellia*	byte	8([11])	3	8	4	1
E2	byte	6([20])	1	6	56	55
MIBS	nibble	8([15])	2	8	8	6
LBlock	nibble	14([14])	1	14	80	79
Piccolo	nibble	7 ([17])	1(unshown)	7	450	449

the plaintext and ciphertext differences. Then, identifying an impossible differential is equivalent to solving this system and showing that it doesn't have any solution.

Experimental results indicate that our tool is indeed efficient and systematic. It independently rediscovers all known truncated impossible differentials on nine round CLEFIA [18]. What's more, it finds new and longest truncated impossible differentials for many other word-oriented block ciphers, such as the AES, Camellia [6] without FL and FL^{-1} layers, MIBS [7], LBlock [14], ARIA [8], E2 and Piccolo [17]. The number of new truncated impossible differentials obtained by our tool are summarized in table 1.

Finally, we compare the ability of our tool with the $\mathcal{U}$-method and UID-method by illustrating an 14-round impossible differentials of LBlock. Results show that our tool is more powerful than the $\mathcal{U}$-method and UID-method.

Outline of this paper. In section 2, we introduce definitions of χ-function and word-oriented block ciphers. In section 3, we describe how to build a system of equations for finding truncated impossible differentials and discuss how to solve this system. Our tool is proposed in section 4. In section 5, we apply our tool to automatically search the longest truncated impossible differentials for various word-oriented block ciphers. The comparison of our tool with the $\mathcal{U}$-method and UID-method is given in section 6. Finally, we conclude this paper.

2 Preliminaries

In this section, we briefly introduce the definition of χ-function and word-oriented block ciphers. Note that we consider the exclusive-or (i.e., $\oplus$) difference in the whole paper.

2.1 χ-Function

In this subsection, we first introduce the definition of χ-function and then discuss some basic properties of χ-function.

Definition 1. *(χ-function) Let χ be the function $\mathbb{F}_{2^t} \rightarrow \mathbb{F}_2$ defined as follows.*

$$\chi(x) = \begin{cases} 1 & \textit{if } x \neq 0, \\ 0 & \textit{if } x = 0. \end{cases}$$

Then, we have

Property 1. Suppose $\mathcal{S} : \mathbb{F}_{2^t} \to \mathbb{F}_{2^t}$ is a bijective S-box, $\Delta\alpha$ is the input difference of $\mathcal{S}$ while $\Delta\beta$ is the output difference of $\mathcal{S}$, then

$$\chi(\Delta\alpha) \oplus \chi(\Delta\beta) = 0.$$

Property 2. Suppose $a \cdot \Delta\alpha \oplus b \cdot \Delta\beta = 0$, where $a, b \in \mathbb{F}_{2^t} \setminus \{0\}$ and $\Delta\alpha, \Delta\beta \in \mathbb{F}_{2^t}$, then

$$\chi(\Delta\alpha) \oplus \chi(\Delta\beta) = 0.$$

The χ-function is suitable for truncated differential cryptanalysis, when we are only concerned about whether there is a difference or not while the concrete value of the difference is not taken into consideration.

2.2 Word-oriented Block Ciphers

A word-oriented block cipher is such a block cipher that (a) its input and output data is a vector of words with fixed size, and (b) all inner operations consist only of calculations of fixed size. In this work, we limit our attention to the class of word-oriented block ciphers with bijective S-boxes. In such block ciphers, bijective S-boxes are the only nonlinear primitives. Thus, it is natural to study wordwise differentials as truncated differentials.

This class of block ciphers are worth considering since they are so popular that many block ciphers are included in it. For instance, the AES, CLEFIA, E2, Camellia without FL and FL^{-1} layers and ARIA are such block ciphers with word length 8-bit (i.e., byte) while the MIBS, LBlock and Piccolo are some lightweight block ciphers with word length 4-bit (i.e., nibble).

3 Difference Propagation System

For any given block cipher $\mathcal{E}$, we always can represent it as a multivariate system $\Phi(P, C, K, V) = 0$ of plaintext P, the corresponding ciphertext C, subkeys K and some intermediate variables V. However, in finding impossible differentials, what we are concerned about is not the concrete values of P, C, K and V but the differences of them. So, we restrict our attention to built a system of equations that describes the propagation behavior when differences pass through the inner primitives of $\mathcal{E}$. This system will be called *difference propagation system* in the subsequent discussions.

We denote by $\Phi'(\Delta P, \Delta C, \Delta V) = 0$ the difference propagation system of $\mathcal{E}$, where ΔP, ΔC and ΔV are the differences of plaintext P, ciphertext C and intermediate variables V respectively. Notice that K vanishes in this system since subkey differences are zero. Then, we have

Proposition 1. *$\Delta P \to \Delta C$ is impossible if system*

$$\begin{cases} \Phi'(\Delta P, \Delta C, \Delta V) = 0, \\ \Psi(\Delta P, \Delta C) = 0, \end{cases} \tag{1}$$

doesn't have any solution, where $\Psi(\Delta P, \Delta C) = 0$ contains some initial constraints on the plaintext difference ΔP and ciphertext difference ΔC.

In the remainder of this section, we first discuss how to built a difference propagation system for finding truncated impossible differentials and then describe how to solve system (1) and decide whether there is any solution.

3.1 Built Difference Propagation Systems

Suppose $\mathcal{E}$ is an lt-bit word-oriented block cipher with bijective S-boxes, where t is the word length. And ΔP and ΔC are the plaintext difference and ciphertext difference of $\mathcal{E}$ respectively. They can be expressed as vectors, i.e., $\Delta P = (\Delta P_1, \Delta P_2, \ldots, \Delta P_l)$ and $\Delta C = (\Delta C_1, \Delta C_2, \ldots, \Delta C_l)$, where ΔP_j, $\Delta C_j \in \mathbb{F}_{2^t}$ for $1 \leq j \leq l$.

We denote by ΔX_i and ΔY_i the input difference and output difference of round i, respectively. For each bijective S-box in round i, we introduce two intermediate variables, $\Delta U_{i,j}$ and $\Delta W_{i,j}$, to represent its input difference and output difference. After introducing these intermediate variables, we can express the differential propagation of round i as a system of linear equations with variables ΔX_i, ΔY_i, $\Delta U_{i,j}$ and $\Delta W_{i,j}$[1], since bijective S-boxes are the only non-linear primitives in $\mathcal{E}$ while other primitives are linear (or affine) functions. The role of introducing two intermediate variables before and after the S-boxes is equivalent to linearizing the system. More precisely, we can describe the propagation behavior of a primitive as follows.

Proposition 2. *Suppose a linear (or affine) primitive $f : \mathbb{F}_{2^t}^m \to \mathbb{F}_{2^t}^n$ of $\mathcal{E}$ is given, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ is the input data and $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$ is the output data of f, respectively. That is,*

$$\beta_j = \bigoplus_{i=1}^{m} a_{i,j} \cdot \alpha_i \oplus c_j, \tag{2}$$

for $1 \leq j \leq n$, where $a_{i,j}$, $c_j \in \mathbb{F}_{2^t}$ are constants. Then, we can built n linear equations

$$\Delta\beta_j = \bigoplus_{i=1}^{m} a_{i,j} \cdot \Delta\alpha_i, \text{ for } 1 \leq j \leq n. \tag{3}$$

For a further step, we have

Proposition 3. *Suppose $\mathcal{S} : \mathbb{F}_{2^t} \to \mathbb{F}_{2^t}$ is a bijective S-box, that is, $W = \mathcal{S}(U)$, then we can built a nonlinear equation*

$$\chi(\Delta U) \oplus \chi(\Delta W) = 0 \tag{4}$$

from property 1.

Utilizing proposition 2 and proposition 3, we can translate the propagation of differences in $\mathcal{E}$ to a system of equations. For the convenience of comprehension, we employ r-round AES as an example to show how to generate a difference propagation systems over $\mathbb{F}_{2^8}$. Similar systems can be constructed for other word-oriented block ciphers with bijective S-boxes (See section 5).

Example 1. The AES is a byte-oriented block cipher with bijective S-boxes over $\mathbb{F}_{2^8}$. An AES round has four consecutive layers: SubBytes, ShiftRows, MixColumns and SubkeyAddition. And the final round doesn't have the MixColumns layer. We needn't consider the SubkeyAddition layer here since it doesn't influence the propagation of differences. Additionally, we may even omit the last ShiftRows layer in finding impossible differentials.

[1] The variables may be changed when we build difference propagation system for a concrete block cipher.

We denote by ΔX_i the internal state entering round i (*i.e.*, befor SubBytes) while ΔY_i denotes the internal state after the SubBytes layer. Each state can be represented by a 4×4 matrix over $\mathbb{F}_{2^8}$. In such a matrix, every byte in row j and column k is numbered as byte $j + 4 \cdot (k-1)$, that is, byte one is in the top-left corner, the first column is made of bytes 1-4, while the last column is made of bytes 13-16, with byte 16 in the bottom-right corner.

Then, $r \times 16 = 16 \cdot r$ nonlinear equations for the SubBytes layers are:

$$\chi(\Delta X_{i,l}) \oplus \chi(\Delta Y_{i,l}) = 0 \text{ for } l = 1, 2, \ldots, 16 \text{ and } i = 1, 2, \ldots, r. \tag{5}$$

And $(r-1) \times 16 = 16 \cdot (r-1)$ linear equations for the ShiftRows and MixColumns layers are summarized as follows:

$$\Delta X_{i+1} \oplus MC \times \begin{pmatrix} \Delta Y_{i,1} & \Delta Y_{i,5} & \Delta Y_{i,9} & \Delta Y_{i,13} \\ \Delta Y_{i,6} & \Delta Y_{i,10} & \Delta Y_{i,14} & \Delta Y_{i,2} \\ \Delta Y_{i,11} & \Delta Y_{i,15} & \Delta Y_{i,3} & \Delta Y_{i,7} \\ \Delta Y_{i,16} & \Delta Y_{i,4} & \Delta Y_{i,8} & \Delta Y_{i,12} \end{pmatrix} = 0 \text{ for } i = 1, 2, \ldots, r-1,$$

where MC denotes the MixColumns matrix. In this system, ΔX_1 is the plaintext difference and ΔY_r is the ciphertext difference.

3.2 Solve Difference Propagation Systems and Finding Impossible Differentials

For a difference propagation system Φ', we can divide it into two subsystems — $\mathcal{L}$ and $\mathcal{NL}$. $\mathcal{L}$ includes all linear equations while $\mathcal{NL}$ contains all nonlinear equations from bijective S-boxes. We denote by Λ_0 the set of all variables with zero difference while Λ_1 the set of all variables with nonzero difference.

Now, we need to decide whether $\Delta P \to \Delta C$ is impossible under initial constraints $\Psi(\Delta P, \Delta C) = 0$. Since we consider truncated differentials in this paper, we focus on initial constraints with form $V_P = (\chi(\Delta P_1), \chi(\Delta P_2), \ldots, \chi(\Delta P_l))$ and $V_C = (\chi(\Delta C_1), \chi(\Delta C_2), \ldots, \chi(\Delta C_l))$. Notice that V_P and V_C are nonzero vectors.

Firstly, for $1 \leq j \leq l$, we add ΔP_j to Λ_0 and a linear equation $\Delta P_j = 0$ to system $\mathcal{L}$ if $\chi(\Delta P_j) = 0$; Else, add ΔP_j to Λ_1. Similar operations can be done on ΔC.

Then, we solve the system (1). What we mean "solve" here is trying to predict the information of unknown variables from known ones. Notice that every prediction we made should be deterministic, that is, with probability one. This can be done in the following two ways:

(i) Solving the linear system $\mathcal{L}$. It can be solved by *Gauss-Jordan Elimination* algorithm, which gets solutions by reducing the augmented matrix of $\mathcal{L}$ to reduced row echelon form using elementary row operations. If a variable is zero, then it is added to Λ_0; Else, if a variable is nonzero, then it is added to Λ_1.

(ii) "Solving" the nonlinear system $\mathcal{NL}$. From proposition 3, we know $\Delta U \in \Lambda_1$ as soon as we have $\Delta W \in \Lambda_1$ and vice versa, which provides an extra bit information for us. What's more, if we know one of them is zero, then another is zero too, which provides an extra linear equation. In this case, we add a linear equation $\Delta U = 0$ (or $\Delta W = 0$) to linear system $\mathcal{L}$ and add an element ΔU (or ΔW) to Λ_0.

The basic strategy for predicting information is clear now: solving system $\mathcal{L}$ and $\mathcal{NL}$ alternately until a contradiction is found or we cannot get new information any longer. An impossible differential is detected by the following proposition:

Proposition 4. *$\Delta P \to \Delta C$ is impossible if one of the following two situations happens*

- **I.** *Linear system $\mathcal{L}$ doesn't have any solution. That is, the rank of its coefficient matrix is unequal to the rank of its augmented matrix.*
- **II.** *$\Lambda_0 \cap \Lambda_1 \neq \emptyset$. It's a contradiction because an element $e \in \Lambda_0$ means $\chi(e) = 0$ while $e \in \Lambda_1$ means $\chi(e) = 1$.*

4 Algorithm to Find the Longest Truncated Impossible Differential

In this section, we discuss how to implement a difference propagation system on a computer and propose a specific algorithm to automatically search the longest truncated impossible differentials.

4.1 Implementation of a Difference Propagation System

As mentioned above, a difference propagation system Φ' is divided into two parts — systems $\mathcal{L}$ and $\mathcal{NL}$. In this subsection, we discuss how to implement them on a computer.

Implement system $\mathcal{L}$ with a matrix. System $\mathcal{L}$ can be written formally as $A \cdot x = b$, where A, b and $B = [A|b]$ are called the *coefficient matrix*, *constant matrix* and *augment matrix* of this system respectively, x is the set of *all* variables (in order) involved in the difference propagation system Φ'. Each column vector of A corresponds to a unique variable in x. Thus, matrix A is determined once the order of variables of x is fixed.

In computer manipulations, we don't deal with the system in terms of equations but instead make use of the augmented matrix. The *Gauss-Jordan Elimination* algorithm is used to solve this system.

Implement system $\mathcal{NL}$ with a table. Once the order of variables in x is given, we can store equations in system $\mathcal{NL}$ with a simple table using the column indexes of B. First, initialize an empty table $\mathcal{T}$, then for each equation in $\mathcal{NL}$, add an element $\{p, q\}$ to $\mathcal{T}$, where p, q are two integers (i.e., the column indexes of B) indicating the two variables involved in the equation. For instance, in example 1, if the order of variables is fixed as $[\Delta X_1, \Delta Y_1, \ldots, \Delta X_r, \Delta Y_r]$, then table $\mathcal{T}$ for equations in (5) is $\mathcal{T} = [\{32(j-1)+i, 32(j-1)+16+i\} : 1 \leq i \leq 16, 1 \leq j \leq r]$.

4.2 Automatically Find Truncated Impossible Differentials

Suppose we obtain the difference propagation system of a word-oriented block cipher, with form of a matrix B and a table $\mathcal{T}$. $p = [p_1, \ldots, p_l]$ and $c = [c_1, \ldots, c_l]$ are two vectors, where p_i and c_i $(1 \leq i \leq l)$ are the corresponding column indexes of ΔP_i and ΔC_i in B, respectively. Let $R_n^{(i)}$ be a $1 \times n$ matrix whose i-th component is 1 while other components are 0, where n is the number of columns of matrix B, and $ColSubMatrix(B, i, j)$ be the submatrix of B with columns from i to j.

After choosing the initial constraints V_P and V_C, our algorithm takes the matrix B, table $\mathcal{T}$, vectors p, c, V_P and V_C as inputs, and then solves the system as described in section 3.2 automatically. Finally, it outputs a $flag$ indicating whether $\Delta P \rightarrow \Delta C$ is impossible. The specific description of our algorithm is shown in algorithm 1. For an r-round word-oriented block cipher, we can obtain all r-round truncated impossible differentials by enumerating all possible nonzero V_P and V_C. To obtain the longest truncated impossible differential, it only needs to try several round numbers.

Algorithm Complexity. Suppose B' is a matrix with m rows and n columns. The time consumption for computing $rank(B')$ and *Gauss-Jordan Elimination* is less than $O(N^3)$, where $N = \max\{m, n\}$. And the time consumption for subprogram **Predict_Info** is about $O(N^2)$. Algorithm 1 terminates if $index$ is false. Otherwise, at least one of sets Λ_0 and Λ_1 is updated after each while loop. According to the pigeon-hole principle, there must be a type **II** contradiction mentioned in proposition 4 when $|\Lambda_0| + |\Lambda_1| > n$. Therefor, while loop runs $n + 1$ times at most. In summary, the time complexity of deciding a truncated impossible differential doesn't exceed $O(N^4)$.

Remark 1. (Validity of our tool with more kinds of constraints.) Besides V_P and V_C, some linear constraints between nonzero variables in ΔP and ΔC can also be added in the choosing initial constraints step. Our tool still works in this case by translating them to row matrixes firstly and then adding these rows to matrix B'.

Remark 2. (Reuse the matrix B and table $\mathcal{T}$.) Since the encryption process of a block cipher is deterministic, we need only to build the difference propagation system once. So, matrix B and table $\mathcal{T}$ can be reused for different V_P and V_C.

5 Results and Applications

We apply our tool to find the longest truncated impossible differentials for various byte-(or nibble-)oriented block ciphers, such as the AES, CLEFIA, E2, Camellia without FL and FL^{-1} layers, ARIA, LBlock, MIBS and Piccolo. We may classify them into three groups according to their underlying structures: the SP structure (AES and ARIA), Feistel structure (Camellia without FL and FL^{-1} layers, LBlock, MIBS and E2), and generalized feistel structure (CLEFIA and Piccolo). Figure 1 shows these structures. Notice that the operation of adding subkeys is exclusive or in these block ciphers. Therefore, we needn't consider them since they don't influence the differential propagation. The results of this section are obtained under a 2.66GHz processor with MAGMA package [27].

5.1 Block Ciphers with SP Structure

For an r round block cipher with SP structure (See Fig.1), we denote by ΔX_i the internal state entering round i (*i.e.*, before the S-box layer) while ΔY_i denotes the internal state after the S-box layer. Each state is a row vector with length l. As mentioned above, we omit the subkey addition layer since it doesn't influence the

```
Input: Matrix $B$, table $\mathcal{T}$, vector $p$ and $c$, $V_P$ and $V_C$.
Output: A $flag$ indicates whether $\Delta P \rightarrow \Delta C$ is impossible.
// Initialize the constraints $V_P$ and $V_C$.
$B' := B$; $\Lambda_0 := \emptyset$; $\Lambda_1 := \emptyset$; $n :=$NumberOfColumns$(B')$;
for $i := 1$ to $l$ do
    $\Lambda_1 := \Lambda_1 \cup \{p_i\}$ if $V_p[i] = 1$;
    $\Lambda_0 := \Lambda_0 \cup \{p_i\}$ and $B' :=$AddRow$(B', R_n^{(p_i)})$ if $V_p[i] = 0$;
    $\Lambda_1 := \Lambda_1 \cup \{c_i\}$ if $V_c[i] = 1$;
    $\Lambda_0 := \Lambda_0 \cup \{c_i\}$ and $B' :=$AddRow$(B', R_n^{(c_i)})$ if $V_c[i] = 0$;
// Predict information and find contradictions.
$flag :=$ false; $index :=$true;
while $index$ do
    $A' :=$ColSubMatrix$(B', 1, n-1)$;
    if $rank(A') \neq rank(B')$ then
        $flag :=$ true;
    else
        // Gauss-Jordan Elimination.
        $B' :=$Reduced-row-echelon-form-of$(B')$;
        $< B', \Lambda_0, \Lambda_1, temp >:=$ Predict_Info$(B', \mathcal{T}, \Lambda_0, \Lambda_1)$;
        $index := temp$;
        if $\Lambda_0 \cap \Lambda_1 \neq \emptyset$ then
            $flag :=$ true; $index :=$false;
return $flag$.

Predict_Info$(B', \mathcal{T}, \Lambda_0, \Lambda_1)$;
$n_0 := |\Lambda_0|$; $n_1 := |\Lambda_1|$; $temp :=$false;
// Extract information from the reduced row echelon form of $B'$.
for $i := 1$ to $NumberOfRows(B')$ do
    $S := \emptyset$;
    for $j := 1$ to $n-1$ do
        $S := S \cup \{j\}$ if $B'[i,j] \neq 0$;
    // Update $\Lambda_0$ using definition 1.
    if $|S| = 1$ and $B'[i,n] = 0$ then
        $\Lambda_0 := \Lambda_0 \cup S$;
    // Update $\Lambda_1$ using definition 1 or property 2.
    if $(|S| = 1$ and $B'[i,n] \neq 0)$ or $(|S| = 2, B'[i,n] = 0$ and $S \cap \Lambda_1 \neq \emptyset)$ then
        $\Lambda_1 := \Lambda_1 \cup S$;
// Scan table $\mathcal{T}$ and use proposition 3 to recover information.
for $j := 1$ to $NumberOfElements(\mathcal{T})$ do
    if $\mathcal{T}[j] \cap \Lambda_0 \neq \emptyset$ and $\mathcal{T}[j] \setminus \Lambda_0 \neq \emptyset$ then
        $B' :=$AddRow$(B', R_n^{(e)})$ for $e \in \mathcal{T}[j] \setminus \Lambda_0$;
        $\Lambda_0 := \Lambda_0 \cup \mathcal{T}[j]$;
    if $\mathcal{T}[j] \cap \Lambda_1 \neq \emptyset$ then
        $\Lambda_1 := \Lambda_1 \cup \mathcal{T}[j]$;
if $|\Lambda_0| > n_0$ or $|\Lambda_1| > n_1$ then
    $temp :=$true;
return $< B', \Lambda_0, \Lambda_1, temp >$.
```

Algorithm 1: Automatically judge a truncated impossible differential.

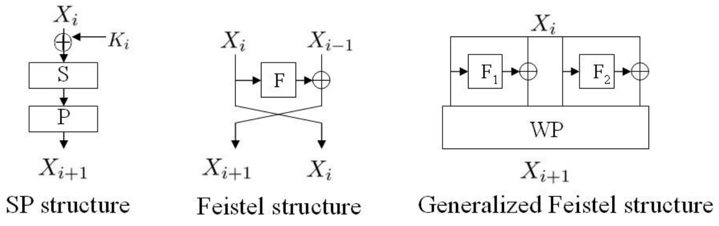

Fig. 1. Underlying structures of block ciphers.

differential propagation. Additionally, we omit the last linear permutation layer. Then, we can built the difference propagation system as follows:

$$\begin{aligned} \chi(\Delta X_{i,j}) \oplus \chi(\Delta Y_{i,j}) &= 0, \\ \Delta X_{k+1}^T \oplus P \cdot \Delta Y_k^T &= 0, \end{aligned}$$

where P is the matrix of linear permutation layer, ΔX_{k+1}^T is the transposed vector of ΔX_{k+1} and $1 \leq i \leq r$, $1 \leq j \leq l$ and $1 \leq k \leq r-1$. In this system, ΔX_1 is the plaintext difference and ΔY_r is the ciphertext difference.

The AES and ARIA are word-oriented block ciphers with $l = 16$. For the AES, our tool finds 3,608,100 of four round truncated impossible differentials in hours by using a variant of our tool (See Appendix A). The classes of impossible differentials discussed in [22–26] are included in our result set. For the ARIA, besides the results shown in [11–13], our tool finds 94,260 new 4-round truncated impossible differentials in several days.

We observe that all impossible differentials of AES follow an inherent rule. We can describe them explicitly (See Appendix A). However, there is no such rule for ARIA (We don't introduce them in this paper due to lack of space).

5.2 Block Ciphers with the Feistel Structure

For an r round block cipher with Feistel structure (See Fig.1), we denote by ΔX_{i-1} the state of right branch and ΔX_i the state of left branch entering round i. Thus, $(\Delta X_1, \Delta X_0)$ is the plaintext difference and $(\Delta X_{r+1}, \Delta X_r)$ is the ciphertext difference. Each state is a row vector with length $\frac{l}{2}$.

Camellia, MIBS and LBlock are Feistel ciphers with SP-type F-function while E2 is a Feistel cipher with SPS-type F-function. If F is an SP structure, we denote by ΔY_i the state after the S-box layer. Then, we have

$$\begin{aligned} \chi(\Delta X_{i,j}) \oplus \chi(\Delta Y_{i,j}) &= 0, \\ \tau(\Delta X_{i-1})^T \oplus \Delta X_{i+1}^T \oplus P \cdot \Delta Y_i^T &= 0, \end{aligned}$$

where P is the matrix of linear permutation layer in F, $1 \leq i \leq r$ and $1 \leq j \leq \frac{l}{2}$. τ is a 2-nibble left rotation of difference ΔX_{i-1} in LBlock and it's an identical permutation in Camellia and MIBS. A similar system can be built by introducing some new variables to represent the input difference and output difference of the second S-boxes layer if F is an SPS structure.

We apply our tool to search the longest truncated impossible differentials for Camellia without FL and FL^{-1} layers, MIBS, LBlock and E2. Since there is always

Table 2. Truncated impossible differentials $(0, \Delta X_1) \nrightarrow_r (\Delta X_{r+1}, 0)$ for some Feistel ciphers. Previous results are marked with bold type. Camellia* is the Camellia without FL and FL^{-1} layers.

Ciphers	r	No.	Truncated Impossible Differentials
Camellia*	8	4	$(\mathbf{e_1}, \mathbf{e_1})$, $(\mathbf{e_2}, \mathbf{e_2})$, $(\mathbf{e_3}, \mathbf{e_3})$[11], (e_4, e_4).
MIBS	8	6	$(\mathbf{e_3}, \mathbf{e_8})$, $(\mathbf{e_7}, \mathbf{e_5})$ [15], (e_3, e_5), (e_5, e_3), (e_5, e_7) , (e_8, e_3).
LBlock	14	80	(e_i, e_j), for $1 \leq i, j \leq 8$, where $(\mathbf{e_3}, \mathbf{e_2})$ is given in [14]. $(e_1, e_{\{4,6\}})$, $(e_2, e_{\{2,8\}})$, $(e_3, e_{\{5,7\}})$, $(e_4, e_{\{5,7\}})$, $(e_5, e_{\{2,8\}})$, $(e_6, e_{\{4,6\}})$, $(e_7, e_{\{1,3\}})$, $(e_8, e_{\{1,3\}})$, $(e_{\{1,2\}}, e_1)$, $(e_{\{3,8\}}, e_2)$, $(e_{\{5,6\}}, e_3)$, $(e_{\{3,8\}}, e_4)$, $(e_{\{5,6\}}, e_5)$, $(e_{\{4,7\}}, e_6)$, $(e_{\{1,2\}}, e_7)$, $(e_{\{4,7\}}, e_8)$.
E2	6	56	(e_1, e_1), $(\mathbf{e_1}, \mathbf{e_3})$[20], (e_1, e_5), (e_1, e_6), (e_1, e_8), (e_2, e_5), (e_2, e_6), (e_2, e_7), (e_2, e_8), (e_3, e_1), (e_3, e_3), (e_3, e_6), (e_3, e_8), (e_4, e_5), (e_4, e_6), (e_4, e_8), (e_5, e_1), (e_5, e_2), (e_5, e_3), (e_5, e_4), (e_5, e_5), (e_5, e_6), (e_5, e_8), (e_6, e_1), (e_6, e_2), (e_6, e_3), (e_6, e_4), (e_6, e_5), (e_6, e_6), (e_6, e_7), (e_7, e_2), (e_7, e_6), (e_7, e_7), (e_7, e_8), (e_8, e_1), (e_8, e_2), (e_8, e_3), (e_8, e_4), (e_8, e_5), (e_8, e_7), $(e_5, e_{\{2,7\}})$, $(e_5, e_{\{4,7\}})$, $(e_6, e_{\{1,8\}})$, $(e_6, e_{\{3,8\}})$, $(e_7, e_{\{2,5\}})$, $(e_7, e_{\{5,8\}})$, $(e_8, e_{\{1,6\}})$, $(e_8, e_{\{3,6\}})$, $(e_{\{2,7\}}, e_5)$, $(e_{\{4,7\}}, e_5)$, $(e_{\{1,8\}}, e_6)$, $(e_{\{3,8\}}, e_6)$, $(e_{\{2,5\}}, e_7)$, $(e_{\{5,8\}}, e_7)$, $(e_{\{1,6\}}, e_8)$, $(e_{\{3,6\}}, e_8)$.

a five round impossible differential $(0, \Delta X_0) \nrightarrow_5 (\Delta X_6, 0)$ for the Feistel structure with bijective round function when $\Delta X_0 = \Delta X_6$, we focus on finding impossible differentials $(0, \Delta X_0) \nrightarrow_r (\Delta X_{r+1}, 0)$ with $r \geq 6$. Results for these ciphers can be obtained in an hour.

Suppose I is a set, e_I denotes a vector whose i-th (for all $i \in I$) component is nonzero while other components are zero. Then, the results obtained by our tool are illustrated in table 2, where (e_I, e_J) is the shortened form of $(0, \Delta X_1) \nrightarrow_r (\Delta X_{r+1}, 0)$. For example, $(e_{\{5,6\}}, e_5)$ in the results of LBlock means

$$(00000000, 0000\alpha_1\alpha_2 00) \nrightarrow_{14} (0000\beta 000, 00000000), \tag{6}$$

where α_1, α_2 and β are nonzero 4-bit differences.

5.3 Block Ciphers with the Generalized Feistel Structure

CLEFIA is a generalized Feistel cipher with SP-type F-functions. And Piccolo has a variant of generalized Feistel structure with an SPS-type F-function. The underlying structures of them is shown in Fig.1. Both of them have $l = 16$.

We denote by ΔX_i the input difference of round i and ΔZ_i the internal state before WP layer, where WP is a word permutation. Additionally, we can introduce some variables to represent the internal state of F function as we have done in the Feistel ciphers. ΔX_1 is the plaintext difference while ΔZ_r is the ciphertext difference since there is not the WP layer in the final round.

In this paper, we focus on finding truncated impossible differentials with form $\Delta X_1 = \Delta Z_r = (0000, ????, 0000, ????)$, where "?" can be zero or nonzero difference. The number of truncated differentials for ΔX_1 and ΔZ_r is reduced to 2^{16} now. Our computer enumerates them in an hour. For the CLEFIA, our tool independently rediscovers all known truncated impossible differentials listed in [18] but doesn't find new results. In [17], the designers claimed that they find a 7-round impossible differential using modified $\mathcal{U}$-method. However, the specification is not shown. With the application of our tool, we find 450 truncated impossible differentials for 7-round Piccolo. They are distributed in two classes

$$(0000, 00\alpha_1\alpha_2, 0000, \alpha_3\alpha_4 00) \not\rightarrow_7 (0000, \beta_1\beta_2 00, 0000, 00\beta_3\beta_4), \tag{7}$$
$$(0000, \alpha_1\alpha_2 00, 0000, 00\alpha_3\alpha_4) \not\rightarrow_7 (0000, 00\beta_1\beta_2, 0000, \beta_3\beta_4 00), \tag{8}$$

where each $\chi(\alpha_i), \chi(\beta_i) \in \{0, 1\}$ for $1 \leq i \leq 4$ and at least one of α_i and β_i is nonzero.

6 Comparison of Our Tool with the $\mathcal{U}$-method and UID-method

In this section, we investigate the relationship of our tool with the $\mathcal{U}$-method and UID-method.

On the one hand, block cipher structures discussed in the $\mathcal{U}$-method and UID-method can be treated as special word-oriented block ciphers with bijective S-boxes, which means they also can be disposed by our tool.

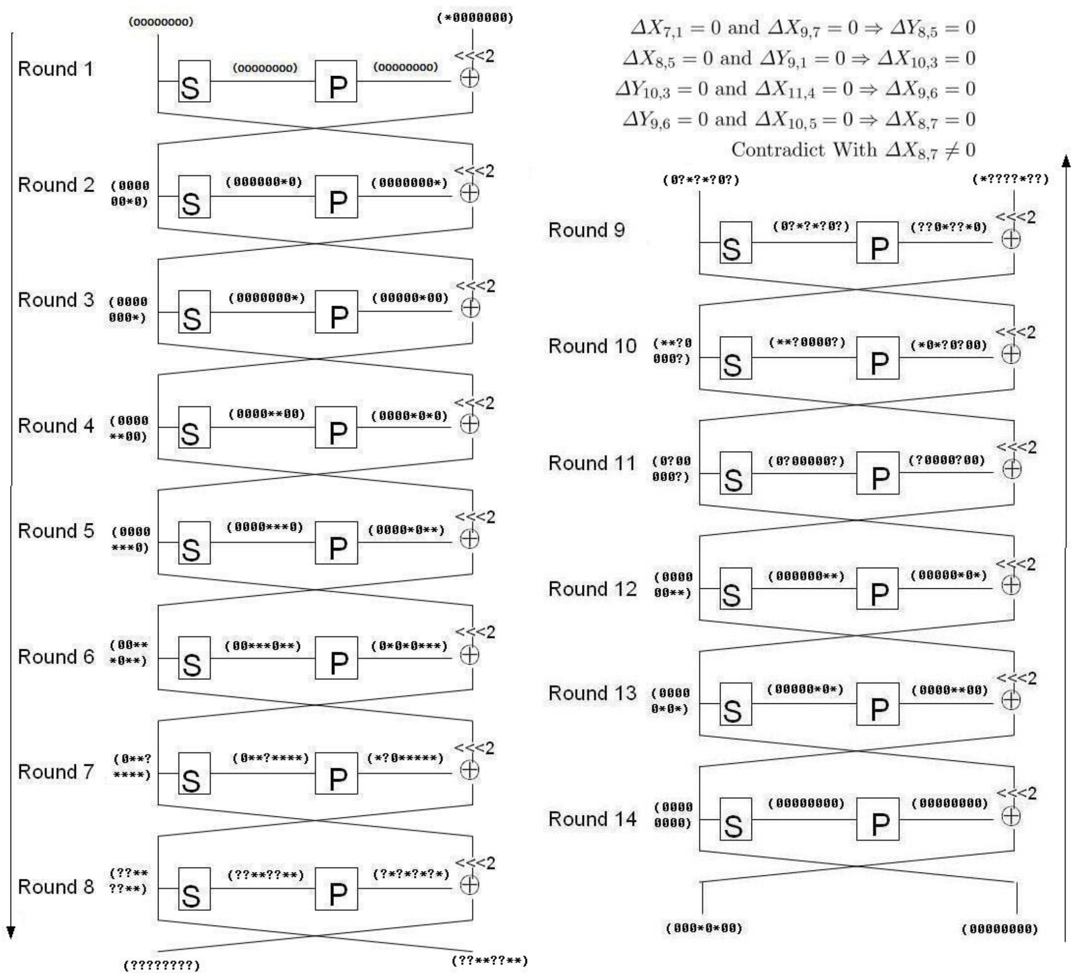

Fig. 2. A truncated impossible differential of 14-round LBlock.

On the other hand, our tool can detect more kinds of impossible differentials. The example we chosen is the 14-round impossible differential $(e_1, e_{\{4,6\}})$ of LBlock (see Table 2). To see it precisely, a step-by-step deduction following by our algorithm is illustrated below in Fig.2, where "0", "*" and "?" are represented a nibble with zero difference, nonzero difference and undetermined difference, respectively.

In Round i of LBlock, the input and output difference of the S-box layer is denoted as $\Delta X_i = (\Delta X_{i,1}\Delta X_{i,2}\cdots\Delta X_{i,8})$ and $\Delta Y_i = (\Delta Y_{i,1}\Delta Y_{i,2}\cdots\Delta Y_{i,8})$, respectively. P is a nibble permutation, that is,

$$\begin{aligned} &P(\Delta Y_{i,1}\Delta Y_{i,2}\Delta Y_{i,3}\Delta Y_{i,4}\Delta Y_{i,5}\Delta Y_{i,6}\Delta Y_{i,7}\Delta Y_{i,8}) \\ =&(\Delta Y_{i,2}\Delta Y_{i,4}\Delta Y_{i,1}\Delta Y_{i,3}\Delta Y_{i,6}\Delta Y_{i,8}\Delta Y_{i,5}\Delta Y_{i,7}). \end{aligned}$$

And "$\lll 2$" means left rotation with two nibbles (8 bits). Thus, we have

$$\begin{aligned} &(\Delta Y_{i,2}\Delta Y_{i,4}\Delta Y_{i,1}\Delta Y_{i,3}\Delta Y_{i,6}\Delta Y_{i,8}\Delta Y_{i,5}\Delta Y_{i,7}) \\ =&(\Delta X_{i+1,1}\Delta X_{i+1,2}\Delta X_{i+1,3}\Delta X_{i+1,4}\Delta X_{i+1,5}\Delta X_{i+1,6}\Delta X_{i+1,7}\Delta X_{i+1,8}) \\ \oplus&(\Delta X_{i-1,3}\Delta X_{i-1,4}\Delta X_{i-1,5}\Delta X_{i-1,6}\Delta X_{i-1,7}\Delta X_{i-1,8}\Delta X_{i-1,1}\Delta X_{i-1,2}) \end{aligned} \tag{9}$$

for $1 \le i \le 14$.

Now, for given $(\Delta X_1, \Delta X_0) = (00000000, *0000000)$ and $(\Delta X_{15}, \Delta X_{14}) = (000*0*00, 00000000)$, we can deduce the internal state of LBlock round by round. After 8 rounds deduction in the forward direction, we obtain that the output difference of round 8 is

$$(\Delta X_9, \Delta X_8) = (????????, ??**??**). \tag{10}$$

While in the backward direction, after 6 round deduction, we obtain that the input difference of round 9 is

$$(\Delta X_9, \Delta X_8) = (0?*?*?0?, *????*??). \tag{11}$$

It seems that it's not an impossible differential when we combine (10) and (11) together, because the symbol "?" can be any value. In fact, the progress is terminated in this step when using the $\mathcal{U}$-method and UID-method.

However, our algorithm still works. From equation (11), we know $\Delta X_{9,1}$ and $\Delta X_{9,7}$ is zero, thus $\Delta Y_{9,1}$ is zero. From Fig.2, we find $\Delta X_{7,1}$ is zero, which implies $\Delta Y_{8,5} = 0$ using equation (9). Thus $\Delta X_{8,5}$ is zero. Now, $\Delta Y_{9,1} = 0$ and $\Delta X_{8,5} = 0$ implies $\Delta X_{10,3} = 0$, which means $\Delta Y_{10,3}$ is zero. Next, from $\Delta Y_{10,3} = 0$ and $\Delta X_{11,4} = 0$ (See Fig.2), we can deduce that $\Delta X_{9,6} = 0$, which means $\Delta Y_{9,6}$ is zero. Finally, from $\Delta Y_{9,6} = 0$ and $\Delta X_{10,5} = 0$ (See Fig.2), we conclude $\Delta X_{8,7}$ is zero, which contradicts with $\Delta X_{8,7} \neq 0$ in equation (10). Therefore, $(e_1, e_{\{4,6\}})$ is indeed a 14-round truncated impossible differential of LBlock.

7 Conclusions and Future Work

In this paper, we focus on automatic tool of finding the longest truncated impossible differentials for word-oriented block ciphers with bijective S-boxes. The block cipher structures studied in the $\mathcal{U}$-method and UID-method are specific cases in this paper.

What's more, our tool is more powerful than them. With the application of our tool, we rediscover all truncated impossible differentials for nine round CLEFIA. And we find new and longest truncated impossible differentials for many byte-(and nibble-)oriented block ciphers.

Since the conditions of judging an impossible differential are sufficient conditions in this paper, results obtained by our tool must be correct if one implements our algorithm and block ciphers on computer correctly. On the other hand, our tool is efficient since it costs polynomial time to judge an truncated impossible differential.

We except that the tool proposed in this article, especially the idea of building and solving a difference propagation system, is not only helpful for evaluating the security of block ciphers against impossible differential cryptanalysis, but also useful in other attacks.

Future work will include the application of difference propagation system in other attacks and the improvement of impossible differential cryptanalysis on block ciphers using new results obtained by our tool.

References

1. Daemen, J., Rijmen, V.: The Design of Rijndael: AES – The Advanced Encryption Standard. Springer-Verlag (2002)
2. Knudsen, L.R.: DEAL—A 128-bit block cipher. Technical Report 151, Department of Informatics, University of Bergen, Bergen, Norway (1998)
3. Biham, E., Biryukov, A., Shamir, A.: Miss in the middle attacks on IDEA, Khufu and Khafre. In: Knudsen, L.(ed.) FSE'99. LNCS, vol.1636, pp.124–138. Springer, Heidelberg(1999)
4. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 12–23. Springer, Heidelberg(1999)
5. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit blockcipher CLEFIA (Extended abstract). In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 181–195. Springer, Heidelberg (2007)
6. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: a 128-bit block cipher suitable for multiple platforms — design and analysis. In: Stinson, D.R., Tavares, S.E. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39– 56. Springer, Heidelberg (2001)
7. Izadi, M.I., Sadeghiyan, B., Sadeghian, S.S., Khanooki, H.A.: MIBS: a new lightweight Block Cipher. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 334–348. Springer, Heidelberg (2009)
8. Kwon, D., Kim, J., Park, S., et al.: New Block Cipher: ARIA. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 432–445. Springer, Heidelberg (2004)
9. Kim, J., Hong, S., Sung, J., Lee, C., Lee, S.: Impossible differential cryptanalysis for block cipher structures. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 82–96. Springer, Heidelberg (2003)
10. Luo, Y., Wu, Z., Lai, X., and Gong, G.: 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.
11. W. Wu, W. Zhang and D. Deng. Impossible Diffrential Cryptanalysis of ARIA and Camellia. In: Journal of Computer Science and Technology. Vol. 22, Num. 3, pp. 449–456 (2007)
12. Li, S., Song, C.: Improved Impossible Differential Cryptanalysis of ARIA. In: ISA 2008. pp. 129-132. IEEE Computer Society, Los Alamitos (April 2008)
13. 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)
14. Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G.(eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)

15. Bay, A., Nakahara Jr, J., Vaudenay, S.: Cryptanalysis of Reduced-Round MIBS Block Cipher. In: Heng, S.-H., Wright, R.N., Goi, B.-M.(eds.) CANS 2010. LNCS, vol. 6467, pp. 1–19. Springer, Heidelberg (2010)
16. Kanda, M., Moriai, S., Aoki, K., Ueda, H., Takashima, Y., Ohta, K., Matsumoto, T.: E2 — A new 128-bit block cipher. In: IEICE Transactions Fundamentals – Special Section on Cryptography and Information Security E83–A (1), pp. 48–59 (2000)
17. Shibutani,K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An Ultra-Lightweight Blockcipher. In: Preneel, B., Takagi, T.(eds.) CHES 2011. LNCS, vol. 6917, pp. 342C357. Springer, Heidelberg(2011)
18. Tsunoo, Y., Tsujihara, E., Shigeri, M., Suzaki, T., Kawabata, T.: Cryptanalysis of CLEFIA Using Multiple Impossible Differentials. In: International Symposium on Information Theory and its Applications, ISITA2008, pp. 1–6 (2008)
19. Lu, J., Kim, J., Keller, N., Dunkelman, O.: Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 370–386. Springer, Heidelberg (2008)
20. Wei, Y., Li, P., Sun, B., Li, C.: Impossible Differential Cryptanalysis on Feistel Ciphers with SP and SPS Round Functions. In: Zhou, J., Yung, M. (eds.) ACNS 2010. LNCS, vol. 6123, pp. 105C122. Springer, Heidelberg (2010)
21. Li,R., Sun, B., Li, C.: Impossible differential cryptanalysis of SPN ciphers. In: IET Information Security. Vol. 5, Iss. 2, pp. 111–120 (2011)
22. Biham, E., Keller, N.: Cryptanalysis of Reduced Variants of Rijndael. In: The Third AES Candidate Conference (2000)
23. Phan, R.C.-W., Siddiqi, M.U.: Generalised Impossible Differentials of Advanced Encryption Standard. Electronics Letters 37(14). pp. 896–898 (2001)
24. Phan, R.C.-W.: Classes of Impossible Differentials of Advanced Encryption Standard. Electronics Letters 38(11). pp. 508–510 (2002)
25. Bahrak, B., Aref, M.R.: Impossible differential attack on seven-round AES-128. IET Information Security 2, 28–32 (2008)
26. Mala, H., Dakhilalian, M., Rijmen, V., Modarres, M.-H.,: Improved Impossible Differential Cryptanalysis of 7-Round AES-128. In: Gong, G., Gupta, K.C. (eds.): INDOCRYPT 2010. LNCS 6498, pp. 282–291, Springer, Heidelberg (2010)
27. Bosma, W., Cannon, J., Playoust, C.: The MAGMA Algebra System I: The User Language. In Journal of Symbolic Computation, Vol. 24, Issues 3-4, pp. 235-265 (1997)

A All Impossible Differentials of 4-round AES

In this section, we describe all 4-round impossible differentials of AES obtained by our tool. We use the same notation introduced in example 1, that is, ΔX_1 is the plaintext difference and ΔY_4 is the ciphertext difference.

For given nonzero ΔX_1 and ΔY_4, we denote by $I = \{j : \Delta X_{1,j} \neq 0 \text{ for } 1 \leq j \leq 16\}$ and $J = \{j : \Delta Y_{4,j} \neq 0 \text{ for } 1 \leq j \leq 16\}$. Suppose $a_1 = \{1, 6, 11, 16\}$, $a_2 = \{2, 7, 12, 13\}$, $a_3 = \{3, 8, 9, 14\}$, $a_4 = \{4, 5, 10, 15\}$ and $b_i = \{4{\cdot}i{-}3, 4{\cdot}i{-}2, 4{\cdot}i{-}1, 4{\cdot}i\}$ for $1 \leq i \leq 4$. We denote by $H(I)$ the number of true events that $I \cap a_i \neq \emptyset$ for $1 \leq i \leq 4$. For example, $H(I) = 2$ if $I = \{1, 2\}$ because $I \cap a_1 \neq \emptyset$ and $I \cap a_2 \neq \emptyset$ but $I \cap a_3 = \emptyset$ and $I \cap a_4 = \emptyset$. Similarly, we denote by $H(J)$ the number of true events that $I \cap b_i \neq \emptyset$ for $1 \leq i \leq 4$. Then, we can search all four round truncated impossible differentials as follows.

1. We set truncated values for only a part of elements in ΔX_1 and ΔY_4 while let other elements undetermined.
2. Use Algorithm 1 to predict the information of undetermined elements in ΔX_1 and ΔY_4.
3. Construct impossible differentials using conditions that contradicts with the information obtained in step 2.

For example, if we choose set a_1 and b_1 and initialize $\Delta X_1 = (?0000?0000?0000?)$ and $\Delta Y_4 = (????000000000000)$ in step 1. That is, only 12 elements of ΔX_1 and ΔY_4 is determined. Then, using Algorithm 1, we find $\Delta X_{1,j}$ for $j \in a_1$ and $\Delta Y_{4,j}$ for $j \in b_1$ are zero. That is, from the conditions in step 1, the undetermined elements in ΔX_1 and ΔY_4 is determined now. Thus, we conclude that any $(\chi(\Delta X_{1,1}), \chi(\Delta X_{1,6}), \chi(\Delta X_{1,11}), \chi(\Delta X_{1,16})) \neq (0000)$ to any $(\chi(\Delta Y_{4,1}), \chi(\Delta Y_{4,2}), \chi(\Delta Y_{4,3}), \chi(\Delta Y_{4,4})) \neq (0000)$ is impossible, which implies that we obtain $15 \times 15 = 225$ truncated impossible differentials simultaneously. Other set of impossible differentials can be obtained by changing the choice in step 1. Finally, we have

Proposition 5. *All 3,608,100 four round truncated impossible differentials of AES can be summarized as* $H(I) + H(J) \leq 4$.

Since $H(I) = 1$ provides $\binom{4}{1} \times (2^4 - 1) = 60$ choices, $H(I) = 2$ provides $\binom{4}{2} \times (2^4 - 1)^2 = 1350$ choices, and $H(I) = 3$ provides $\binom{4}{3} \times (2^4 - 1)^3 = 13500$ choices, the number of $H(I) + H(J) \leq 4$ is

$$60 \times (60 + 1350 + 13500) + 1350 \times (60 + 1350) + 13500 \times 60 = 3,608,100.$$

Impossible differentials discussed in [22–26] are subsets of our generalized results shown in proposition 5. They include 269,554 impossible differentials in total.

What should be stressed here is that the method used in this section is highly relevant to the structure properties of AES. Thus, it's difficult to extend it to other block ciphers in this paper.