Impossible Differential Cryptanalysis of 3D Block Cipher

Mohsen Shakiba, Mohammad Dakhilalian, and Hamid Mala

***Abstract*—3D is a secret-key block cipher, designed to secure and fast encryption of large amounts of data. This block cipher uses multi-dimensional states to generalize the design of Rijndael. Thus, while maintaining the benefits of the AES design, 3D operates on 512-bit blocks of data and can also be used as a cryptographic primitive in the cryptographic systems with the large internal states. Since its proposal in 2008, the cryptanalysis of 3D has been considered in several papers. While the previous impossible differential attacks on 3D cipher can analyze up to 10 rounds of the cipher, this paper, using a new 6-round impossible differential, presents an impossible differential attack on 11 rounds of 3D. The proposed distinguisher begins in the input of AddRoundKey operation of round 3, and ends in the output of ShiftRows of round 8. Results show that the proposed attack on 11-round of 3D cipher requires about 2^{501} chosen plaintexts and a time complexity of about 2^{495} 11-round encryptions.**

***Index Terms*—Block ciphers, Cryptanalysis, Impossible Differential, Symmetric cryptography**

I. Introduction

The block cipher 3D is an AES-based block cipher proposed by Nakahara at CANS 2008 [1]. This block cipher operates on 512-bit blocks and supports a 512-bit secret-key. Inspired from the design of AES [2], the main round transformations of 3D are basically the same as those of AES, while it operates on larger blocks of data and a larger key size. In fact, 3D cipher puts four AES states in parallel and applies the diffusion of AES in two different directions in every two rounds in turn. Security of 3D block cipher has been considered through several cryptanalysis methods including multiset [1], [3], impossible differential [1], [3]-[4], truncated differential [5] and square attack [6]. Results of key recovery attacks on 3D block cipher are summarized in Table I. According to this table, the best known attack on 3D is a truncated differential attack which can be mounted on 13 rounds of it [5]. Two known-key distinguishers on 9.75-round and 15-round 3D have been also proposed in [3] and [7], respectively.

In this paper we focus on impossible differential cryptanalysis of 3D. Impossible differential cryptanalysis, an extension of the differential attack [8], was first introduced by Knudsen [9] and Biham [10] to analyze DEAL and Skipjack, respectively. This kind of attack uses differentials that hold with probability zero to derive the right key by discarding the wrong keys which lead to the impossible differential. This cryptanalysis technique has achieved considerable results on AES [11]-[14]. Also, for block cipher Camellia [15], which has been approved by NESSIE and the Japanese CRYPTREC projects, the best cryptanalytic results are obtained by the impossible differential attacks [16]-[18].

TABLE I
RESULTS OF KEY-RECOVERY ATTACKS ON 3D BLOCK CIPHER

Rounds #	Time (Encryptions)	Data	Memory	Success Rate	Ref	Attack Type
5.75	2^{139}	2^{129} CP	2^{128}	1	[1]	Multiset
7, 8, 9	2^{133}, 2^{189}, 2^{414}	?	?	?	[6]	Square
5.75	$2^{65.6}$	2^{36} CP	2^{32}	1	[1]	Imp. Diff
9	2^{478}	2^{445} CP	?	1	[4]	Imp. Diff
10	2^{401}	2^{501} CP	2^{311}	1	[3]	Imp. Diff
11	2^{288}	2^{251} CP	2^{128}	0.24	[5]	Trunc. Diff
11	2^{113}	2^{252} CP	2^{128}	0.0034	[5]	Trunc. Diff
13	2^{308}	2^{470} CP	2^{128}	≈1	[5]	Trunc. Diff
11	2^{495}	$2^{500.5}$ CP	2^{381}	1	This Paper	Imp. Diff

As it is seen in Table I, previous impossible differential analyses of 3D cipher are applicable up to 10 rounds [1], [3]-[4]. In this paper, using a new 6-round impossible differential (ID), we propose an impossible differential cryptanalysis of 11 rounds of 3D cipher. This attack requires about $2^{500.5}$ chosen plaintexts and 2^{505} memory accesses which is equivalent to about 2^{495} 11-round encryptions. Also, the attack needs about 2^{381} bytes of memory to store the intermediate values and the precomputations.

The rest of this paper is organized as follow. Section II provides a brief description of the 3D cipher. Section III introduces a new 6-round ID distinguisher of 3D. A new key recovery attack on 11 rounds of 3D is described in Section IV. Finally, the paper is concluded in Section V.

II. Preliminaries and a Brief Description of 3D

The block cipher 3D is a 22-round SPN block cipher with 512-bit block length and 512-bit key. Each state in the cipher is composed of 64 bytes $(a_0, a_1, \ldots, a_{63})$ which are ordered column-wise as follows:

Manuscript received August 28, 2016; accepted November 27, 2016.
(Corresponding Author) Mohsen Shakiba is with the Department of Electrical and Computer Engineering, Jundi-Shapur University of Technology, Dezful, Iran (e-mail: m.shakiba@jsu.ac.ir).

Mohammad Dakhilalian is with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, 84156-83111, Iran (e-mail: mdalian@cc.iut.ac.ir).

Hamid Mala is with the Department of Information Technology, Faculty of Computer Engineering, University of Isfahan, 81746-73441, Hezar Jerib Avenue, Isfahan, Iran (e-mail: h.mala@eng.ui.ac.ir).

$$A=\left(\begin{array}{cccc|cccc|cccc|cccc} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_1 & a_5 & a_9 & a_{13} & a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} \\ a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} & a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} \\ a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} & a_{51} & a_{55} & a_{59} & a_{63} \end{array}\right)$$

Moreover, a key scheduling algorithm generates 22 rounds of 512-bit subkeys, each with the same order and structure as the state. Like AES, each round of 3D consists of four transformations, each one is considered as a fraction of 0.25 of one round. Using the terminology of [1], these transformations are as follow:

- κ_i: bit-wise XOR with the 512-bit subkey of i-th round (k_i), equivalent to the *AddRoundKey* operation in the AES.
- γ: a byte-wise S-box layer with the 8-bit S-box of AES, equivalent to the *SubBytes* operation in the AES.
- θ_1, θ_2: two different byte transpositions equivalent to the *ShiftRows* operation of AES, applied in two different directions, alternately. θ_1 and θ_2 are applied in odd-numbered rounds and even-numbered rounds according to the following permutations, respectively:

$$\theta_1:\left(\begin{array}{cccc|cccc|cccc|cccc} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_5 & a_9 & a_{13} & a_1 & a_{21} & a_{25} & a_{29} & a_{17} & a_{37} & a_{41} & a_{45} & a_{33} & a_{53} & a_{57} & a_{61} & a_{49} \\ a_{10} & a_{14} & a_2 & a_6 & a_{26} & a_{30} & a_{18} & a_{22} & a_{42} & a_{46} & a_{34} & a_{38} & a_{58} & a_{62} & a_{50} & a_{54} \\ a_{15} & a_3 & a_7 & a_{11} & a_{31} & a_{19} & a_{23} & a_{27} & a_{47} & a_{35} & a_{39} & a_{43} & a_{63} & a_{51} & a_{55} & a_{59} \end{array}\right)$$

$$\theta_2:\left(\begin{array}{cccc|cccc|cccc|cccc} a_0 & a_4 & a_8 & a_{12} & a_{16} & a_{20} & a_{24} & a_{28} & a_{32} & a_{36} & a_{40} & a_{44} & a_{48} & a_{52} & a_{56} & a_{60} \\ a_{17} & a_{21} & a_{25} & a_{29} & a_{33} & a_{37} & a_{41} & a_{45} & a_{49} & a_{53} & a_{57} & a_{61} & a_1 & a_5 & a_9 & a_{13} \\ a_{34} & a_{38} & a_{42} & a_{46} & a_{50} & a_{54} & a_{58} & a_{62} & a_2 & a_6 & a_{10} & a_{14} & a_{18} & a_{22} & a_{26} & a_{30} \\ a_{51} & a_{55} & a_{59} & a_{63} & a_3 & a_7 & a_{11} & a_{15} & a_{19} & a_{23} & a_{27} & a_{31} & a_{35} & a_{39} & a_{43} & a_{47} \end{array}\right)$$

As it is discussed in [19], these two permutations could be unified in a single byte permutation with the same diffusion property.

- π: a 4×4 matrix multiplication is applied to columns of the state, equivalent to the *MixColoumns* operation of AES.

The i-th round of an r-round 3D cipher $(0 \le i \le r-1)$ is denoted by $\pi \circ \theta_{(i \bmod 2)+1} \circ \gamma \circ \kappa_i(X)$, where X is the round input state. In the last round, the π operation is replaced by an additional *AddRoundKey* (round subkey k_r). We refer to [1] for more details of 3D structure. Because *AddRoundKey* and MixColumns operations are both linear, it is possible to replace them by each other. In such a case, we have an equivalent round subkey $k^{eq} = \pi(k)$.

Notations. The specific bytes are indicated by brackets; for example bytes 2 and 8 of the round subkey k_i are indicated by $k_{i,[2,8]}$. x_i^{γ}, x_i^{θ}, x_i^{π} and x_i^{κ} denote the intermediate values after the application of *SubBytes*, *ShiftRows*, *MixColoumns* and *AddRoundKey* operations of round *i*, respectively. Moreover, $x_i^{\kappa(eq)}$ is used to denote the *AddRoundKey* operation with the equivalent round subkey k_i^{eq}.

III. New 6-round Impossible Differential of 3D

The impossible differential attack presented in this paper is based on a new 6-round impossible differential. As illustrated in Fig. 1, this distinguisher begins in the input of AddRoundKey operation of round 3 with a difference which is nonzero in any desired three bytes of the 16-th column and is zero in the other 61 bytes. So, there exist four different types of the input difference. Fig. 1 shows how one of these input differences leads to a difference in the output of round 5 which is zero in one slice and nonzero in the others. On the other hand, in the decryption direction, the distinguisher starts in the output of ShiftRows of round 8 with a difference which is nonzero in only one byte of the 16th column and zero in the other 63 bytes. Fig. 1 shows one of the four possible output differences leading to a difference which is nonzero in all 64 bytes in the output of round 5. This contradicts the output difference of the differential in the encryption direction.

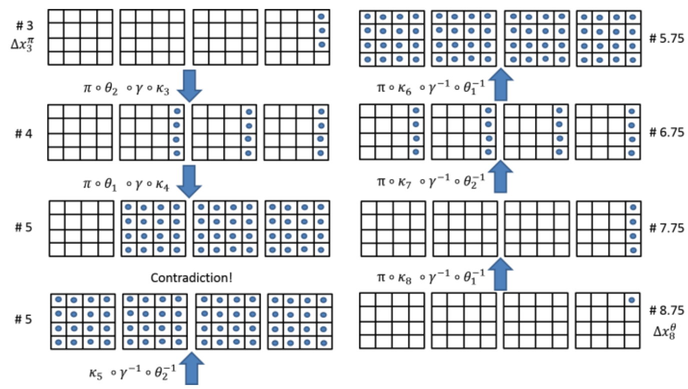

Fig. 1. The new 6-round impossible differential of 3D cipher.

IV. Impossible Differential Attack on 11 Rounds of 3D

For mounting the impossible differential attack on 3D cipher, three rounds are added to the beginning of the 6-round distinguisher and two rounds to the end of it. Fig. 2 and Fig. 3 illustrate these added rounds, respectively. As it can be seen, 62 bytes of subkeys are involved in this attack and the ultimate goal is to recover the correct value of these subkeys. The attack scenario is composed of two phases. At the first stage some tables are pre-computed and then the online stage begins.

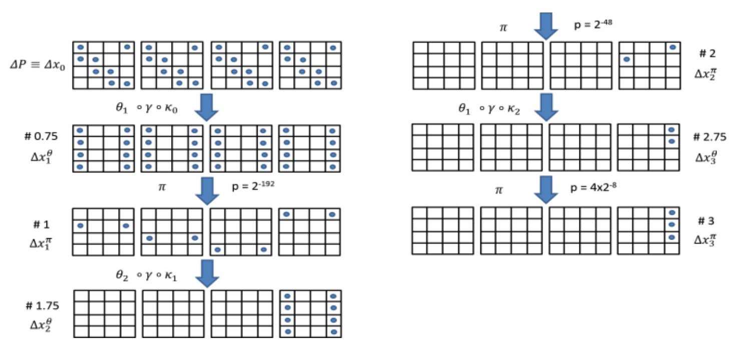

Fig. 2. Three additional rounds before the distinguisher

An issue, which must be considered in the online stage, is the amount of required proper plaintext/ciphertext pairs. Proper pairs are the plaintext/ciphertext pairs which satisfy the input difference ΔP and output difference ΔC as it is indicated in Fig. 2 and 3, respectively. Based on Fig. 2, the probability for a plaintext pair with the difference ΔP to meet the input difference of the distinguisher is about $2^{-192} \times 2^{-48} \times 4 \times 2^{-8} =$

2^{-246}. For a ciphertext pair with the difference ΔC the probability to meet the output difference of the distinguisher is also about $2^{-96} \times 4 \times 2^{-24} = 2^{-118}$. So, for a proper plaintext/ciphertext pair, the probability to meet the impossible differential is about $2^{-246} \times 2^{-118} = 2^{-364}$. For a specific value of the 62-byte target subkey, if a proper plaintext/ciphertext pair meets the impossible differential, then the value of subkey is wrong and must be eliminated from the key space. Therefore, using 2^n proper pairs, the probability for a wrong key not to be eliminated is about $\left(1-2^{-364}\right)^{2^n}$. So, using 2^n proper pairs, about $2^{496} \times \left(1-2^{-364}\right)^{2^n}$ wrong 62-byte subkeys remain in the key space. For $n = 372.43$, there will remain only about one wrong subkey (in addition to the correct subkey which is not eliminated). In the two upcoming subsections, at first, some required precomputations are described and then we will illustrate the online attack procedure.

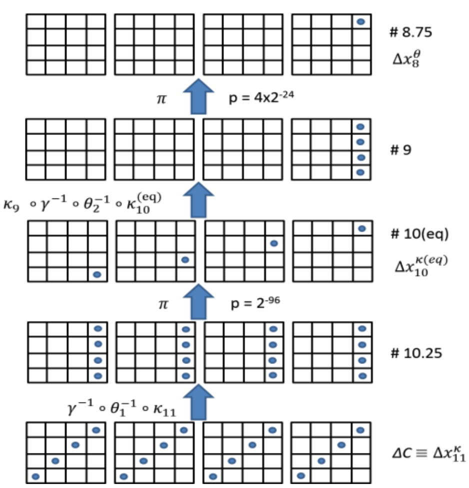

Fig. 3. Two additional rounds after the distinguisher

A. *Precomputations*

In this section, we prepare three tables H_1, H_2 and T to reduce the amount of partial encryption/decryptions in the online stage of the attack.

- H_1: For all of the $2^{8\times 3} \times \left(2^8 \times \left(2^8 - 1\right)\right) \approx 2^{40}$ possible pairs of $\left(x^{\pi}_{2,[48,49,50,51]}, x'^{\pi}_{2,[48,49,50,51]}\right)$ which have non-zero difference only in the byte 49, perform a partial decryption to compute the values of pairs $\left(x^{\kappa}_{2,[48,1,18,35]}, x'^{\kappa}_{2,[48,1,18,35]}\right)$. Store the obtained pairs in a hash table H_1 indexed by their difference $x^{\kappa}_{2,[48,1,18,35]} \oplus x'^{\kappa}_{2,[48,1,18,35]}$. Such a table has 2^{32} rows and on average $2^{40}/2^{32} = 2^8$ pairs lie in each row. Clearly, for an intermediate pair $\left(x^{\pi}_{1,[48,1,18,35]}, x'^{\pi}_{1,[48,1,18,35]}\right)$ it is sufficient to access the row indexed by $x^{\pi}_{1,[48,1,18,35]} \oplus x'^{\pi}_{1,[48,1,18,35]}$ in H_1 to obtain on average 2^8 values for $k_{1,[48,1,18,35]}$.

H_2: For all of the $2^{8\times 3} \times \left(2^8 \times \left(2^8 - 1\right)\right) \approx 2^{40}$ possible pairs of $\left(x^{\pi}_{2,[60,61,62,63]}, x'^{\pi}_{2,[60,61,62,63]}\right)$ which have non-zero difference only in byte 60, compute the values of pairs $\left(x^{\kappa}_{2,[60,13,30,47]}, x'^{\kappa}_{2,[60,13,30,47]}\right)$. Store the obtained pairs in a hash table H_2 indexed by their difference $x^{\kappa}_{2,[60,13,30,47]} \oplus x'^{\kappa}_{2,[60,13,30,47]}$. Such a table has 2^{32} rows and on average $2^{40}/2^{32} = 2^8$ pairs lie in each row. So, for an intermediate pair $\left(x^{\pi}_{1,[60,13,30,47]}, x'^{\pi}_{1,[60,13,30,47]}\right)$ it is sufficient to access the row indexed by $x^{\pi}_{1,[60,13,30,47]} \oplus x'^{\pi}_{1,[60,13,30,47]}$ in H_2 to obtain on average 2^8 values for $k_{1,[60,13,30,47]}$. Note that we can also use table H_2 to obtain on average 2^8 values of $k_{0,[60,49,54,59]}$ for each plaintext pair of $\left(x_{0,[60,49,54,59]}, x'_{0,[60,49,54,59]}\right)$ which take this pair to a difference $\Delta x^{\pi}_{1,[60,61,62,63]}$ with only one non-zero byte in location 60.

-T: For all of the 2^{32} possible pairs of $\left(x^{\kappa}_{3,[60,49]}, x'^{\kappa}_{3,[60,49]}\right)$ with non-zero difference, perform a partial encryption through $\pi \circ \theta_1 \circ \gamma$ to compute the difference value $\Delta x^{\pi}_{3,[60,61,62,63]}$. Now, if this difference is non-zero exactly in three bytes, then store the pair in table T indexed by $x^{\kappa}_{3,[60,49]} \oplus x'^{\kappa}_{3,[60,49]}$. The probability to meet this condition is about $4 \times 2^{-8} = 2^{-6}$, so about $2^{32-6}/2^{16} = 2^{10}$ pairs lie in each of the 2^{16} rows of table T. Clearly, for an intermediate pair $\left(x^{\pi}_{2,[60,49]}, x'^{\pi}_{2,[60,49]}\right)$ it is sufficient to access the row of T with index $x^{\pi}_{2,[60,49]} \oplus x'^{\pi}_{2,[60,49]}$ to obtain on average 2^{10} values for $k_{2,[60,49]}$.

The overall computations for preparing these tables are less than 2^{41} partial encryptions. Further, the required memory for these tables is about 2^{43}, 2^{43} and 2^{28} bytes for the tables H_1, H_2 and T, respectively.

B. *The online stage of the attack procedure*

After preparing the hash tables, the attack is carried out by the following steps. In the first step of the attack the required proper plaintext/ciphertext pairs are prepared. Then, attack proceeds by removing wrong values of involved 62-byte subkeys from the key space until only the right value of the 62-byte subkey remains. For reducing the time complexity, we use the early aborting technique [16] as well as the precomputed tables H_1, H_2 and T. The overall required computations for each step (with respect to the 11-round 3D encryptions) are indicated in the end of that step. Also, in the following time complexities, the coefficient α equals to $2 \times (1/11) \times (1/16) \approx 2^{-6.46}$.

- *Step* 1. Take $2^{373.43}$ structures of ciphertexts such that each structure contains about 2^{128} ciphertexts that are fixed in the 48 bytes of ΔC with zero difference indicated in Fig. 3 and take all the possible values in other 16 bytes. So, about $2^{128} \times 2^{128}/2 = 2^{255}$ ciphertext pairs are obtained from each structure which their difference is coincident to the required ΔC. Obtain the corresponding plaintexts of each structure in a chosen ciphertext scenario. Since the probability of having a plaintext pair with the difference ΔP in Fig. 2 is 2^{-256}, then for each structure we can collect about $2^{255} \times 2^{-256} = 2^{-1}$ plaintext pairs with the difference ΔP. Hence, after examining all of the $2^{373.43}$ structures, we can collect about $2^n = 2^{373.43} \times 2^{-1} = 2^{372.43}$ distinct ciphertext/plaintext pairs satisfying the desired ΔC and ΔP, respectively. As it was discussed in the beginning of Section IV, this amount of data is sufficient to recover the correct value of 62-byte subkey. The time complexity of this step, which corresponds to the data complexity of the attack, is about $2^{128} \times 2^{373.43} = 2^{501.43}$ 11-Round 3D encryptions. We also need $4 \times 2^{372.43} \times 64 = 2^{380.43}$ bytes of memory to store the proper ciphertext/plaintext pairs.

- *Steps* (2, 3, 4, 5). For i = 0,1,2,3 do the following steps 2, 3, 4 and 5 sequentially:

Guess 32 bits of $k_{11,[3+16i,6+16i,9+16i,12+16i]}$ and for all of the 2^{n-24i} proper ciphertext pairs $\left(x^{\kappa}_{11,[3+16i,6+16i,9+16i,12+16i]}, x'^{\kappa}_{11,[3+16i,6+16i,9+16i,12+16i]}\right)$, perform a partial decryption to obtain the pairs $\left(x^{\kappa(eq)}_{10,[12+16i,13+16i,14+16i,15+16i]}, x'^{\kappa(eq)}_{10,[12+16i,13+16i,14+16i,15+16i]}\right)$. If for a proper pair, the difference of the obtained pairs is non-zero only in the byte $\Delta x^{\kappa(eq)}_{10,[15+16i]}$, then keep this proper pair for the next step. The probability of this condition is about 2^{-24}, so there remain about $2^{n-24(i+1)}$ proper pairs for the next step. The time complexity of this step is about $\alpha \times 2^{n-24i+32(i+1)} = \alpha \times 2^{n+32+8i}$ encryptions.

- *Step* 6. Now, we have the intermediate remaining pair values $\left(x^{\kappa(eq)}_{10,[15,30,45,60]}, x'^{\kappa(eq)}_{10,[15,30,45,60]}\right)$. Guess 32 bits of $k^{eq}_{10,[15,30,45,60]}$ and for all of the 2^{n-96} remaining pairs $\left(x^{\kappa(eq)}_{10,[15,30,45,60]}, x'^{\kappa(eq)}_{10,[15,30,45,60]}\right)$ perform a partial decryption to obtain the difference value $\Delta x^{\theta}_{8,[60,61,62,63]}$. If for a pair, this difference is nonzero in only one byte, then we have met the desired difference in the output of the distinguisher in Fig. 1, so keep such a pair for the next step. The probability of this condition is about 4×2^{-24}, so it is expected to remain about 2^{n-118} proper ciphertext pairs which meet the output difference of the distinguisher. This step requires about $\alpha \times 2^{(n-96)+160} = \alpha \times 2^{n+64}$ encryptions.

- *Steps* (7, 8, 9, 10). For all of the 2^{n-118} remaining ciphertext pairs take their corresponding plaintext pairs. Then for i = 0,1,2,3 do the following steps 7, 8, 9 and 10 sequentially:

Guess four bytes $k_{0,[16i,5+16i,10+16i,15+16i]}$ and for all of the $2^{n-118-24i}$ remaining proper plaintext pairs $\left(x_{0,[16i,5+16i,10+16i,15+16i]}, x'_{0,[16i,5+16i,10+16i,15+16i]}\right)$ perform a partial encryption to obtain pairs $\left(x^{\pi}_{1,[16i,1+16i,2+16i,3+16i]}, x'^{\pi}_{1,[16i,1+16i,2+16i,3+16i]}\right)$. If for a pair, the difference value of $\Delta x^{\pi}_{1,[16i+((i+1)\bmod 4)]}$ is nonzero and the difference of the other three bytes of $(1+4i)$-th column of Δx^{π}_{1} are zero, then store this pair for the next step. The probability of this condition is about 2^{-24}, so about $2^{n-118-24(i+1)}$ proper pairs remain for the next step. This step requires about $\alpha \times 2^{(n-118-24i)+160+32(i+1)} = \alpha \times 2^{n+74+8i}$ encryptions.

- *Steps* (11, 12, 13). So far, we have guessed 288 bits of subkeys and obtained about 2^{n-214} remaining pairs from previous steps. For i = 0, 1, 2 do the following steps 11, 12 and 13 sequentially:

Guess four bytes $k_{0,[1+16i,6+16i,11+16i,12+16i]}$ and for all of the $2^{n-214-24i}$ remaining plaintext pairs $\left(x_{0,[1+16i,6+16i,11+16i,12+16i]}, x'_{0,[1+16i,6+16i,11+16i,12+16i]}\right)$ perform a partial encryption to obtain pairs $\left(x^{\pi}_{1,[12+16i,13+16i,14+16i,15+16i]}, x'^{\pi}_{1,[12+16i,13+16i,14+16i,15+16i]}\right)$. If for a pair, the difference value of $\Delta x^{\pi}_{1,[17i+13]}$ is nonzero and the difference of the other three bytes of $4(i+1)$-th column of Δx^{π}_{1} are zero, then store this pair for the next step. The probability of this condition is about 2^{-24}. So, about $2^{n-214-24(i+1)}$ proper pairs remain for the next step. About $\alpha \times 2^{(n-214-24i)+288+32(i+1)} = \alpha \times 2^{n+106+8i}$ encryptions are performed in this step.

- *Step* 14. So far, we have guessed 384 bits of subkeys and obtained about 2^{n-286} remaining pairs from the previous steps. Prepare a vector U of 2^{112} bits, each corresponds to a possible value of 112 bits $k_{0,[60,49,54,59]} \,|\, k_{1,[48,1,18,35]} \,|\, k_{1,[60,13,30,47]} \,|\, k_{2,[60,49]}$. Now, for each of the 2^{n-286} remaining pairs do the following steps:

- *Step* 14.1. In this step, we have the intermediate pair values $\left(x^{\pi}_{1,[48,1,18,35]}, x'^{\pi}_{1,[48,1,18,35]}\right)$. So, by reference to the row indexed by $x^{\pi}_{1,[48,1,18,35]} \oplus x'^{\pi}_{1,[48,1,18,35]}$ in table H_1, obtain 2^8 values for $k_{1,[48,1,18,35]}$. Then, for these key values make a partial encryption to obtain corresponding 2^8 pair values $\left(x^{\pi}_{2,[49]}, x'^{\pi}_{2,[49]}\right)$. The time complexity of this step is $2^{384+(n-286)+8} = 2^{n+106}$ memory accesses (MA) and $\alpha \times 2^{n+106}$ encryptions.

- *Step* 14.2. We have the plaintext pair values $\left(x_{0,[60,49,54,59]}, x'_{0,[60,49,54,59]}\right)$. So, by reference to the row indexed by $x_{0,[60,49,54,59]} \oplus x'_{0,[60,49,54,59]}$ in table H_2, obtain 2^8 values of $k_{0,[60,49,54,59]}$. Then, for these key values make a partial encryption to obtain the corresponding 2^8 pair values $\left(x^{\pi}_{1,[60]}, x'^{\pi}_{1,[60]}\right)$. This step is performed by $2^{384+(n-286)+8} = 2^{n+106}$ MA and $\alpha \times 2^{n+106}$ encryptions.

- *Step* 14.3. For each of the obtained 2^8 pair values $\left(x^{\pi}_{1,[60]}, x'^{\pi}_{1,[60]}\right)$, we have the intermediate pair values $\left(x^{\pi}_{1,[60,13,30,47]}, x'^{\pi}_{1,[60,13,30,47]}\right)$. So, by reference to the row indexed by $x^{\pi}_{1,[60,13,30,47]} \oplus x'^{\pi}_{1,[60,13,30,47]}$ in table H_2, obtain 2^8 values of $k_{1,[60,13,30,47]}$ and for these key values make a partial encryption to obtain the corresponding 2^8 pair values $\left(x^{\pi}_{2,[60]}, x'^{\pi}_{2,[60]}\right)$. This step requires $2^{384+(n-286)+8+8}=2^{n+114}$ MA and $\alpha\times 2^{n+114}$ encryptions.

- *Step* 14.4. For each of the $(2^8\times 2^8)\times 2^8$ obtained pair values $\left(x^{\pi}_{2,[60,49]}, x'^{\pi}_{2,[60,49]}\right)$, access to the row indexed by $x^{\pi}_{2,[60,49]}\oplus x'^{\pi}_{2,[60,49]}$ in table T to obtain 2^{10} values for $k_{2,[60,49]}$. Then, for each of the $2^8\times 2^8\times 2^8\times 2^{10}=2^{34}$ obtained subkey values $k_{0,[60,49,54,59]}\,|\,k_{1,[48,1,18,35]}\,|\,k_{1,[60,13,30,47]}\,|\,k_{2,[60,49]}$ mark the corresponding bits in the vector U to indicate them as wrong keys. This step requires $2^{384+(n-286)+34}=2^{n+132}$ MA.
- *Step* 15. Check all of the bits of the vector U and if there is a bit that is not marked, you have found a candidate for the correct 496-bit subkey which consists of the corresponding 112-bit value of this unmarked bit along with the current 384-bit guessed subkey. So, store this candidate and continue the procedure. This step requires $2^{384+112}=2^{496}$ MA.

As it is expected, the above procedure proceeds until all of the 2^{384} possible values of 384-bit subkeys are guessed in order. It is expected that eventually, there will remain about 2 candidates for the correct 496-bit target subkey. For each candidate, 256 bits of these 496 bits are the bits of k_0. Finally, for each candidate we perform an exhaustive search for the other 256 bits of k_0 with 2×2^{256} 11-round encryptions.

C. Complexity of the attack

As mentioned in Step 1 of the online stage of the attack, the required data to mount the attack contains about $2^{128}\times 2^{373.43}=2^{501.43}$ chosen plaintexts. The memory complexity consists of $4\times 2^{372.43}\times 64=2^{380.43}$ bytes of memory to store the proper ciphertext/plaintext pairs, 2^{112} bits to store the U vector in step 14, and about 2^{44} bytes for the precomputed tables. Also, note that the required memory for storing the intermediate pairs is far less than $2^{380.43}$ bytes of memory, hence the total memory complexity is about 2^{381} bytes of memory.

As mentioned in the online stage of the attack, for $n = 372.43$, the dominant parts of the time complexity include $2^{504.43}$ MA in Step 14-4, 2^{496} MA in Step 15, and $2^{487.97}$ 11-round 3D encryptions in step 13. On the other hand, for each round of 3D, transformations γ and π can be evaluated by 64 and 16 memory accesses, respectively. In a same way, according to the key schedule of 3D, described in [1], each round key k_1 to k_{11} is obtained with 16 + 16 = 32 memory accesses. Thus, the application of 11-round encryptions requires about $(11\times 64+10\times 16)+11\times 32=1216\approx 2^{10.25}$ memory accesses (The complexity of key additions and byte permutations are negligible).

Hence, the total time complexity is equivalent to about $2^{505-10.25}+2^{488}\approx 2^{495}$ 11-round encryptions. However, if we increase the number of remaining candidates from one candidate to about 2^{230} candidates, then, according to the corresponding equality $2^{496}\times\left(1-2^{-364}\right)^{2^n}=2^{230}$, the data complexity decreases to about $2^{500.43}$ chosen plaintexts. Due to this change, the only affected part of time complexity is the complexity of the exhaustive search (after the last step) with $2^{230}\times 2^{256}=2^{486}$ 11-round encryptions, which does not change the overall time complexity of the attack.

V. Conclusion

Security of 3D block cipher has been considered through several cryptanalysis methods. The best known single-key attack on 3D is a truncated differential attack on 13 rounds of this cipher which has been proposed by Takuma et al. [5], with the time complexity of 2^{308} encryptions and data complexity of 2^{470} chosen plaintexts. In this paper, we focused on advancing the impossible differential cryptanalysis of 3D block cipher. While the previous impossible differential attacks on this cipher can analyze up to 10 rounds of the cipher, this paper, using a new 6-round impossible differential, presents an impossible differential attack on 11-round variant of 3D. The proposed attack requires about 2^{501} chosen plaintexts and about 2^{381} bytes of memory. Also, the overall time complexity of the attack is equivalent to about 2^{495} 11-round encryptions of 3D.

References

[1] J. Nakahara, "3D: a Three-Dimensional Block Cipher", CANS 2008, LNCS, 5339, 2008, pp. 252-267.

[2] J. Daemen, V. Rijmen, "The design of Rijndael: AES - the Advanced Encryption Standard", Springer Verlag, 2002.

[3] J. Nakahara, "New Impossible Differential and Known-Key Distinguishers for the 3D Cipher", ISPEC 2011, LNCS, 6672, 2011, pp. 208-221.

[4] T. Xue-hai, L. Chao, W. Mei-yi, Q. Long-jiang, "Impossible Differential Attack on 3D Cipher", Journal of Electronics & Information Technology, 32 (10), 2010, pp. 2516-2520.

[5] K. Takuma, L. Wang, Y. Sasaki, K. Sakiyama, K. Ohta, "New Truncated Differential Cryptanalysis on 3D Block Cipher", ISPEC 2012, LNCS, 7232, 2012, pp. 109-125.

[6] W. Mei-yi, T. Xue-hai, L. Chao, Q. Long-jiang, "Square Attacks on 3D Cipher", Journal of Electronics & Information Technology, 32 (1), 2010, pp. 157-161.

[7] L. Dong, W. Wu, S. Wu, J. Zou, "Known-key distinguisher on round-reduced 3D block cipher", WISA 2011, LNCS, 7115, 2012, p. 55-69.

[8] E. Biham, A. Shamir, "Differential cryptanalysis of DES-like cryptosystems", J. Cryptol., 4 (1), 1991, pp. 3-72.

[9] L.R Knudsen, "DEAL a 128-bit block cipher", Technical report, 1998

[10] E. Biham, A. Biryukov, A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds using impossible differentials", EUROCRYPT ’99, LNCS, 1592, 1999, pp. 12-23.

[11] H. Mala, M. Dakhilalian, V. Rijmen, M. Modarres-Hashemi, "Improved Impossible Differential Cryptanalysis of 7-Round AES-128", INDOCRYPT 2010, LNCS, 6498, 2010, pp. 282-291.

[12] J. Lu, O. Dunkelman, N. Keller, J. Kim, "New Impossible Differential Attacks on AES", INDOCRYPT 2008, LNCS, 5365, 2008, p. 279-293.

[13] B. Bahrak, M.R. Aref, "Impossible differential attack on seven-round AES-128", IET Inf. Secur., 2 (2), 2008, pp. 28-32.

[14] W. Zhang, W. Wu, D. Feng, "New Results on Impossible Differential Cryptanalysis of Reduced AES", LNCS, 4817, ICISC 2007, 2007, pp.239-250.

[15] K. Aoki, T. Ichikawa, M. Kanda, "Camellia: a 128-bit block cipher suitable for multiple platforms - design and analysis", SAC 2000, LNCS, 2012, 2000, pp. 39-56.

[16] J. Lu, J. Kim, N. Keller, O. Dunkelman, "Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1", CT-RSA 2008, LNCS, 4964, 2008, pp. 370-386.

[17] H. Mala, M. Dakhilalian, M. Shakiba, "Impossible differential cryptanalysis of reducedround Camellia-256", IET Inf. Secur., 5 (3), 2011, pp. 129-134.

[18] J. Lu, Y. Wei, P.A. Fouque, J. Kim, "Cryptanalysis of reduced versions of the Camellia block cipher", IET Inf. Secur., 6 (3), 2012, pp. 228-238.

[19] H. Mala, "Unified Byte Permutations for the Block Cipher 3D", Journal of Computing and Security, 1 (1), 2014, pp. 15-22.