IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 11, 2018, accepted February 18, 2018, date of publication February 22, 2018, date of current version March 13, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808484

Impossible Differential Cryptanalysis of 8-Round Deoxys-BC-256

ZILONG JIANG AND CHENHUI JIN
Information Science and Technology Institute, Zhengzhou 450000, China
Corresponding author: Zilong Jiang (dracipher@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61772547, Grant 61402523, and Grant 61272488.

ABSTRACT Deoxys is a third-round candidate of the CAESAR authenticated encryption competition. In this paper, we present the first cryptanalysis of Deoxys in the single-key model. Specifically, we propose a multiple impossible differentials attack of 8-round Deoxys-BC-256, which can reuse the plaintexts to sieve subkeys, so that the sieving efficiency can be improved. Meanwhile, we improve the process of sieving subkeys and utilize various techniques, including tweak schedule considerations, early abort technique, the new early abort technique, and so on, which help to reduce the complexity. The time, memory, and data complexities are $2^{123.9}$ memory accesses, $2^{99.2}$ bytes, and 2^{117} chosen plaintexts, respectively.

INDEX TERMS Tweakable block cipher, multiple impossible differentials, Deoxys, TWEAKEY, new early abort technique.

I. INTRODUCTION

Tweakable block cipher was proposed in 2002 by Liskov et al. [1]. For the sequence of the 16-byte plaintexts, in contrast to the encrypt process of traditional block cipher $C = E_{key}(P)$, tweakable block cipher provides an additional parameter called "tweak" and the encrypt process can be expressed as: $C = E_{key}(P, Tweak)$. This tweak is usually public but can increase the variability.

To make the design more resistant to related-key attacks, Jean et al. [8] presented the TWEAKEY framework at ASIACRYPT 2014. Based on the TWEAKEY framework, the design of Deoxys-BC follows AES, but both the tweakey schedule and the number of rounds are different. Deoxys-BC was submitted to the CAESAR competition and was selected as one of the third-round candidates in 2016. The designers' analysis in [9] focused on linear and differential cryptanalysis. Then Cid et al. provided the first independent security analysis of Deoxys, and they presented boomerang and rectangle attacks to Deoxys-BC in the related-tweakey model [17], which need both non-zero key difference and non-zero tweak difference.

Since Deoxys is an AES-type cipher with an improved tweakey schedule, the designers claimed that the security bound of Deoxys-BC against most of attacks matches the bounds of AES, and they "*encourage to investigate attack vectors that rely on some additional property of the AES key schedule of Deoxys-BC, for instance* ***impossible differential attacks***" [9]. Table I summaries some attacks of AES-128 and the best-performing attacks are meet-in-the-middle and impossible differential cryptanalysis on 7-round AES-128. We want to find a stronger cryptanalysis in the single-model than these.

TABLE 1. Summary of the attacks on 7-round AES-128 in the single-key model.

Type	Time	Data	Ref.
Collisions	2^{128-e}	2^{32}	[5]
Partial sum	2^{120}	2^{128-e}	[3]
Meet-in-the-middle	2^{123}	2^{80}	[10]
Meet-in-the-middle	2^{99}	2^{97}	[11]
Impossible differential	$2^{117.2}$MA	$2^{112.2}$	[6]
Impossible differential	$2^{110.2}$	$2^{106.2}$	[7]

MA: memory accesses

A. OUR RESULTS

We first provide security analysis of Deoxys in the single-key model. Specifically, we first propose an impossible differential attack on 8-round Deoxys-BC-256, which is one round more than the best result of AES-128. Only 1-byte non-zero tweak difference is needed. In order to ensure that there exits the non-zero tweak difference, we provide the loose constraint that the key size should be less than 240 bits (key size

of Deoxys-BC-256 can be greater than or equal to 128 bits). Utilizing the idea that non-zero tweak difference input will cancel a difference in the attack trail [16], we construct three attack trails and present a multiple impossible differentials attack. Different from the scenario in [4], we improve the process in sieving subkeys. More precisely, we first fix the common subkey bits of the three attack trails, and then sieve other 7-byte subkeys. If all the 7-byte subkeys are wrong, the current common subkey is wrong, so it is unnecessary to be sieved again in another attack trail. Therefore, the sieving efficiency can be improved. Furthermore, we also utilize additional techniques, including tweak schedule considerations, the new early abort [12], early abort [15] etc., which help to reduce the complexity. We obtain the best result of Deoxys-BC-256 in the single-key model so far.

B. ORGANIZATION

The rest of the paper is organized as follows: Section II briefly describes Deoxys-BC and provides the notations used in this paper. Section III proposes 4-round impossible differentials of Deoxys. Section IV presents a multiple impossible differentials attack on 8-round Deoxys-BC-256 and performs a complexity analysis on our attack. Section V concludes this paper.

II. PRELIMINARIES

A. A BRIEF DESCRIPTION OF DEOXYS-BC

Deoxys, an AES-like tweakable block cipher with *SPN* structure, supports the key and the tweak with the sizes of 256 and 384 bits. The numbers of rounds for the two variants are 14 and 16, respectively. The plaintext, the ciphertext and the internal state of Deoxys are treated as a 4×4 matrix over the finite field $GF(2^8)$. Similar to AES, round function of Deoxys applies four operations as follows:

(1) AddRoundTweakey(ART): This operation includes an XOR with the round subtweakeys which are derived from the master tweakey.

(2) SubBytes(SB): This operation applies the AES S-box on each byte of the state.

(3) ShiftRows(SR): This operation is a linear transformation, which rotates the j-th row of the 4×4 matrix to the left by j bytes for $j = 0, 1, 2, 3$.

(4) MixColumns(MC): Another linear transformation is a multiplication by the MDS matrix of AES.

After the last round, an extra AddRoundTweakey ART operation is applied to produce the ciphertext.

Fig. 1 describes the tweakey schedule algorithm of Deoxys-BC-256. Designers denote the concatenation of the key K and the tweak T as KT, i.e. $KT = K||T$. The key size can be greater than or equal to 128 bits. The tweakey state is divided into two 128-bit words, with the first and second words of KT being W_1 and W_2. In order to obtain other round tweakeys, the following tweakey schedule is adopted:

$$TK^1_{i+1} = h(TK^1_i), \quad TK^1_0 = W_1;$$
$$TK^2_{i+1} = h(LFSR_2(TK^2_i)), \quad TK^2_0 = W_2;$$

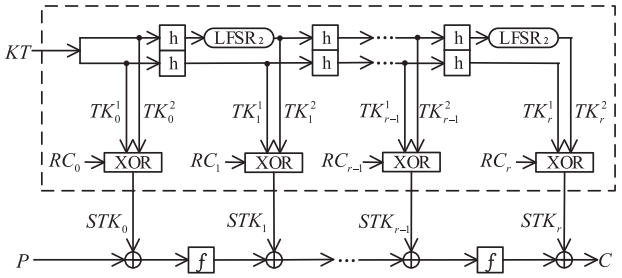

FIGURE 1. The TWEAKEY framework for Deoxys-BC-256.

where $LFSR_2$ is the applications of $LFSR$ on each byte of tweakey words. The byte permutation h is given as follows:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 6 & 11 & 12 & 5 & 10 & 15 & 0 & 9 & 14 & 3 & 4 & 13 & 2 & 7 & 8 \end{pmatrix}$$

The i-th round subtweakey is defined as: $STK_i = TK^1_i \oplus TK^2_i \oplus RC_i$, where RC_i are constants.

For more details, please refer to [9].

B. NOTATIONS

Some notations are given as follows:

P / C / T:	Plaintext/Ciphertext/Tweak
$x^{SB/SR/MC/ART}_{i,[p,\cdots,r]}$:	The intermediate values of the $p^{th}, \cdots, r^{th}$ bytes after the $SB/\ SR/MC/ART$ of Round i
$x^I_{i,[p,\cdots,r]}$:	The input values of the $p^{th}, \cdots, r^{th}$ bytes of Round i
Δx:	The difference of x and x'
$k_{i,[p,\cdots,r]}$:	the values of $p^{th}, \cdots, r^{th}$ subkey bytes of Round i
$x_{i,col(j)}$:	The j-th column of $x_i, j = 0, 1, 2, 3$
$SR[k_{i,col(j)}]$:	The j-th column of k_i through the SR operation, $j = 0, 1, 2, 3$

In this paper, we denote the whitening tweakey as k_0.

III. IMPOSSIBLE DIFFERENTIAL DISTINGUISHERS OF DEOXYS

In this section, we will present three impossible differential distinguishers with same input difference. We utilize the tweak input difference with a single active byte $T_{0,[14]}$ (i.e. $\Delta T_{0,[14]} \neq 0$, and other bytes difference are zero). In order to ensure $\Delta T_{0,[14]} \neq 0$, the key size must be less than 240 bits in this paper. Based on the tweakey schedule, we can get $\Delta T_{3,[1]} \neq 0$, $\Delta T_{4,[6]} \neq 0$ and $\Delta T_{5,[15]} \neq 0$.

We construct three distinguishers with the consideration of non-zero tweak difference. More precisely, given the third round input difference that $\Delta x^I_{3,[0]} \neq 0$ and other bytes are zero, then after 4 rounds we cannot get the output difference that $\Delta x^{SR}_{6,[0,1,3]} \neq 0$ and other bytes are zero. Another two positions of non-zero output difference are $\Delta x^{SR}_{6,[0,2,3]}$ and $\Delta x^{SR}_{6,[1,2,3]}$. Fig. 2 shows one sample of distinguishers.

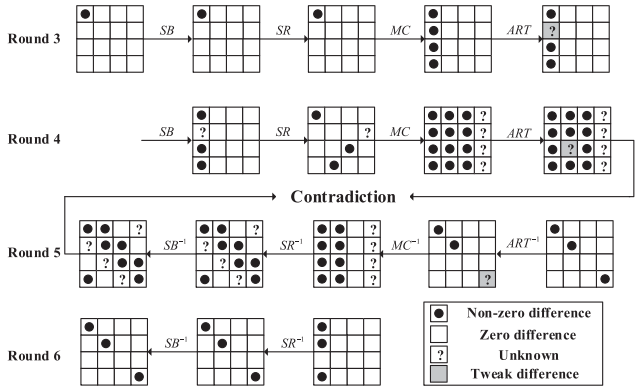

FIGURE 2. One sample of the 4-round impossible differential distinguishers of Deoxys-BC.

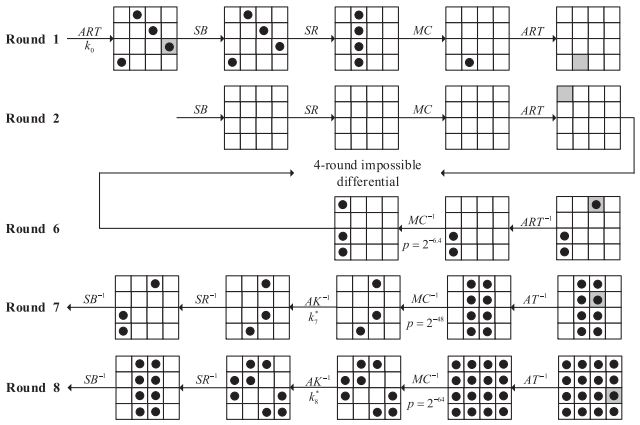

FIGURE 3. One of 8-round impossible differential attack trails.

IV. MULTIPLE IMPOSSIBLE DIFFERENTIAL ATTACK ON 8-ROUND DEOXYS-BC-256

We construct three attack trails and present a 8-round multiple impossible differentials attack. Fig. 3 shows one of 8-round attack trails, and we can see that the non-zero tweak difference $T_{1,[7]}$ will cancel the difference $x^{ART}_{1,[7]}$ in the state at first round.

We denote the key size as $128 + x$ bits ($x \leq 112$), so the tweak size is $128 - x$ bits. For $STK_0 = TK^1_0 \oplus TK^2_0 \oplus RC_0$, we define AT_0 (tweak) equal to the final $\lfloor 128 - x \rfloor$-byte TK^2_0 XOR RC_0, and AK_0 (key) equal to the first $\lceil x \rceil$-byte TK^2_0 XOR TK^1_0, so $STK_0 = AK_0 \oplus AT_0$. Using the tweakey schedule algorithm of Deoxys-BC-256, we can obtain AT_7, AK_7, AT_8 and AK_8. As shown in Fig. 3, we swap the order of AT and MC operations of 7-th and 8-th round. Then we can get the equivalent round subkey $k^*_i = MC^{-1}[k_i]$. Note that AK^{-1}, AT^{-1}, SB^{-1}, SR^{-1} and MC^{-1} are denoted as the reversion of AK, AT, SB, SR and MC, respectively.

A. THE PROCESS OF 8-ROUND ATTACK ON DEOXYS-BC-256

In this section, the process can be divided into three phases:

Precomputation Phase: In order to reduce the complexity, we precompute tables H, Q and Λ_i, $i = 1, 2, 3$.

Table H: Let S denote the 8-bit S-box of Deoxys, Δ_{in} and Δ_{out} denote the input and output difference of S-box. When Δ_{in} and Δ_{out} are non-zero bytes, the equation $S(x) \oplus S(x \oplus \Delta_{in}) = \Delta_{out}$ has one solution on average. We construct a table H, and then store the calculated x in the table H indexed by 2^{16} possible values of $(\Delta_{in}, \Delta_{out})$.

Table Q: We guess all possible values of $T_{0,[14]}$. Based on the tweakey schedule algorithm, we calculate 2^8 values of 4-byte $T_{1,[7]}|T_{6,[8]}|T_{7,[9]}|T_{8,[14]}$, and store them in the table Q indexed by 2^8 possible values of $T_{0,[14]}$.

Table Λ_1: For all possible non-zero values of $\Delta x^{MC}_{6,[2,3]}$, calculate the values of $\Delta x^{SR}_{6,col(0)}$. Choose the values of $\Delta x^{MC}_{6,[2,3]}$ whose corresponding difference $\Delta x^{SR}_{6,col(0)}$ is zero in byte 0 or 1 or 2 (i.e. it satisfies the output difference of three distinguishers). Store the remaining $2^{16} \times 3 \times 2^{-8} \approx 2^{9.6}$ values of $\Delta x^{MC}_{6,[2,3]}$ in the table Λ_1.

Table Λ_2: For all possible non-zero values of $\Delta x^{MC}_{6,[0,2]}$, calculate the values of $\Delta x^{SR}_{6,col(0)}$. Choose the values of $\Delta x^{MC}_{6,[0,2]}$ whose corresponding difference satisfies the output difference of three distinguishers. Store the remaining $2^{9.6}$ values of $\Delta x^{MC}_{6,[0,2]}$ in the table Λ_2.

Table Λ_3: For all possible non-zero values of $\Delta x^{MC}_{6,[1,2]}$, calculate the values of $\Delta x^{SR}_{6,col(0)}$. Choose the values of $\Delta x^{MC}_{6,[1,2]}$ whose corresponding difference satisfies the output difference of three distinguishers. Store the remaining $2^{9.6}$ values of $\Delta x^{MC}_{6,[1,2]}$ in the table Λ_3.

We prepare tables Λ_i, $i = 1, 2, 3$ for extracting the equivalent subkeys of $k^*_{7,[7,10]}$, $k^*_{7,[0,10]}$ and $k^*_{7,[10,13]}$, respectively.

Data Collecting Phase: Considering the non-zero tweak difference $\Delta T_{1,[7]}$ and $\Delta T_{8,[14]}$, which can be obtained from the table Q, the data collecting phase is divided into three steps:

(1) Select 2^{32} plaintexts that take all possible values in the 4 bytes at position [3,4,9,14] and other bytes remain constant. Meanwhile, select 2^8 tweaks that differ only in one tweak byte $T_{0,[14]}$. Then we take this 2^{40} plaintext-tweak inputs (P, T) as a structrue and get the corresponding ciphertexts. For the 2^{40} ciphertexts, calculate $\Delta x^{AK}_{8,col(0,1,2)} = MC^{-1}[C_{\Delta col(0,1,2)}]$. Using the plaintext pair sieve method based on quicksort algorithm [14], select $2^{40+39} \times 2^{-48} = 2^{31}$ pairs of plaintext-tweak-ciphertext $(P, T, C; P', T', C')$ with $\Delta x^{AK}_{8,[0,3,6,7,9,10]} = 0$.

(2) For the 2^{31} remaining ciphertext pairs, considering the corresponding $\Delta T_{8,[14]}$, calculate $\Delta x^{AK}_{8,col(3)} = MC^{-1}[\Delta C_{col(3)} \oplus \Delta T_{8,[14]}]$ and select $2^{31} \times 2^{-16} = 2^{15}$ pairs with $\Delta x^{AK}_{8,[12,13]} = 0$. 2^{15} pairs of $(P, T; P', T')$ and (x^{AK}_8, x'^{AK}_8) can be obtained.

(3) Because the non-zero tweak difference $\Delta T_{1,[7]}$ will cancel the difference in the state at first round, $\Delta x^{SB}_{1,[3,4,9,14]} = SR^{-1}[MC^{-1}[\Delta T_{1,[7]}]]$ can be obtained. For the pair of $(P, T; P', T')$, calculate $\Delta x^{ART}_{0,[3,4,9,14]} = \Delta P[3, 4, 9, 14] \bigoplus \Delta T[3, 4, 9, 14]$. Then $x^{ART}_{0,[3,4,9,14]}$ can be obtained by accessing the table H, and we calculate $k_{0,[3,4,9,14]} = P_{[3,4,9,14]} \bigoplus T_{0,[3,4,9,14]} \bigoplus x^{ART}_{0,[3,4,9,14]}$. Store

2^{15} corresponding pairs of (T, T') and $(x^{AK}_{8,SR[col(1,2)]}, x'^{AK}_{8,SR[col(1,2)]})$ in table Ω_1, which are indexed by 2^{32} subkeys $k_{0,[3,4,9,14]}$. There remains $2^{15} \div 2^{32} = 2^{-17}$ on average for each $k_{0,[3,4,9,14]}$.

We take 2^n structures and store 2^{n+15} pairs of $(T|x^{AK}_{8,SR[col(1,2)]}, T'|x'^{AK}_{8,SR[col(1,2)]})$ in the table Ω_1. There remains 2^{n-17} on average for each $k_{0,[3,4,9,14]}$. For other two attack trails, using the similar Data Collecting steps, we also store corresponding 2^{n+15} pairs of $(T|x^{AK}_{8,SR[col(0,2)]}, T'|x'^{AK}_{8,SR[col(0,2)]})$ and $(T|x^{AK}_{8,SR[col(2,3)]}, T'|x'^{AK}_{8,SR[col(2,3)]})$ in the table Ω_2 and Ω_3, respectively.

Online Attack Phase: The online attack phase can be summarized in the following steps. Utilizing the early abort technique [15], step 1 and step 2 select the pairs which satisfy the expected attack trail of 7-th and 8-th rounds. Using the early abort [6], [15] and new early abort [12], step 3 discards wrong subkeys efficiently. Utilizing other two impossible differential trails, step 4-5 fix the common 8-byte subkey $k_{0,[3,4,9,14]}|k^*_{8,[2,5,8,15]}$ and sieve other 7-byte subkeys $k^*_{7,[0,8,10]}|k^*_{8,[0,7,10,13]}$ and $k^*_{7,[8,10,13]}|k^*_{8,[3,6,9,12]}$, respectively. In step 6, we can get $k_{0,[3,4,9,14]}|k^*_{7,[0,7,8,10,13]}|k^*_8$. Using the tweakey schedule algorithm, first, we sieve the candidate subkeys. Then the remaining candidates can be tested by brute force until the correct key is obtained.

The specific steps of online phase are:

1) For current $k_{0,[3,4,9,14]}$, attack $k^*_{8,SR[col(2)]}$.
Accessing the table Ω_1, obtain 2^{n-17} values of $\Delta x^{SB}_{8,col(2)}$ through the SR^{-1} operation. For each $\Delta x^{SB}_{8,col(2)}$, guess all possible values of $\Delta x^{AK}_{7,[8,10]}$ and obtain 2^{16} values of $\Delta x^{MC}_{7,col(2)}$ through the MC operation. For current $(T_{0,[14]}, T'_{0,[14]})$, the value of $\Delta T^{AT}_{7,[9]}$ can be obtained by accessing the table Q. Based on 2^{16} pairs of $(\Delta x^{AT}_{7,col(2)}, \Delta x^{SB}_{8,col(2)})$, access the table H and calculate $k^*_{8,SR[col(2)]}$ such that $k^*_{8,SR[col(2)]} = x^{AK}_{8,SR[col(2)]} \oplus x^{SR}_{8,SR[col(2)]}$.
Therefore, $2^{n-17+16} = 2^{n-1}$ values of $k^*_{8,SR[col(2)]}$ can be obtained. Then store the corresponding pairs of $(T|x^{AK}_{8,SR[col(1)]}, T'|x'^{AK}_{8,SR[col(1)]})$ and $(x^{AK}_{7,[8,10]}, x'^{AK}_{7,[8,10]})$ in the table R_1 indexed by $k^*_{8,SR[col(2)]}$. There remains $2^{n-1} \div 2^{32} = 2^{n-33}$ on average for each $k^*_{8,SR[col(2)]}$.
2) For current $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$, attack $k^*_{8,SR[col(1)]}$.
Accessing the table R_1, obtain 2^{n-33} values of $\Delta x^{SB}_{8,col(1)}$ through the SR^{-1} operation. For each $\Delta x^{SB}_{8,col(1)}$, guess all possible values of $\Delta x^{AK}_{7,[7]}$ and obtain 2^8 values of $\Delta x^{AT}_{7,col(1)}$ through the MC and AT operations. Based on 2^8 pairs of $(\Delta x^{AT}_{7,col(1)}, \Delta x^{SB}_{8,col(1)})$, access the table H and calculate $k^*_{8,SR[col(1)]}$ such that $k^*_{8,SR[col(1)]} = x^{AK}_{8,SR[col(1)]} \oplus x^{SR}_{8,SR[col(1)]}$.
Therefore, $2^{n-33+8} = 2^{n-25}$ values of $k^*_{8,SR[col(1)]}$ can be obtained. Then store the corresponding pairs of $(T|x^{AK}_{7,[7,8,10]}, T'|x'^{AK}_{7,[7,8,10]})$ in the table R_2 indexed by $k^*_{8,SR[col(1)]}$. There remains $2^{n-25} \div 2^{32} = 2^{n-57}$ on average for each $k^*_{8,SR[col(1)]}$.
3) For current $k_{0,[3,4,9,14]}|k^*_{8,SR[col(1,2)]}$, attack $k^*_{7,[7,8,10]}$.
Accessing the table R_2, obtain 2^{n-57} pairs of (T, T') and values of $\Delta x^{SB}_{7,[2,3,8]}$ through the AK^{-1} and SR^{-1} operations.
For each $\Delta x^{SB}_{7,[8]}$, the pair of $(T_{6,[8]}, T'_{6,[8]})$ can be obtained by accessing the table Q for the corresponding $(T_{0,[14]}, T'_{0,[14]})$. Because the tweak difference $\Delta T_{6,[8]} = \Delta x^{ART}_{6,[8]}$, the pair of $(x^{SR}_{7,[8]}, x'^{SR}_{7,[8]})$ can be obtained by accessing the table H. Then calculate $k^*_{7,[8]}$ such that $k^*_{7,[8]} = x^{SR}_{7,[8]} \oplus x^{AK}_{7,[8]}$.
For each $\Delta x^{SB}_{7,[2,3]}$, access the table Λ_1 and obtain $2^{9.6}$ values of $\Delta x^{ART}_{6,[2,3]}$ through the ART operation. Then access the table H and obtain $2^{9.6}$ values of $x^{SR}_{7,[7,10]}$ through the SR operation. Calculate $k^*_{7,[7,10]}$ such that $k^*_{7,[7,10]} = x^{SR}_{7,[7,10]} \bigoplus x^{AK}_{7,[7,10]}$.
Construct a table M which has 2^{24} addresses indexed by 3-byte equivalent subkey $k^*_{7,[7,8,10]}$. For each address, store 0 or 1 and the initial value is 0. Then we set a counter by variable $Flag$ with initial value 0. For each pair in the table R_2, discard $2^{9.6}$ wrong equivalent subkeys $k^*_{7,[7,8,10]}$. We check if the value at corresponding address of $k^*_{7,[7,8,10]}$ is 0 in the table M. If so, update the location to 1 and increase the value of $Flag$ by 1. If $Flag = 2^{24}$, we judge that current subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(1,2)]}$ is wrong. Then check the next subkey.
If $Flag < 2^{24}$ after checking all the remaining pairs in the table R_2, we search out all equivalent subkeys $k^*_{7,[7,8,10]}$ with $M_1[k^*_{7,[7,8,10]}] = 0$. We conclude that corresponding subkeys $k_{0,[3,4,9,14]}|k^*_{7,[7,8,10]}|k^*_{8,SR[col(1,2)]}$ are candidates and store $k^*_{7,[7,8,10]}|k^*_{8,SR[col(1)]}$ in the table S_1.
After sieving all 7-byte subkeys $k^*_{7,[7,8,10]}|k^*_{8,SR[col(1)]}$, if the table S_1 is empty, we conclude that the current subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$ is wrong and check the next subkey, or store all the candidates in the table S_1 and proceed to the next step.
4) For current $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$, use the second trail to attack $k^*_{7,[0,8,10]}|k^*_{8,SR[col(0)]}$.
(4.1) 2^{n-17} pairs of $(T|x^{AK}_{8,SR[col(0,2)]}, T'|x'^{AK}_{8,SR[col(0,2)]})$ can be obtained by accessing the current $k_{0,[3,4,9,14]}$ in the table Ω_2. Utilizing the step very similar to step 1, 2^{n-1} values of $k^*_{8,SR[col(2)]}$ can be obtained. Then store the corresponding pairs of $(T|x^{AK}_{8,SR[col(0)]}, T'|x'^{AK}_{8,SR[col(0)]})$ and $(x^{AK}_{7,[8,10]}, x'^{AK}_{7,[8,10]})$ in the table R_3 indexed by $k^*_{8,SR[col(2)]}$. There remains 2^{n-33} on average for each $k^*_{8,SR[col(2)]}$.
(4.2) Utilizing the step similar to step 2, 2^{n-25} values of $k^*_{8,SR[col(0)]}$ can be obtained. Then store the corresponding pairs of $(T|x^{AK}_{7,[0,8,10]}, T'|x'^{AK}_{7,[0,8,10]})$ in the table R_4

indexed by $k^*_{8,SR[col(0)]}$. There remains 2^{n-57} on average for each $k^*_{8,SR[col(0)]}$.
(4.3) Utilizing the step similar to step 3, attack 3-byte subkeys $k^*_{7,[0,8,10]}$. Store the candidate subkeys $k^*_{7,[0,8,10]}|k^*_{8,SR[col(0)]}$ in the table S_2. After sieving all 7-byte subkeys $k^*_{7,[0,8,10]}|k^*_{8,SR[col(0)]}$, if the table S_2 is empty, the current subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$ is wrong and check the next subkey, or store all the candidates in the table S_2 and proceed to the next step.

5) For current $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$, use the third trail to attack $k^*_{7,[8,10,13]}|k^*_{8,SR[col(3)]}$.
Use the step similar to step 4. After sieving all 7-byte subkeys $k^*_{7,[8,10,13]}|k^*_{8,SR[col(3)]}$, if the table S_3 is empty, the current subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(3)]}$ is wrong and check the next subkey, or store all the candidates in the table S_3 and proceed to the next step.

6) For current $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$, the values of $k^*_{7,[0,7,8,10,13]}|k^*_{8,SR[col(0,1,3)]}$ can be obtained by accessing tables S_1, S_2 and S_3. The key size is $128 + x$ bits ($x \leq 112$).
(6.1) We discard candidates $k_{0,[3,4,9,14]}|k^*_{7,[0,7,8,10,13]}|k^*_8$ in case of mismatch of 2-byte subkeys $k^*_{7,[8,10]}$. The pass rate is 2^{-32}.
(6.2) Guessing all possible values of the first x-bit TK^2_8, the values of TK^1_8 and TK^2_8 can be obtained. Using the tweakey schedule, discard candidates in case of mismatch of 9-byte subkeys $k_{0,[3,4,9,14]}|k^*_{7,[0,7,8,10,13]}$. The pass rate is 2^{-72}.
(6.3) The remaining candidates can be checked by brute force until the right key is obtained.

B. COMPLEXITY ANALYSIS

The complexity of Precomputation Phase can be neglected compared with other two phases. Data Collecting Phase requires $3 \times 2^n \times 2^{40}\log_2 2^{40} \approx 2^{n+46.9}$ comparisons, and $3 \times 2^{n+15} \times 2 \times (16 + 8) \approx 2^{n+22.2}$ bytes of memory.

The complexities of Online Attack Phase are composed of the following steps:

1) Step 1 requires $2^{32} \times 2^{n-17} \times 2^{16} = 2^{n+31}$ memory access (MA) and $2^{n-1} \times 2 \times (16 + 4 + 2) \approx 2^{n+4.5}$ bytes of memory.
2) Step 2 requires $2^{64} \times 2^{n-33} \times 2^8 = 2^{n+39}$ MA and $2^{n-25} \times 2 \times (16 + 3) \approx 2^{n-19.8}$ bytes of memory.
3) In step 3, a wrong subkey can pass one test with a probability $P_1 = 1 - 2^{-14.4}$, and the probability that 2^{24} wrong subkeys $k^*_{7,[7,8,10]}$ cannot pass the test of $2^{14.4}$ pairs of paintexts is $[1-(1-2^{-14.4})^{2^{14.4}}]^{2^{24}} \approx e^{-2^{24-1.4425}}$. If all the 3-byte subkey $k^*_{7,[7,8,10]}$ cannot pass the test, we conclude that the corresponding subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(1,2)]}$ cannot pass the test. So the probability that a wrong subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(1,2)]}$ can pass the test of $2^{14.4}d$ pairs but cannot pass the test of $2^{14.4}(d+1)$ pairs is $[1-(1-2^{-14.4})^{2^{14.4}(d+1)}]^{2^{24}} - [1-(1-2^{-14.4})^{2^{14.4}d}]^{2^{24}} \approx e^{-2^{24-1.4425(d+1)}} - e^{-2^{24-1.4425d}}$. Thus, the mathematical expectation of d is
$E(d) \approx \sum_{d=1}^{\infty} d(e^{-2^{24-1.4425\times(d+1)}} - e^{-2^{24-1.4425\times d}}) \approx 2^{4.1}$.
So step 3 requires $2^{32} \times 2^{64} \times 2^{14.4+4.1} = 2^{114.5}$ MA.
Similarly, if all 7-byte subkeys $k^*_{7,[7,8,10]}|k^*_{8,SR[col(1)]}$ are wrong, the current common 8-byte subkey $k_{0,[3,4,9,14]}|k^*_{8,SR[col(2)]}$ is wrong, so the probability that commom subkeys can pass the test of step 3 is $P_2 = 1-[1-(1-2^{-14.4})^{2^{n-57}}]^{2^{56}} \approx 1-e^{-2^{56-1.4425\times 2^{n-71.4}}}$, and the number of the remaining common subkeys is $2^{64} \times P_2$.
4) The complexity of step 4.1 is the same as the complexity of step 1. The time complexities of step 4.2 and 4.3 are $2^{n+39} \times P_2$ and $2^{114.5} \times P_2$, respectively.
5) The complexity of step 5.1 is the same as the complexity of step 1. The time complexities of step 5.2 and 5.3 are $2^{n+39} \times (P_2)^2$ and $2^{114.5} \times (P_2)^2$, respectively.
6) Take $n = 77$, and then obtain $P_1 \approx 2^{-67}$ and $P_2 \approx 2^{-14}$. The number of the remaining candidates is $\xi = 2^{64} \times 2^{-14\times 3} = 2^{22}$. The time complexity of step 6.1 is 2^{22} MA and $2^{22} \times 2^{-32} = 2^{-10}$ candidates remain. Because we guess all possible values of the first x-bit TK^2_8 ($x \leqslant 112$), the time complexity of step 6.2 is 2^x MA and $2^x \times 2^{-72} = 2^{x-72}$ candidates remain. So the time complexity of step 6.3 is 2^{x-72} 8-round encryptions.

All in all, our attack needs $2^{n+40} = 2^{117}$ chosen plaintexts. The time complexity is dominated by Data Collecting Phase, so the time complexity of our attack is $2^{n+46.9} = 2^{123.9}$ memory access. The memory complexity is decided by Data Collecting Phase, which requires $2^{n+22.2} = 2^{99.2}$ bytes.

V. CONCLUSION

In this paper, the first impossible differential cryptanalysis of Deoxys-BC-256 is proposed in the single-key model. We choose the input tweak difference with a single active byte $T_{0,[14]}$, thus obtaining a loose constraint (i.e. the key size should be less than 240 bits). Using the idea that tweak input allows to cancel a difference in the trail, we construct three 8-round impossible differential trails which are one more round than the best result of AES-128. Furthermore, combined various additional techniques (e.g. tweak schedule considerations), we improve the procedures of sieving subkeys, so as to improve the sieving efficiency and the complexities can be reduced. To the best of our knowledge, this is the best result of Deoxys-BC-256 in the single-key model so far.

ACKNOWLEDGMENT

The authors thank anonymous reviewers for their valuable suggestions.

REFERENCES

[1] M. Liskov, L. Rivest, and D. Wagner, "Tweakable block ciphers," in *Proc. Annu. Int. Cryptol. Conf.*, 2002, pp. 31–46.

[2] M. Liskov, L. Rivest, and D. Wagner, "Tweakable block ciphers," *J. Cryptol.*, vol. 24, no. 3, pp. 588–613, 2011.

[3] N. Ferguson *et al.*, "Improved cryptanalysis of Rijndael," in *Proc. Int. Workshop Fast Softw. Encryption*, 2000, pp. 213–230.

[4] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Suzaki, and T. Kawabata, "Cryptanalysis of CLEFIA using multiple impossible differentials," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, 2008, pp. 1–6.

[5] H. Gilbert and M. Minier, "A collision attack on 7 rounds of Rijndael," in *Proc. AES Candidate Conf.*, 2000, p. 241.

[6] J. Lu, O. Dunkelman, N. Keller, and J. Kim, "New impossible differential attacks on AES," in *Proc. Indocrypt*, vol. 8. 2008, pp. 279–293.

[7] H. Mala, M. Dakhilalian, V. Rijmen, and M. Modarres-Hashemi, "Improved impossible differential cryptanalysis of 7-round AES-128," in *Proc. Indocrypt*, vol. 10. 2010, pp. 282–291.

[8] J. Jean, I. Nikolić, and T. Peyrin, "Tweaks and keys for block ciphers: The TWEAKEY framework," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, 2014, pp. 274–288.

[9] J. Jean, I. Nikolić, and T. Peyrin. (2014). *Deoxys v1.4, Submission to the CAESAR Competition*. [Online]. Available: http://www1.spms.ntu.edu.sg/~syllab/Deoxys

[10] H. Demirci, I. Taşkin, M. Çoban, and A. Baysal, "Improved meet-in-the-middle attacks on AES," in *Proc. Int. Conf. Cryptol.*, 2009, pp. 144–156.

[11] P. Derbez, P.-A. Fouque, and J. Jean, "Improved key recovery attacks on reduced-round AES in the single-key setting," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Technol.*, 2013, pp. 371–387.

[12] X. Li, F. Fu, and X. Guang, "Multiple impossible differential cryptanalysis on reduced FOX," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E98-A, pp. 906–911, Mar. 2015.

[13] X. Li, C.-H. Jin, and F.-W. Fu, "Improved results of impossible differential cryptanalysis on reduced FOX," *Comput. J.*, vol. 59, no. 4, pp. 541–548, Apr. 2016.

[14] Q. Zhang, "Plaintext pair sieve methods in impossible differential attack," *Comput. Eng.*, vol. 2, no. 2010, p. 046, 2010.

[15] J. Lu, J. Kim, N. Keller, N. Keller, and O. Dunkelman, "Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1," *Topics Cryptol.*, vol. 496, no. 2, pp. 370–386, 2008.

[16] C. Dobraunig and E. List, "Impossible-differential and boomerang cryptanalysis of round-reduced KIASU-BC," in *Proc. Cryptographers Track RSA Conf.*, 2017, pp. 207–222.

[17] C. Cid, T. Huang, T. Peyrin, Y. Sasaki, and L. Song, "A security analysis of Deoxys and its internal tweakable block ciphers," *IACR Trans. Symmetric Cryptol.*, vol. 217, no. 3, pp. 73–107, 2017.

[18] E. Biham, O. Dunkelman, and N. Keller, "Related-key boomerang and rectangle attacks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2005, pp. 507–525.

[19] C. Beierle *et al.*, "The SKINNY family of block ciphers and its low-latency variant MANTIS," in *Proc. Annu. Cryptol. Conf.*, 2016, pp. 1–3.

ZILONG JIANG is currently pursuing the Degree with the Information Science and Technology Institute, Zhengzhou, China. His research interests include the design and analysis of block ciphers.

CHENHUI JIN is currently a Professor with the Information Science and Technology Institute, Zhengzhou, China. His research interests include cryptology and information security.