Improved Related-Tweakey Boomerang Attacks on Deoxys-BC

Yu Sasaki(✉)

NTT Secure Platform Laboratories,
3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585, Japan
sasaki.yu@lab.ntt.co.jp

Abstract. This paper improves previous distinguishers and key recovery attacks against Deoxys-BC that is a core primitive of the authenticated encryption scheme Deoxys, which is one of the remaining candidates in CAESAR. We observe that previous attacks by Cid et al. published from ToSC 2017 have a lot of room to be improved. By carefully optimizing attack procedures, we reduce the complexities of 8- and 9-round related-tweakey boomerang distinguishers against Deoxys-BC-256 to 2^{28} and 2^{98}, respectively, whereas the previous attacks require 2^{74} and 2^{124}, respectively. The distinguishers are then extended to 9-round and 10-round boomerang key-recovery attacks with a complexity 2^{112} and 2^{170}, respectively, while the previous rectangle attacks require 2^{118} and 2^{204}, respectively. The optimization techniques used in this paper are conceptually not new, yet we believe that it is important to know how much the attacks are optimized by considering the details of the design.

Keywords: CAESAR · Cryptanalysis · Deoxys-BC
Boomerang attack

1 Introduction

Authenticated encryption (AE) schemes are symmetric-key cryptographic algorithms that provide both confidentiality and authenticity of data in a single primitive. AE schemes offer several advantages when compared with the use of two separate algorithms. For example, it simplifies security arguments and key management, which avoids the risk of misuse of the schemes by non-experts of cryptography. It also offers better efficiency by sharing a part of the computation for confidentiality and authenticity. In the present time, CAESAR [2] is organized by the international cryptologic research community to identify a portfolio of AE schemes that offer advantages over GCM [15].

Deoxys [11] is one of the CAESAR third-round candidates. Its design is based on the tweakable block cipher Deoxys-BC, which is an AES [19] based tweakable block cipher using the tweakey framework [10]. Tweakable block ciphers (TBC) were first introduced and formalized by Liskov et al. [13], and in addition to the two standard inputs, a plaintext and a key, it takes an additional input called

A. Joux et al. (Eds.): AFRICACRYPT 2018, LNCS 10831, pp. 87–106, 2018.
https://doi.org/10.1007/978-3-319-89339-6_6

a tweak. The Deoxys AE scheme makes use of two versions of the cipher as its internal primitive: Deoxys-BC-256 and Deoxys-BC-384.

The tweakey framework unifies the vision of key and tweak as the *tweakey*. An n-bit block cipher using the framework will take a k-bit key and a t-bit tweak, and a dedicated *tweakey schedule* will use the $(k+t)$-bit tweakey to produce the n-bit *round subtweakeys*. This approach allows designers to claim full security of the tweakable block cipher.

The number of existing public security analysis of the Deoxys-BC is limited. The designers provided a few analyses [11]. As the cipher uses the AES round function, with the only differences to AES being the number of rounds (14 for Deoxys-BC-256 and 16 Deoxys-BC-384) and the tweakey schedule, much of the analysis leverages the existing analysis of the AES.

The work by Cid et al. [7] recently published from ToSC 2017 is the only third-party analysis so far. Cid et al. developed automated differential trail search method using the mixed integer linear programming (MILP) [17] to show that the lower bound of the number of active S-boxes is higher than the original expectation by the designers in the related-tweakey setting. Cid et al. then constructed boomerang attacks [20] by combining two short differential trails discovered by their tool. This leads to 8- and 9-round distinguishers against Deoxys-BC-256 with complexity of 2^{74} and 2^{124}, respectively, and 10- and 11-round distinguishers against Deoxys-BC-384 with complexity of 2^{44} and 2^{122}, respectively. Those are further extended to related-tweakey rectangle attacks for recovering key against 9-round and 10-round Deoxys-BC-256 and 12-round and 13-round Deoxys-BC-384. The summary of the previous attacks are given in Table 1.

We noticed that after the submission of this paper, Mehrdad et al. uploaded their analysis that studies related-tweakey impossible differential attacks against Deoxys-BC-256 [16]. One of their focuses is the related-tweak single-key model that is not covered in this paper but the number of attacked rounds is at most 8. Under the same (related-tweak related-key) model as ours, their attack reaches 9 rounds with complexity $(Time, Data, Memory) = (2^{118}, 2^{118}, 2^{114})$.

Our Contributions. In this paper, we present the best cryptanalysis against Deoxys-BC block cipher in the present time. Our attacks utilize the differential trails for boomerang-like attacks found by Cid et al. [7]. We observe that the automated differential trail search in [7] is very optimized, whereas the utilization of the detected trails in the attack procedure is not well-optimized, thus there is a lot of room to be improved. This is perhaps the main innovation of [7] is the development of new MILP models for automated differential search method. Yet we think that optimizing the attack complexity is important especially considering that Deoxys is one of the remaining candidates in CAESAR. For example, to compare security margin of several designs, known cryptanalytic results should be optimized as much as possible for all of the designs.

The optimization techniques used in our paper are conceptually not entirely new, e.g. changing differential trail to truncated differential trail in one of two pairs in the boomerang quartet, reducing the data complexity by using structure,

Table 1. Comparison of the Attacks against Deoxys-BC. SK, RK, KR, and dist stand for single-key, related-key, key-recovery and distinguisher, respectively.

Deoxys-BC-256

Rounds	Model	Approach	Goal	Time	Data	Mem.	Size set up	Ref.
8/14	SK	MitM	KR	$< 2^{128}$	–	–	$t = 128, k = 128$	[11]
8/14	SK	differential	KR	$< 2^{128}$	–	–	$t = 128, k = 128$	[11]
8/14	RK	boomerang	dist	2^{74}	2^{74}	negl.	$t = 128, k = 128$	[7]
				2^{28}	2^{28}	2^{27}	$t = 128, k = 128$	Ours
9/14	RK	boomerang	dist	2^{124}	2^{124}	negl.	$t = 128, k = 128$	[7]
				2^{98}	2^{98}	2^{17}	$t = 128, k = 128$	Ours
9/14	RK	boomerang	KR	2^{118}	2^{117}	2^{117}	$t = 128, k = 128$	[7]
				2^{112}	2^{98}	2^{17}	$t = 128, k = 128$	Ours
10/14	RK	rectangle	KR	2^{204}	$2^{127.58}$	$2^{127.58}$	$t < 52, k > 204$	[7]
		boomerang		2^{170}	2^{170}	2^{17}	$t < 86, k > 170$	Ours
		boomerang		2^{170}	2^{98}	2^{98}	$t < 86, k > 170$	Ours

Deoxys-BC-384

Rounds	Model	Approach	Goal	Time	Data	Mem.	Size set up	Ref.
8/16	SK	MitM	KR	$< 2^{256}$	–	–	$t = 128, k = 256$	[11]
10/16	RK	boomerang	dist	2^{44}	2^{44}	negl.	$t = 128, k = 256$	[7]
				2^{22}	2^{22}	2^{17}	$t = 128, k = 256$	Ours
11/16	RK	boomerang	dist	2^{122}	2^{122}	negl.	$t = 128, k = 256$	[7]
				2^{100}	2^{100}	2^{17}	$t = 128, k = 256$	Ours
12/16	RK	rectangle	KR	2^{127}	2^{127}	2^{125}	$t = 128, k = 256$	[7]
		boomerang		2^{148}	2^{148}	2^{17}	$t = 128, k = 256$	Ours
		boomerang		2^{148}	2^{100}	2^{100}	$t = 128, k = 256$	Ours
13/16	RK	rectangle	KR	2^{270}	2^{127}	2^{144}	$t < 114, k > 270$	[7]

and choosing the best way to append the key-recovery round. However applying a lot of optimization attempts including failure attempts that cannot be included in the paper and considering many details of the computation structure require hard work and significant amount of time. After the careful analysis, the attack complexity against Deoxys-BC-256 with respect to $\min(Time, Data, Memory)$ is improved by a factor of 2^{26} and 2^{34} for the longest distinguishing and key-recovery attacks, respectively. The improved complexities are listed in Table 1.

Finally, we provide a discussion toward developing an automated differential search tool such that the impact of the key recovery attack is taken into account.

Paper Outline. The remaining of this paper is organized as follows. Section 2 describes the specification of Deoxys-BC. Section 3 recalls the previous attacks by Cid et al. Sects. 4 and 5 discuss the improved attacks on Deoxys-BC-256 and Deoxys-BC-384, respectively. We give a discussion toward an improved differential search tool and conclude this paper in Sect. 6.

2 Specification of Deoxys-BC

Deoxys-BC-256 and Deoxys-BC-384 are AES-based tweakable block ciphers [11]. Both versions adopt 128-bit block sizes which besides a plaintext P (or a ciphertext C) and a key K, also take a *tweak* T. The concatenation of the key and tweak states is called the *tweakey* state. For Deoxys-BC-256 the tweakey size is 256 bits, while for Deoxys-BC-384 it is 384 bits. The breakdown of the key and tweak sizes in the tweakey can be chosen by the user, as long as the key size is greater or equal to the block size, i.e. 128 bits.

Deoxys-BC is an AES-like design. It transforms the initial plaintext (viewed as a 4×4 two-dimension array of bytes) using the AES round function. Deoxys-BC-256 and Deoxys-BC-384 have 14 and 16 rounds, respectively.

Deoxys-BC Round Function. Similarly to the AES, one round of Deoxys-BC has the following four transformations applied in the order specified below:

- AddRoundTweakey – XOR the 128-bit round subtweakey to the state.
- SubBytes – Apply the 8-bit AES S-box $\mathcal{S}$ to each byte in parallel.
- ShiftRows – Rotate the 4-byte i-th row left by i positions.
- MixColumns – Multiply the state by the 4×4 constant MDS matrix of AES.

MixColumns is not omitted in the last round. After the last round, a final AddRoundTweakey operation is performed to produce the ciphertext.

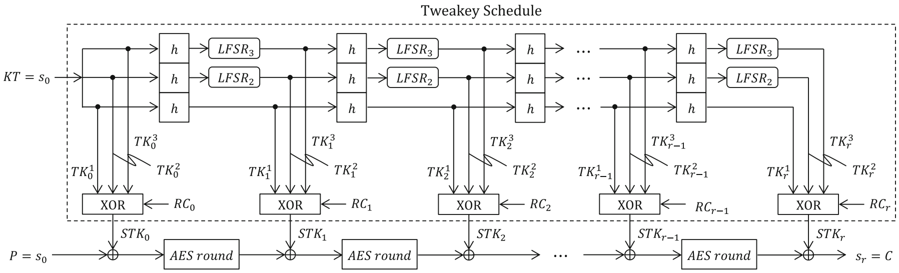

Fig. 1. Encryption of Deoxys-BC-384.

Definition of Subtweakeys. The *tweakey* state is composed of the key K and the tweak T. The tweakey state is divided into 128-bit words denoted by $TK_1, TK_2, \cdots$. More precisely, the tweakey state in Deoxys-BC-256 is composed of TK_1 and TK_2, and the tweakey state in Deoxys-BC-384 is composed of TK_1, TK_2, and TK_3. Finally, we denote by STK_i the 128-bit *subtweakey* that is added to the state at round i during the AddRoundTweakey operation. For Deoxys-BC-256, a subtweakey is defined as $STK_i = TK_i^1 \oplus TK_i^2 \oplus RC_i$, whereas for Deoxys-BC-384 it is defined as: $STK_i = TK_i^1 \oplus TK_i^2 \oplus TK_i^3 \oplus RC_i$.

The 128-bit words TK_i^1, TK_i^2, TK_i^3 are outputs produced by a special *tweakey schedule* algorithm, initialised with $TK_0^1 = W_1$ and $TK_0^2 = W_2$ for Deoxys-BC-256 and with $TK_0^1 = W_1$, $TK_0^2 = W_2$ and $TK_0^3 = W_3$ for Deoxys-BC-384. The tweakey schedule algorithm is defined as

$$\begin{aligned} TK_{i+1}^1 &= h(TK_i^1), \\ TK_{i+1}^2 &= h(LFSR_2(TK_i^2)), \\ TK_{i+1}^3 &= h(LFSR_3(TK_i^3)), \end{aligned}$$

where the byte permutation h is defined as

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

numbered by the usual AES byte ordering.

The $LFSR_2$ and $LFSR_3$ functions are simply the application of an LFSR to each on the 16 bytes of a 128-bit tweakey word. The two LFSRs used are given in Table 2 (x_0 stands for the LSB of the cell).

Table 2. $LFSR_2$ and $LFSR_3$.

$LFSR_2$	$(x_7\|\|x_6\|\|x_5\|\|x_4\|\|x_3\|\|x_2\|\|x_1\|\|x_0) \rightarrow (x_6\|\|x_5\|\|x_4\|\|x_3\|\|x_2\|\|x_1\|\|x_0\|\|x_7 \oplus x_5)$
$LFSR_3$	$(x_7\|\|x_6\|\|x_5\|\|x_4\|\|x_3\|\|x_2\|\|x_1\|\|x_0) \rightarrow (x_0 \oplus x_6\|\|x_7\|\|x_6\|\|x_5\|\|x_4\|\|x_3\|\|x_2\|\|x_1)$

Finally, RC_i denotes the key schedule round constants. We omit the details of constant because it does not impact to the attacks. Encryption of Deoxys-BC-384 is illustrated in Fig. 1.

3 Previous Attacks on Deoxys-BC

Our attacks are based on the related-tweakey boomerang distinguishers and related-tweakey rectangle key-recovery attacks by Cid et al. [7]. In this section, we first briefly recall the framework of the related-tweakey boomerang-like attacks in Sect. 3.1. Then, in Sect. 3.2, we introduce the attacks by Cid et al. that are the main target of this paper.

3.1 Brief Introduction of Boomerang Attacks

Boomerang attacks and variants combine short differential trails with high probability. Here we briefly introduce the framework of the boomerang attack.

Boomerang and Rectangle Attacks. Boomerang attack [20] regards the target cipher as a composition of two sub-ciphers E_0 and E_1. The first sub-cipher is supposed to have a differential $\alpha \rightarrow \beta$, and the second one to have a differential $\gamma \rightarrow \delta$, with probabilities p and q, respectively. The basic boomerang attack requires an adaptive chosen plaintext/ciphertext scenario, and plaintext pairs result in a right quartet with probability p^2q^2. The amplified boomerang attack (also called the rectangle attack) works in a chosen-plaintext scenario and a right quartet is obtained with probability $p^2q^22^{-n}$ [12]. Further, it was pointed out in [3,4] that any value of β and γ is allowed as long as $\beta \neq \gamma$. As a result, the probability of the right quartet is increased to $2^{-n}\hat{p}^2\hat{q}^2$, where

$$\hat{p} = \sqrt{\sum_i \Pr^2(\alpha \rightarrow \beta_i)} \text{ and } \hat{q} = \sqrt{\sum_j \Pr^2(\gamma_j \rightarrow \delta)}.$$

Boomerang and rectangle attacks under related-key setting were formulated in [5]. Let ΔK and ∇K be the key differences for the first and second sub-ciphers, respectively. The attack needs access to four related-key oracles with $K_1 \in \mathbb{K}$, where $\mathbb{K}$ is the key space, $K_2 = K_1 \oplus \Delta K$, $K_3 = K_1 \oplus \nabla K$ and $K_4 = K_1 \oplus \Delta K \oplus \nabla K$. In the related-key boomerang attack, paired plaintexts P_1, P_2 such that $P_1 \oplus P_2 = \alpha$ are queried to K_1 encryption oracle and K_2 encryption oracle, and the attacker receives ciphertexts C_1 and C_2. Then C_3 and C_4 are calculated by $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$, and then queried to K_3 decryption oracle and K_4 decryption oracle. The resulting plaintext difference $P_3 \oplus P_4$ equals to α with probability $\hat{p}^2\hat{q}^2$. The distinguishing game can be described more formally in an algorithmic form as Algorithm 1. The game returns a distinguishing bit $b \in \{0,1\}$ that is set to 1 if the oracle is a target algorithm and 0 if the oracle is an ideal permutation. Algorithm 1 is $(\hat{p}\hat{q})^{-2}$ many iterations of 2 chosen-plaintext and 2 adaptively chosen-ciphertext queries. Hence the attack complexity is $(\mathit{time}, \mathit{data}, \mathit{memory}) = (4 \cdot (\hat{p}\hat{q})^{-2}, 4 \cdot (\hat{p}\hat{q})^{-2}, \mathit{negligible})$.

Algorithm 1. Basic Procedure of Related-Key Boomerang Distinguishers

```
Input: $\alpha, \delta, K_1, K_2, K_3, K_4, \hat{p}\hat{q}$
Output: $b \in \{0,1\}$
for $i \leftarrow 1, 2, \ldots, (\hat{p}\hat{q})^{-2}$ do
    Choose distinct input $P_1$. Set $P_2 \leftarrow P_1 \oplus \alpha$.
    Obtain $C_1 = E_{K_1}(P_1)$ and $C_2 = E_{K_2}(P_2)$ by making encryption queries.
    Set $C_3 \leftarrow C_1 \oplus \delta$ and $C_4 \leftarrow C_2 \oplus \delta$.
    Obtain $P_3 = D_{K_3}(C_3)$ and $P_4 = D_{K_4}(C_4)$ by making decryption queries.
    if $P_3 \oplus P_4 = \alpha$ then
        return 1
    end if
end for
return 0
```

In the attacks against full AES-192 and AES-256 [6], Biryukov and Khovratovich introduced *the boomerang switch* in order to gain free rounds at the boundary of two trails. The idea was to optimize the transition between the sub-trails of E_0 and E_1 in order to minimize the overall complexity of the distinguisher. In the previous boomerang and rectangle attacks against Deoxys-BC by Cid et al. [7], the following two types of switch techniques are exploited.

Ladder switch. A cipher is decomposed into rounds by default. However, decomposition regarding smaller operations, like columns and bytes, may lead to better distinguishers.

S-box switch. Suppose E_0 ends with an S-box and the output difference of this S-box is Δ. If the same difference Δ comes from the path of E_1, then the propagation through this S-box is for free in one of the directions.

The theoretical explanation behind those techniques were later formalized as the sandwich framework [8,9].

3.2 Previous Boomerang and Rectangle Attacks on Deoxys-BC

Cid et al. [7] presented related-tweakey boomerang distinguishers and related-tweakey rectangle key-recovery attacks against reduced rounds of Deoxys-BC. In short, Cid et al. [7] developed a new MILP-based differential search method for related-tweakey boomerang or rectangle attacks. Their tool has the following two advantages; (1) it takes into account incompatibility of linear relations between independently chosen subtweakey differences and (2) it optimizes the active-byte positions by taking into account the gain from the ladder switch technique.

Consequently, Cid et al. found 8- and 9-round related-tweakey boomerang trails against Deoxys-BC-256 with probability $\hat{p}\hat{q} = 2^{-36}$ and 2^{-61}, respectively and 10- and 11-round trails against Deoxys-BC-384 with probability $\hat{p}\hat{q} = 2^{-21}$ and 2^{-60}, respectively. The 8-round trail for Deoxys-BC-256 is presented in Table 3. In Table 3, the first column show the round number, the second, third, fourth, and fifth columns are the difference before `AddRoundTweakey`, the subtweakey difference, the difference before `SubBytes` and the difference before `MixColumns`, respectively. The sixth column is the probability of the total differential propagation during `SubBytes` in each round. The middle two rounds (rounds 4 and 5) are included in both of E_0 and E_1 due to the ladder switch technique. Namely, the boundary of E_0 and E_1 is defined byte-wise or column-wise instead of state-wise, thus some part of the state belongs to E_0 and the other part belongs to E_1. In round 1, the plaintext should have differences in 5 bytes, and one of them is canceled by the subtweakey difference. Then, each non-zero difference is converted to the specific output difference with probability 2^{-6}, thus the total probability in round 1 is $2^{-6\times 4} = 2^{-24}$. The other rounds can be explained similarly.

The other related-tweakey boomerang trails can be found in Tables 4, 5 and 6 in Appendix A. Note that Cid et al. also showed similar trails for more rounds

Table 3. 8-round distinguisher of Deoxys-BC-256

ΔTK_0^1 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46
ΔTK_0^2 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 d1
∇TK_0^1 : 00 00 02 00 00 00 00 b3 00 00 00 00 00 00 00 00
∇TK_0^2 : 00 00 a8 00 00 00 00 96 00 00 00 00 00 00 00 00

R	before ATK	ΔSTK	before SB	before MC	p_r
1	00 b9 00 00 00 00 d1 00 00 00 00 ab 61 00 00 97	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 97	00 b9 00 00 00 00 d1 00 00 00 00 ab 61 00 00 00	00 35 00 00 00 5d 00 00 00 01 00 00 00 8c 00 00	2^{-24}
2	00 00 00 00 00 00 00 00 00 e5 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 e5 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
3	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	ca 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00	1
5	00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00	1
4			00 00 00 00	00 00 00 00	1
5	00 00 00 00	00 00 00 00 00	00 00 00 00 00	00 00 00 00 00	1
6	00 00 00 00 00 03 00 00 00 6a 00 00 00 00 00 00	00 00 00 00 00 03 00 00 00 6a 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
7	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	d5 00 00 00 00 00 00 00 00 00 00 00 00 00 06 00	d5 00 00 00 00 00 00 00 00 00 00 00 00 00 06 00	60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c	2^{-12}

but with much lower probability. Because how to utilize them in the attack is unclear and we do not use them in this paper, we omit those trails.

As the last remarks, in the boomerang-type attacks, the attackers need to be very careful about the compatibility of two independently chosen short trails because such two independent trails may not be connected [18]. Cid et al. did the experimental verification (sometimes by reducing the number of rounds for each trail) to show that the two trails can be connected with the probability evaluated by them.

4 New Attacks on Deoxys-BC-256

In this section, we improve the distinguisher and key recovery attacks by Cid et al. [7] against Deoxys-BC-256. Considering that the differential search method in [7] is fairly optimized, we focus our attention on how to utilize the discovered boomerang trails rather than finding new trails.

4.1 Improved Boomerang Distinguishers

Truncating the Differential Trail. The boomerang distinguishers in [7] are the straightforward applications of Algorithm 1 to their boomerang trails. As shown in Tables 3 and 4, the trails have $(\hat{p}\hat{q}) = 2^{-36}$ and 2^{-61} for 8 rounds and 9 rounds, respectively, thus the data complexities are 2^{74} and 2^{124}, respectively. Hereafter, we first discuss the improvement for the 8-round distinguisher, and will later apply the improvement to 9 rounds.

The first improving point is Step 6 in Algorithm 1 that checks the match between the n-bit computed difference and α. It may be sufficient to match a part of differences as long as the number of matched bits is sufficient to discard all the wrong quartets. Recall the first half of the differential trail in the 8-round distinguisher in Table 3. In the previous attack, the attacker matches 128-bits of differences of the form

```
00 00 00 01  b9 00 00 00  00 d1 00 00  00 00 ab 97
```

at Step 6 in Algorithm 1. We now ignore the match of differences for 4 active bytes before the active S-boxes in the first round, thus only check whether or not 12 bytes have the following difference.

$$00\ 00\ 00\ *\quad *\ 00\ 00\ 00\quad 00\ *\ 00\ 00\quad 00\ 00\ *\ 97$$

Here, 97 in the last byte of the plaintext difference comes from AddRoundTweakey in the first round, which XORs subtweakey difference 97 to the zero-difference byte. Hence, we basically check whether S-boxes in 12 bytes are active or inactive. This saves us the probability of satisfying the differential transition through S-boxes in the first round, which increases the probability of the distinguisher from 2^{-72} to 2^{-48}. Hence, with this effort, the complexity of the distinguisher becomes $(time, data, memory) = (2^{50}, 2^{50}, negligible)$.

It should be noted that the probability that an ideal permutation satisfies the property also increases: $2^{-8\times 12} = 2^{-96}$. Since this is smaller than 2^{-48}, the 8-round distinguisher can work.

Structure Technique. The data complexity of the above distinguisher can be further reduced by constructing the plaintext structure. This requires the use of additional memory, but the total memory amount is still practical.

In the differential trail in Table 3, we need the 4-byte difference of the following form before the MixColumns in the first round:

$$\mathtt{00\ 00\ 00\ 00\ \ 35\ 5d\ 01\ 8c\ \ 00\ 00\ 00\ 00\ \ 00\ 00\ 00\ 00}$$

The corresponding plaintext difference does not have to be the one specified in Table 3 but can be any difference as long as the desired 4-byte difference can be generated with non-zero probability. Let $\mathcal{I}_{\mathtt{35}}$ be a set of differences defined as follows.

$$\mathcal{I}_{\mathtt{35}} \triangleq \{\varDelta \in \{0,1\}^8 \mid \exists x \in \{0,1\}^8 \ s.t. \ \mathcal{S}(x) \oplus \mathcal{S}(x \oplus \varDelta) = \mathtt{35}\}.$$

The size of $\mathcal{I}_{\mathtt{35}}$ is 126. Similarly, $\mathcal{I}_{\mathtt{5d}}$, $\mathcal{I}_{\mathtt{01}}$ and $\mathcal{I}_{\mathtt{8c}}$ can be defined and the size of each set is 126.

To generate the structure, we generate two sets of 2^{32} plaintexts $\mathcal{P}_1$ and $\mathcal{P}_2$, where $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) contains plaintexts to be queried to the oracle with K_1 (resp. K_2). In each set, all possible values are contained for four active bytes in the first round and the other 12 bytes are fixed to some specified value. To be more precise, 12 byte-values $c_i \in \{0,1\}^8, 0 \le i \le 15, i \ne 3,4,9,14$ are fixed, and all the 2^{32} values are collected in the other 4 bytes. $\mathcal{P}_1$ and $\mathcal{P}_2$ are defined as follows. Note that we need to make difference $\mathtt{97}$ in the last byte to cancel the first subtweakey difference.

$$\mathcal{P}_1 \triangleq \{c_0, c_1, c_2, * \quad *, c_5, c_6, c_7 \quad c_8, *, c_{10}, c_{11}, \quad c_{12}, c_{13}, *, c_{15}\},$$
$$\mathcal{P}_2 \triangleq \{c_0, c_1, c_2, * \quad *, c_5, c_6, c_7 \quad c_8, *, c_{10}, c_{11}, \quad c_{12}, c_{13}, *, c_{15} \oplus \mathtt{97}\}.$$

To utilize $\mathcal{P}_1$ and $\mathcal{P}_2$ in the boomerang attack framework, each of the 2^{32} plaintexts in $\mathcal{P}_1$ is queried to the encryption oracle with K_1, then δ is XORed to the resulting ciphertext, and finally the generated ciphertext is queried to the decryption oracle with K_3. Namely the attacker performs the following process and stores the results in $\mathcal{P}_3$:

$$P_3^i \leftarrow D_{K_3}\big(E_{K_1}(P_1^i) \oplus \delta\big),$$

where P_1^i for $i = 0, 1, \cdots, 2^{32} - 1$ is each plaintext in $\mathcal{P}_1$.

The attacker then computes $P_4^i \leftarrow D_{K_4}\big(E_{K_2}(P_2^i) \oplus \delta\big)$ for each plaintext in $\mathcal{P}_2$, and checks the match of 12-byte difference between P_4^i and $\mathcal{P}_3$ (for Step 6 in Algorithm 1) along with another check if 4-byte difference between P_1^i and P_2^i are included in $\mathcal{I}_{\mathtt{35}}$, $\mathcal{I}_{\mathtt{5d}}$, $\mathcal{I}_{\mathtt{01}}$ and $\mathcal{I}_{\mathtt{8c}}$.

Time Complexity and Optimization. In the single structure, $4 \cdot 2^{32}$ queries are made, thus the data complexity is 2^{34}. The number of pairs that can be made from $\mathcal{P}_1$ and $\mathcal{P}_2$ is $2^{32*2} = 2^{64}$, of which $2^{64} \cdot \frac{126}{255}^4 \approx 2^{60}$ pairs satisfy the constraint for P_1 and P_2 (included in $\mathcal{I}_{35}$, $\mathcal{I}_{\mathtt{5d}}$, $\mathcal{I}_{01}$ and $\mathcal{I}_{\mathtt{8c}}$) and there should be $2^{60-48} = 2^{12}$ pairs satisfying the boomerang trail with probability 2^{-48}. This is sufficient to distinguish 8-round Deoxys-BC-256 from an ideal permutation.

One may note that generating 2^{12} quartets is too much for the attack. Indeed, the attack works by setting the size of $\mathcal{P}_1$ and $\mathcal{P}_2$ to 2^{26}. Then among 2^{52} possible pairs, about $2^{52-4-48} = 1$ pair will satisfy the boomerang trail. Overall, the data complexity is $4 \cdot 2^{26} = 2^{28}$ queries and the time complexity is 2^{28} memory access. The attacker needs to store 2^{26} plaintexts in $\mathcal{P}_1$ and in $\mathcal{P}_3$, thus the memory complexity is 2^{27} plaintexts.

Application to 9-Round Distinguisher. Recall the differential trail for the 9-round distinguisher in Table 4. Differently from the 8-round distinguisher, there is no active S-box in the first round. Hence, to apply the similar improvement, we need to change the attack model to the opposite direction i.e. chosen-ciphertext and adaptively chosen-plaintext attack. Note that Deoxys does not omit MixColumns in the last round but it is well-known that having such linear operation in the last round does not change the impact of the differential cryptanalysis. Indeed, the attacker can do analysis by considering $\mathtt{MixColumns}^{-1}(\Delta C)$ instead of ΔC in general.

The rest is similar as the 8-round distinguisher, thus we only give a summary.

- By using the truncation of the differential trail, we can avoid the probability loss of 2^{-12}. Hence, the probability of the boomerang trail becomes $2^{-122+12} = 2^{-110}$. Note that the probability that an ideal permutation provides a pair with zero difference in 14 bytes is 2^{-112}, which is smaller than 2^{-110} in the boomerang trail.
- By using the structure, the attacker can define two sets of 2^{16} ciphertexts $\mathcal{C}_1$ and $\mathcal{C}_3$. 2^{32} pairs can be generated and $2^{32} \cdot \frac{126}{255} \approx 2^{30}$ pairs satisfy the ciphertext difference in the last round.
- After generating 2^{80} structures by changing the constant in 14 bytes of the ciphertext, the number of pairs reaches 2^{110}, thus the attacker can expect to find 1 quartet satisfying the boomerang trail. The data complexity is $4 * 2^{16} = 2^{18}$ per structure, thus 2^{98} in total. The time complexity is 2^{98} memory access. The attacker needs to store 2^{16} ciphertexts in $\mathcal{C}_1$ and $\mathcal{C}_2$, thus the required memory amount is 2^{17}.

Remarks to Increase Distinguishing Advantage. One may think that the gap of the probabilities between the actual cipher and an ideal permutation (2^{-110} vs 2^{-112}) is too small. We argue that the distinguishing advantage can be significantly increased with negligible cost.

The idea is after Step 6 of Algorithm 1 is passed, we inject another check whether the generated pair actually follows the boomerang trail. If so, the distinguishing game returns $b = 1$, otherwise it continues the algorithm. Suppose that a boomerang quartet C_1, C_2, C_3, C_4 is generated. The attacker then modifies C_1 and C_3 to C_1' and C_3' so that the two active S-boxes in the last round will not be modified and obtains the corresponding C_2' and C_4'. If C_1, C_2, C_3, C_4 follows the trail, the differential transitions through the two S-boxes in the last round are already satisfied, hence C_1', C_2', C_3', C_4' should form another boomerang quartet with probability $2^{-110+12} = 2^{-98}$. Thus after generating 2^{98} pairs of C_1' and C_3', the false positive can be eliminated. Note that 2^{98} pairs of C_1' and C_3' can be generated by using the structure technique, thus the complexity of this additional check is 2^{82} with the memory of size 2^{17}.

4.2 Improved Key Recovery Attacks

Cid et al. [7] discussed related-tweakey rectangle attacks. In [7], they directly applied the general complexity analysis in [14] for related-tweakey rectangle attack against SKINNY [1]. Hence, it is of interest to optimize the attack by taking into account the structure of Deoxys-BC. Indeed, we show that the attack complexity can be improved by using boomerang attacks instead of rectangle attacks.

Note that in the 9-round attack, we assume the default setting by the designers in which the 256-bit tweakey consists of 128-bit tweak and 128-bit key.

9-Round Attacks Using 9-Round Distinguisher. Our 9-round attack is very simple but effective. We do not append extra key-recovery round to the distinguisher, but recovers the key inside the distinguisher. We first run the 9-round distinguisher explained in Sect. 4.1 with complexity $(Time, Data, Memory) = (2^{98}, 2^{98}, 2^{17})$ to find a quartet of texts satisfying the boomerang trail. Due to the truncation of the differential trail, we did not pay attention to the output difference of the active S-boxes in the last round to identify the quartet. We use those information for key recovery.

As shown in Table 4, the input differences to the two active S-boxes in the last round are `e3` and `0c`. The output differences can be computed by $\texttt{MixColumns}^{-1}(\Delta C)$. From the property of the S-box, the actual values in those S-boxes can be reduced to about 2 choices per byte. Those immediately reveal 2-byte information of the last subkey STK_9 converted by inverse `MixColumns` and `ShiftRows`, namely $\texttt{ShiftRows}^{-1} \circ \texttt{MixColumns}^{-1}(STK_9)$. Because we have 2 pairs in a quartet, the 2-byte information of STK_9 is uniquely identified. Because STK_9 is an XOR of tweak and key and tweak is known to he attacker, 2-byte information of the master key can be obtained.

The remaining is a simple exhaustive search on the other 14 bytes, which requires 2^{112} computations. Thus the total complexity of the 9-round key recovery is $(Time, Data, Memory) = (2^{112}, 2^{98}, 2^{17})$.

10-Round Attacks Using 9-Round Distinguisher. In this attack, we append 1 round to the end of the 9-round distinguisher in Table 4, which is illustrated in Fig. 2. Note that appending 1-round before the plaintext is hard because of too many active bytes in the initial state of the 8-round distinguisher.

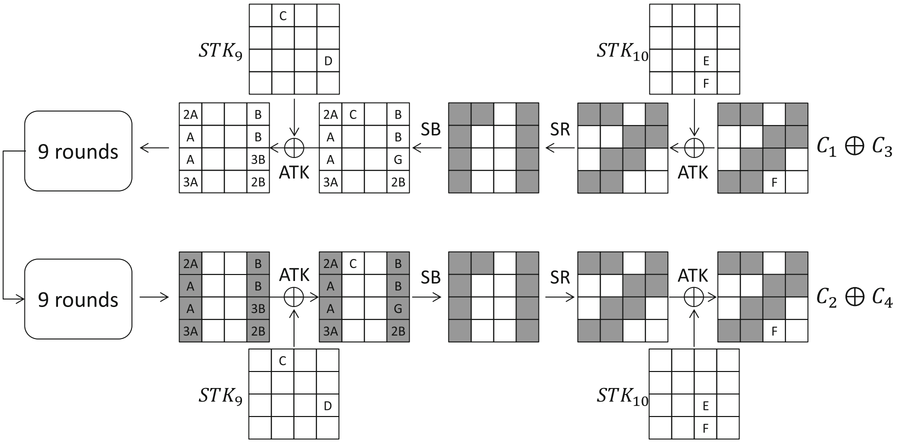

Fig. 2. Key recovery for 10-round Deoxys-BC-256. 'A', 'B', 'C', 'D', 'E' and 'F' are uniquely fixed difference by the boomerang trail and 'G' is $3B \oplus D$.

As discussed before, `MixColumns` in the last round does not have any impact to the attack. To simplify the discussion, we describe the attack by omitting the `MixColumns` in the last round.

In Fig. 2, the upper part is the computations for the first pair (C_1 and C_3). We need to ensure the exact difference of the distinguisher's input (in inverse direction). After P_1 and P_3 are obtained, P_2 and P_4 are generated and encryption queries are made to obtain C_2 and C_4, which is illustrated in the bottom half. Because the output of the distinguisher is truncated, we only know that 2 bytes are active (and then expanded to 2 columns after `MixColumns` in round 9).

As clearly illustrated in Fig. 2, the partial computation from the ciphertext to 9 active bytes through $\texttt{AddRoundTweakey}^{-1}, \texttt{ShiftRows}^{-1}, \texttt{SubBytes}^{-1}$ involves 9 bytes of STK_{10}. By guessing those 9 bytes, we can compute the values of active bytes at the end of distinguisher from the ciphertext, which is enough to construct the structure for the remaining 9 rounds. Besides, for each obtained C_2 and C_4, we can compute back to the difference of the distinguisher's output.

In summary, by exhaustively guessing 9 bytes of STK_{10}, we can apply the 9-round distinguisher, which returns a valid boomerang quartet only when the guess of STK_{10} is correct. The attack is 2^{72} repetitions of the 9-round distinguisher that requires 2^{98} data and time complexities with 2^{17} memory, thus $(Time, Data, Memory) = (2^{170}, 2^{170}, 2^{17})$.

Note that during 2^{72} iterations of the 9-round distinguisher, 2^{98} queried data can be reused. (For different guess, only the order to make pairs changes.) Thus by storing 2^{98} queried data in the memory, the attack complexity can be $(Time, Data, Memory) = (2^{98}, 2^{170}, 2^{98})$.

In both cases, the attack is faster than the exhaustive search only if the key size in 256-bit tweakey is bigger than 170 bits. Given that the previous attack

[7] only works when the key size is bigger than 204 bits, our attack not only improves the complexity but also extends the attacked parameters.

5 New Attacks on Deoxys-BC-384

10-Round Distinguisher. Recall the previous 10-round distinguisher that uses the trail in Table 5. The boomerang trail is satisfied with probability 2^{-42} thus the previous attack requires 2^{44} queries.

We attack in the chosen-ciphertext and adaptively chosen-plaintext model to save the probability loss of 2^{-12} by truncating the trail in the last round. By constructing the structure for 2 bytes, the data complexity can further be reduced by a factor of $2^{12-2} = 2^{10}$. In summary, the 10-round distinguisher can be improved to $(Time, Data, Memory) = (2^{22}, 2^{22}, 2^{17})$.

11-Round Distinguisher. Recall the previous 11-round distinguisher that uses the trail in Table 6. The boomerang trail is satisfied with probability 2^{-120} thus the previous attack requires 2^{122} queries.

We attack in the chosen-ciphertext and adaptively chosen-plaintext model to save the probability loss of 2^{-12} by truncating the trail in the last round. By exploiting the structure for 2 bytes, the 11-round distinguisher can be improved to $(Time, Data, Memory) = (2^{100}, 2^{100}, 2^{17})$.

12-Round Key Recovery. We append 1 round to the end of 11-round distinguisher in Table 6, which is illustrated in Fig. 3. Differently from the 10-round

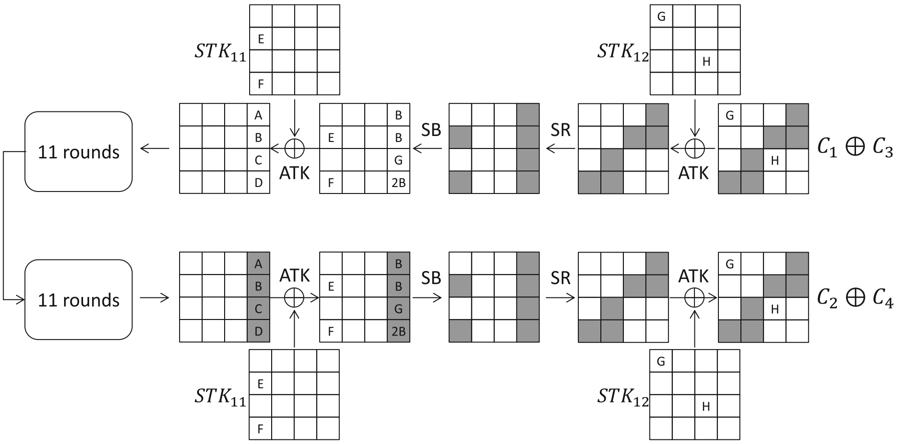

Fig. 3. Key recovery for 12-round Deoxys-BC-384.

key recovery attack against Deoxys-BC-256, the appended round only involves 6 bytes of STK_{12}.

Because the attack is the iteration for the 11-round distinguisher for exhaustive guesses of 6 subtweakey bytes in the key-recovery part, the attack complexity becomes $(Time, Data, Memory) = (2^{148}, 2^{148}, 2^{17})$ or $(2^{148}, 2^{100}, 2^{100})$.

6 Discussion and Conclusion

Discussion. The following features can be extracted by comparing the attacks in Sects. 4 and 5.

- Active byte positions at the input and output of the distinguisher significantly impact to the complexity of the key-recovery. If multiple active bytes locate in the same column in subsequent `MixColumns` like Fig. 3 rather than Fig. 2, the complexity of key recovery is much smaller.
- The attack complexity is reduced if the positions of subtweakey differences for subsequent `AddRoundTweakey` overlap with active bytes in the state.
- The existence of probabilistic propagation in the first and the last round of the distinguisher allows the attacker to optimize the attack. In contrast, it is hard to exploit probabilistic propagation in middle rounds.

The differential search by Cid et al. [7] did not consider those features and they found a lot of the best trails with the same score. Developing a new automated search method considering those features is a promising future research direction, which may allow us to identify the really best differential trail.

Concluding Remarks. In this paper, we showed how the attacks on Deoxys-BC can be optimized by considering the design details. Our attacks are based on the boomerang trails discovered by Cid et al. [7], but the attack procedures are carefully chosen to reduce the complexity. In particular, the improvement is big for Deoxys-BC-256. The complexity improvement with respect to $\max(Time, Data, Memory)$ is from 2^{74} to 2^{28} for the 8-round distinguisher, from 2^{124} to 2^{98} for the 9-round distinguisher, from 2^{118} to 2^{112} for the 9-round key recovery, and from 2^{204} to 2^{170} for the 10-round key recovery. We believe that the presented analyses provide better understanding of Deoxys-BC.

A Details of Boomerang Trails

Table 4. 9-round distinguisher of Deoxys-BC-256

ΔTK_0^1 : `00 7f 00 00  00 ff 00 00  0b 00 f1 00  00 00 00 7c`
ΔTK_0^2 : `00 cf 00 00  00 3f 00 00  70 00 5e 00  00 00 00 be`
∇TK_0^1 : `00 00 00 00  00 a1 00 04  00 00 00 00  00 00 00 00`
∇TK_0^2 : `00 00 00 00  00 bf 00 a8  00 00 00 00  00 00 00 00`

R	before ATK	ΔSTK	before SB	before MC	p_r
1	00 00 7b 00 b0 c0 00 00 00 00 af 00 61 00 00 c2	00 00 7b 00 b0 c0 00 00 00 00 af 00 00 00 00 c2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	e0 80 00 00 00 4d 00 00 00 00 00 00 00 00 00 ea	e0 80 00 00 00 4d 00 00 00 00 00 00 00 00 00 ea	b4 c9 00 00 21 00 00 00 00 00 00 00 73 00 00 00	2^{-28}
3	63 89 00 00 85 c9 00 00 00 c9 00 00 00 40 00 00	00 89 00 00 85 00 00 00 00 c9 00 00 00 40 00 00	63 00 00 00 00 c9 00 00 00 00 00 00 00 00 00 00	8d 00 00 00 8c 00 00 00 00 00 00 00 00 00 00 00	2^{-14}
4	8e 00 00 00 8e 00 00 00 01 00 00 00 00 00 00 00	8e 00 00 00 8e 00 00 00 01 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
5	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 80 03 13 00 00 00 00 98 00 00	00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00	1
6	00 00 00 00 00 00 00 00	00 07 00 35 00 b4 00 00	00 00 00 00 00	00 00 00 00 00	1

5			00 00 00 00 00	00 00 00 00 00	1
6	00 00 32 00 05 00 00 00	00 00 00 00 05 00 00 00	00 00 00 32 00 00 00 00 00 00 00	00 00 00 2f 00 00 00 00 00 00 00	2^{-7}
7	00 00 00 00 06 00 00 00 00 00 00 00 71 00 00 00	00 00 00 00 06 00 00 00 00 00 00 00 71 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
9	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 e3 00 00 00 00 00 00 00 0c 00 00	00 00 00 00 00 e3 00 00 00 00 00 00 00 0c 00 00	00 00 00 00 72 00 00 00 00 00 00 00 00 00 9d 00	2^{-12}

Table 5. 10-round distinguisher of Deoxys-BC-384. The S-box switch is used in round 6 (lower) for the S-box at position (1,1).

ΔTK_0^1 : 00 00 8b 00 / 00 00 00 90 / 90 00 00 00 / 00 1b 00 00
ΔTK_0^2 : 00 00 21 00 / 00 00 00 63 / 63 00 00 00 / 00 42 00 00
ΔTK_0^3 : 00 00 34 00 / 00 00 00 7d / 7d 00 00 00 / 00 49 00 00
∇TK_0^1 : 00 00 00 00 / 00 00 00 6e / 00 00 00 00 / b1 00 00 00
∇TK_0^2 : 00 00 00 00 / 00 00 00 42 / 00 00 00 00 / f5 00 00 00
∇TK_0^3 : 00 00 00 00 / 00 00 00 b3 / 00 00 00 00 / d3 00 00 00

R	before ATK	ΔSTK	before SB	before MC	p_r
1	00 00 8e 00 a3 00 00 10 9e 00 00 00 00 8e 00 00	00 00 8e 00 00 00 00 10 9e 00 00 00 00 8e 00 00	00 00 00 00 a3 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 69 00 00 00 00 00 00 00 00	2^{-6}
2	00 00 00 bb 00 00 00 d2 00 00 00 69 00 00 00 69	00 00 00 bb 00 00 00 d2 00 00 00 69 00 00 00 69	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
3	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
5	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	69 00 00 00 00 bb 00 00 00 00 d2 00 00 00 00 69	00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00	1
6	00 00 00 00 00 00 00 00 00 00 00 00	10 00 00 9e 00 00 8e 00 00 8e 00 00	00 00 **9e** 00 00 00 00 00	00 00 00 00 00 00 00 00	1
5			00 00 00 00	00 00 00 00	1
6	00 00 00 00	00 00 00 00	00 00 00 **9e** 00 00 00 00	00 00 00 00 68 00 00 00 01 00 00 00 b9 00 00 00	2^{-6}
7	00 00 00 00 6a 00 00 00 ba 00 00 00 00 00 00 00	00 00 00 00 6a 00 00 00 ba 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
9	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 6a ba 00 00 00	00 00 00 00 00 00 00 00 00 00 00 6a ba 00 00 00	00 00 00 00 00 00 00 00 00 61 00 00 00 97 00 00	2^{-12}

Table 6. 11-round distinguisher of Deoxys-BC-384

$\varDelta TK_0^1$: 00 8b 00 00 c4 00 00 00 7a 00 c5 a6 00 00 00 00
$\varDelta TK_0^2$: 00 ad 00 00 c4 00 00 00 73 00 21 d8 00 00 00 00
$\varDelta TK_0^3$: 00 a3 00 00 9a 00 00 00 3b 00 0d 2e 00 00 00 00
∇TK_0^1 : 00 00 02 00 00 00 00 00 d7 00 00 00 00 00 00 00
∇TK_0^2 : 00 00 99 00 00 00 00 00 bc 00 00 00 00 00 00 00
∇TK_0^3 : 00 00 0c 00 00 00 00 00 f1 00 00 00 00 00 00 00

R	before ATK	$\varDelta STK$	before SB	before MC	p_r
1	00 9a 32 00 85 00 00 00 00 00 e9 00 00 00 50 00	00 9a 32 00 85 00 00 00 00 00 e9 00 00 00 50 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
2	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
3	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 4f f1 7a 00 00 00 57 00 00	00 00 00 00 00 00 00 4f f1 7a 00 00 00 57 00 00	00 00 00 00 00 00 2a 00 00 00 15 a6 00 00 6b 00	2^{-28}
4	00 00 00 a6 00 00 00 f1 00 00 bd 57 00 00 e9 a6	00 00 00 a6 00 00 00 f1 00 00 00 57 00 00 e9 00	00 00 00 00 00 00 00 00 00 00 bd 00 00 00 00 a6	00 00 00 00 00 00 00 00 19 00 00 00 2b 00 00 00	2^{-13}
5	32 00 00 00 00 00 00 00 4f 00 00 00 4f 00 00 00	32 00 00 00 00 00 00 00 4f 00 00 00 4f 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
6	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 85 00 00 00 00 b9 00 00 00 00 9a 34 00 00	00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	1
7	00 00 00 00 00 00 00 00	00 08 00 00 00 09 00 1b	00 00 00 00 00	00 00 00 00 00	1
6			00 00 00 00 00	cb 00 ff 00 1a 00 00 00	1
7	8d 00 00 00 00 00 a3 00	8d 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 a3 00 00	00 00 00 00 00 00 00 00 00 b5 00	2^{-7}
8	00 00 00 00 00 00 c4 00 00 00 00 00 00 00 05 00	00 00 00 00 00 00 c4 00 00 00 00 00 00 00 05 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
9	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	1
11	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 05 00 00 00 00 00 c4 00 00 00 00 00 00	00 00 00 05 00 00 00 00 00 c4 00 00 00 00 00 00	00 00 00 08 00 00 00 00 00 00 00 7f 00 00 00 00	2^{-12}

References

1. Beierle, C., et al.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 123–153. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_5
2. Bernstein, D.: CAESAR competition (2013). http://competitions.cr.yp.to/caesar.html
3. Biham, E., Dunkelman, O., Keller, N.: The rectangle attack – rectangling the serpent. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 340–357. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_21
4. Biham, E., Dunkelman, O., Keller, N.: New results on boomerang and rectangle attacks. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 1–16. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45661-9_1
5. Biham, E., Dunkelman, O., Keller, N.: Related-key boomerang and rectangle attacks. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_30
6. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_1
7. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: A security analysis of deoxys and its internal tweakable block ciphers. IACR Trans. Symmetric Cryptol. **2017**(3), 73–107 (2017)
8. Dunkelman, O., Keller, N., Shamir, A.: A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 393–410. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_21
9. Dunkelman, O., Keller, N., Shamir, A.: A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. J. Cryptol. **27**(4), 824–849 (2014)
10. Jean, J., Nikolić, I., Peyrin, T.: Tweaks and keys for block ciphers: the TWEAKEY framework. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 274–288. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_15
11. Jean, J., Nikolić, I., Peyrin, T., Seurin, Y.: Deoxys v1.41. Submitted to CAESAR, October 2016
12. Kelsey, J., Kohno, T., Schneier, B.: Amplified boomerang attacks against reduced-round MARS and Serpent. In: Goos, G., Hartmanis, J., van Leeuwen, J., Schneier, B. (eds.) FSE 2000. LNCS, vol. 1978, pp. 75–93. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44706-7_6
13. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_3
14. Liu, G., Ghosh, M., Ling, S.: Security analysis of SKINNY under related-tweakey settings (long paper). IACR Trans. Symmetric Cryptol. **2017**(3), 37–72 (2017). https://doi.org/10.13154/tosc.v2017.i3.37-72
15. McGrew, D.A., Viega, J.: The security and performance of the Galois/Counter Mode (GCM) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343–355. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30556-9_27

16. Mehrdad, A., Moazami, F., Soleimany, H.: Impossible differential cryptanalysis on Deoxys-BC-256. Cryptology ePrint Archive, Report 2018/048 (2018). https://eprint.iacr.org/2018/048
17. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C.-K., Yung, M., Lin, D. (eds.) Inscrypt 2011. LNCS, vol. 7537, pp. 57–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34704-7_5
18. Murphy, S.: The return of the cryptographic boomerang. IEEE Trans. Inf. Theory **57**(4), 2517–2521 (2011). https://doi.org/10.1109/TIT.2011.2111091
19. National Institute of Standards and Technology: Federal Information Processing Standards Publication 197: Advanced Encryption Standard (AES). NIST, November 2001
20. Wagner, D.: The boomerang attack. In: Knudsen, L. (ed.) FSE 1999. LNCS, vol. 1636, pp. 156–170. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48519-8_12