IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2017, No. 3, pp. 73–107. DOI:10.13154/tosc.v2017.i3.73-107

A Security Analysis of Deoxys and its Internal Tweakable Block Ciphers

Carlos Cid[1], Tao Huang[2], Thomas Peyrin[2,3,4], Yu Sasaki[5] and Ling Song[2,3,6]

[1] Information Security Group, Royal Holloway University of London, Egham, United Kingdom
carlos.cid@rhul.ac.uk
[2] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore
huangtao@ntu.edu.sg,songling@iie.ac.cn,thomas.peyrin@ntu.edu.sg
[3] Temasek Laboratories, Nanyang Technological University, Singapore, Singapore
[4] School of Computer Science and Engineering, Nanyang Technological University, Singapore, Singapore
[5] NTT Secure Platform Laboratories, Tokyo, Japan
sasaki.yu@lab.ntt.co.jp
[6] State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

Abstract. In this article, we provide the first independent security analysis of Deoxys, a third-round authenticated encryption candidate of the CAESAR competition, and its internal tweakable block ciphers Deoxys-BC-256 and Deoxys-BC-384. We show that the related-tweakey differential bounds provided by the designers can be greatly improved thanks to a Mixed Integer Linear Programming (MILP) based search tool. In particular, we develop a new method to incorporate linear incompatibility in the MILP model. We use this tool to generate valid differential paths for reduced-round versions of Deoxys-BC-256 and Deoxys-BC-384, later combining them into broader boomerang or rectangle attacks. Here, we also develop a new MILP model which optimises the two paths by taking into account the effect of the ladder switch technique. Interestingly, with the tweak in Deoxys-BC providing extra input as opposed to a classical block cipher, we can even consider beyond full-codebook attacks. As these primitives are based on the TWEAKEY framework, we further study how the security of the cipher is impacted when playing with the tweak/key sizes. All in all, we are able to attack 10 rounds of Deoxys-BC-256 (out of 14) and 13 rounds of Deoxys-BC-384 (out of 16). The extra rounds specified in Deoxys-BC to balance the tweak input (when compared to AES) seem to provide about the same security margin as AES-128. Finally we analyse why the authenticated encryption modes of Deoxys mostly prevent our attacks on Deoxys-BC to apply to the authenticated encryption primitive.

Keywords: Deoxys-BC · AES · authenticated encryption · block cipher · differential cryptanalysis · boomerang attack · MILP · linear incompatibility · ladder switch

1 Introduction

Authenticated Encryption (AE) schemes are symmetric-key cryptographic algorithms that provide both confidentiality and authenticity of data in one single primitive. AE schemes offer several advantages when compared with the use of two separate algorithms (e.g. AES combined with HMAC) for securing digital communications: it typically gives rise to more efficient and compact constructions, it simplifies key management, and it may allow more refined security arguments. A popular AE scheme is McGrew and Viega's Galois/Counter Mode (GCM) [MV05], which has been standardised by NIST [Nat07] and is widely deployed.

Received: 2017-06-01, Accepted: 2017-08-01, Published: 2017-09-19

The recent growing interest in new AE schemes resulted in the launch in 2013 of CAESAR, a competition organised by the international cryptologic research community to identify a portfolio of authenticated ciphers that offer advantages over AES-GCM and are suitable for widespread adoption[1]. The competition received 57 submissions in March 2014, with 30 candidates advancing to the second round in 2015. In August 2016, the competition selected 15 third-round candidates. The final portfolio is expected in late 2017. Deoxys is one of the CAESAR third-round authenticated encryption candidates [JNPS16]. Its design is based on the tweakable block cipher Deoxys-BC, used in two different fully parallel and provably secure authenticated encryption modes: one for which the nonce must not be reused, the other one providing security even when the nonce is reused.

Deoxys-BC is an AES-based tweakable block cipher, based on the TWEAKEY framework [JNP14]. Tweakable block ciphers (TBC) were first introduced and formalised by Liskov et al. [LRW02], and in addition to the two standard inputs, a plaintext and a key, it takes an additional input called a *tweak*. Tweakable block ciphers are popular primitives for constructing authenticated encryption schemes, and the Deoxys AE scheme makes use of two versions of the cipher as its internal primitive: Deoxys-BC-256 and Deoxys-BC-384.

Most tweakable block cipher constructions take an existing block cipher (or permutation) as a black box, and use the tweak to modify the input/output of the cipher. In contrast, the TWEAKEY framework proposes a novel approach: it unifies the vision of key and tweak inputs of a cipher, as the *tweakey*. This allows one to add a tweak of (almost) any length to a key-alternating block cipher and/or to extend the key space of the block cipher to (almost) any size: 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, which in turn translates to the AE scheme when employing a provable secure authenticated encryption mode. Besides Deoxys-BC, two other AES-like tweakable block ciphers were also introduced in [JNP14]: Joltik-BC and Kiasu-BC. Similarly to Deoxys-BC, these ciphers were also used as the internal primitives of two CAESAR submissions (Joltik and Kiasu, respectively), although neither were selected for the third round. Other examples of block ciphers adopting the TWEAKEY framework include SKINNY, MANTIS [BJK+16] and QARMA [Ava17].

The only existing public security analysis of the Deoxys-BC block cipher is the one provided by the designers [JNP14, JNPS16]. 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. When considering differential cryptanalysis, the designers provide in [JNPS16] upper bounds on the probability of the best round-reduced related-key related-tweak differential paths for both Deoxys-BC-256 and Deoxys-BC-384. For Deoxys-BC-256, it is shown that the number of active S-boxes for 10 rounds is lower-bounded by 22, meaning that the probability of the associated differential path is upper-bounded by 2^{-132} (the maximal differential probability of the AES S-box being 2^{-6}). Similarly, for Deoxys-BC-384 at least 22 S-boxes are active after 12 rounds. This led the designers to claim that "*all versions of* Deoxys-BC *(used in* Deoxys*) have a security margin of at least four rounds and thus [are] highly resistant against related-key related-tweak attacks*" [JNPS16]. They also briefly discuss linear cryptanalysis, meet-in-the-middle attacks, among other attacks, noting however that "*all the attacks that do not exploit the key schedule will have the same success on* Deoxys-BC *as on* AES".

Our contributions. In this paper we provide the first independent security analysis of Deoxys and its internal tweakable block ciphers Deoxys-BC-256 and Deoxys-BC-384. First, we exhibit greatly improved lower bounds for the number of active S-boxes for both

[1] https://competitions.cr.yp.to/caesar.html

versions of Deoxys-BC, by performing a special MILP-based search, taking into account the features of the cipher, in particular its linear tweakey schedule and linear incompatibilities [FJP13] between differential propagations in the tweakey and the state.[2] The obtained bounds are given in Table 1. For example, for 10 rounds of Deoxys-BC-256 we can prove 31 actives S-boxes while the Deoxys designers only proved 22. Similarly, for 12 rounds of Deoxys-BC-384 we can prove 35 actives S-boxes while the Deoxys designers only proved 22. Not only the bounds are improved, but our tool also covers many more rounds than what the Deoxys designers could achieve.

Table 1: Old and new lower bounds on the number of active S-boxes for Deoxys-BC.

Deoxys-BC-256

types of lower bounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14
in spec. ([JNPS16])	0	0	1	5	9	12	16	17	-	22	-	-	-	-
simple model	0	0	1	5	9	12	16	19	23	26	29	32	35	38
linear incompatibility†	0	0	1	5	10	14	18	22	27	31	35	40	44	48

Deoxys-BC-384

types of lower bounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
in spec. ([JNPS16])	0	0	0	1	4	8	-	-	-	-	-	22	-	-	-	-
simple model	0	0	0	1	4	8	10	14	18	21	24	28	31	35	37	45
linear incompatibility†	0	0	0	1	5	9	13	18	22	27	31	35	40	44	48	52

† Bounds for linear incompatibility are obtained under different assumptions from other standard researches using MILP. More details will be explained in Sect. 3.3.

Since the MILP tool is not suitable for finding exact differential paths where 8-bit S-boxes are used, but it is rather good at generating active-byte patterns in truncated differential paths, we design a dedicated algorithm for searching differential paths which works given a byte-wise active pattern. Then, combining the MILP tool and the dedicated algorithm, we generate valid differential paths for reduced-round versions of Deoxys-BC-256 and Deoxys-BC-384. These differential paths can in turn be employed in a broader attack process, such as boomerang or rectangle attacks. In particular, we develop a new MILP model which optimises two paths by taking into account the effect of the ladder switch proposed by [BK09]. We study how these attacks can apply to Deoxys-BC. We remark that one can potentially use the tweak input to help in applying differential attacks and gain a few more rounds when compared to a classical block cipher. These so-called *beyond full-codebook* attacks have shown to be powerful and realistic against other tweakable block ciphers [BHT16, DV17]. Our work gives some insight into the extent to which adding an extra tweak input to a block cipher can reduce the security margin.

All in all, we are able to attack 10 rounds of Deoxys-BC-256 and 13 rounds of Deoxys-BC-384 (compared to the best previous attacks reaching only 8 rounds for both ciphers [JNPS16]). We have verified our cryptanalysis work by conducting practical experiments; all our results are summarised in Table 2.

Finally, we discuss how these attacks on the internal tweakable block ciphers can be applied to the entire AE scheme. We argue that our attacks are difficult to extend to

[2] Two types of bounds are provided in this paper. The first type using simple model is obtained in the same assumption as previous work. The second type using linear incompatibility requires an additional assumption that truncated differential paths can only be satisfied when the degrees of freedom with respect to the choice of differences is greater than or equal to the degrees of consumption.

Deoxys-II, but some of them can be applicable to Deoxys-I under certain conditions.

Table 2: Previous and new cryptanalysis results on Deoxys-BC-256 and Deoxys-BC-384 (top table, Section 4), as well as on the four Deoxys CAESAR candidates Deoxys-I-128-128, Deoxys-II-128-128, Deoxys-I-256-128 and Deoxys-II-256-128 (bottom table, Section 6). The attack marked with ($\star$) has a success probability of about 75% compared to about 95% for the other attacks.

Deoxys **internal primitives**

	number of rounds	tweak size	key size	time	data	memory	attack type	ref.
Deoxys-BC-256	8/14	128	128	$\leq 2^{128}$	-	-	MitM	[JNPS16]
	$\leq 8/14$	128	128	$\leq 2^{128}$	-	-	differential	[JNPS16]
	9/14	128	128	2^{118}	2^{117}	2^{117}	rectangle	this paper
	10/14	$t < 52$	$k > 204$	2^{204}	$2^{127.58}$	$2^{127.58}$	rectangle	this paper
Deoxys-BC-384	8/16	128	256	$\leq 2^{256}$	-	-	MitM	[JNPS16]
	12/16	128	256	2^{127}	2^{127}	2^{125}	rectangle	this paper
	13/16	$t < 114$	$k > 270$	2^{270}	2^{127}	2^{144}	rectangle	this paper

Deoxys **AE schemes**

Deoxys-I-128-128	9/14	-	128	2^{118}	2^{117}	2^{117}	rectangle	this paper
Deoxys-II-128-128	-	-	128	-	-	-	-	-
Deoxys-I-256-128	12/16	-	256	2^{236}	2^{126} $\star$	2^{124}	rectangle	this paper
Deoxys-II-256-128	-	-	256	-	-	-	-	-

Outline. In Section 2 we provide the specification of Deoxys and Deoxys-BC. In Section 3 we describe our new MILP model incorporating linear incompatibility for Deoxys-BC and provide our improved bounds for the number of active S-boxes for Deoxys-BC-256 and Deoxys-BC-384. In Section 4 we describe our dedicated algorithm for finding differential paths and then study the application of rectangle and boomerang attacks against Deoxys-BC-256 and Deoxys-BC-384; we discuss that these attacks can potentially be extended to beyond full-codebook boundaries in Section 5. Finally, we analyse in Section 6 why these findings on Deoxys-BC seem generally difficult to extend to Deoxys versions where Deoxys-BC is plugged into the AEAD modes.

2 Description of Deoxys and Deoxys-BC

Deoxys-BC is an AES-based tweakable block cipher [JNPS16], based on the TWEAKEY framework [JNP14]. The Deoxys authenticated encryption scheme makes use of two versions of the cipher as its internal primitive: Deoxys-BC-256 and Deoxys-BC-384. Both versions are ad-hoc 128-bit tweakable block ciphers which besides the two standard inputs, a plaintext P (or a ciphertext C) and a key K, also take an additional input called 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. In Deoxys, the size of the key and tweak can vary within the tweakey boundaries, as long as the key size is greater or equal to the block size, i.e. 128 bits. In this section we recall the details of the Deoxys-BC block cipher and the Deoxys AEAD operating modes. We assume that the reader is familiar with the AES block cipher [Nat01].

Deoxys-BC is an AES-like design, i.e. it is an iterative substitution-permutation network (SPN) that transforms the initial plaintext (viewed as a 4×4 matrix of bytes) using

the AES round function, with the main differences with AES being the number of rounds and the round subkeys that are used every round. Deoxys-BC-256 has 14 rounds, while Deoxys-BC-384 has 16 rounds.

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

- AddRoundTweakey – XOR the 128-bit round subtweakey (defined below) to the internal state.
- SubBytes – Apply the 8-bit AES S-box $\mathcal{S}$ to each of the 16 bytes of the internal state.
- ShiftRows – Rotate the 4-byte i-th row left by $\rho[i]$ positions, where $\rho = (0, 1, 2, 3)$.
- MixColumns – Multiply the internal state by the 4×4 constant MDS matrix of AES.

After the last round, a final AddRoundTweakey operation is performed to produce the ciphertext.

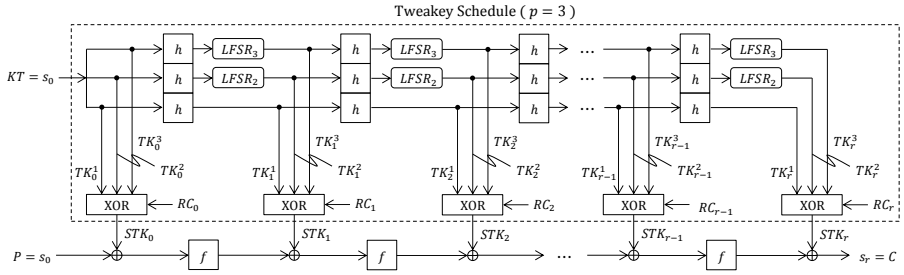

Figure 1: Instantiation of the TWEAKEY framework for Deoxys-BC-384.

Definition of the Subtweakeys. We denote the concatenation of the key K and the tweak T as KT, i.e. $KT = K||T$. The *tweakey* state is then divided into 128-bit words. More precisely, in Deoxys-BC-256 the size of KT is 256 bits with the first (most significant) 128 bits of KT being denoted W_2; the second word is denoted by W_1. For Deoxys-BC-384, the size of KT is 384 bits, and we denote the first (most significant), second and third 128-bit words of KT by W_3, W_2 and W_1, respectively. 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

$$TK_{i+1}^1 = h(TK_i^1), \quad TK_{i+1}^2 = h(LFSR_2(TK_i^2)), \quad TK_{i+1}^3 = h(LFSR_3(TK_i^3)),$$

where the byte permutation h is defined as

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

with the 16 bytes of a 128-bit tweakey word numbered by the usual AES byte ordering.

Table 3: The two LFSRs used in Deoxys-BC tweakey schedule.

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

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 3 (x_0 stands for the LSB of the cell).

Finally, RC_i denotes the key schedule round constants, of which we omit the details. A schematic diagram of the instantiation of the TWEAKEY framework for Deoxys-BC is shown in Figure 1.

We note that when active byte positions in the 16-byte words TK_0^1, TK_0^2, and TK_0^3 are specified, active byte positions in TK_*^1, TK_*^2, and TK_*^3 are then uniquely determined. We introduce the terminology "lane i" to denote the ith byte of TK_0^1, TK_0^2 or TK_0^3 and the corresponding byte position in TK_*^1, TK_*^2 or TK_*^3.

The Deoxys AEAD operating modes. Based on the Deoxys-BC-256 and Deoxys-BC-384 block ciphers, the Deoxys designers proposed two AEAD modes, Deoxys-I and Deoxys-II. The first mode, Deoxys-I, is a nonce-based AEAD, to be used in a nonce-respecting setting. The second mode, Deoxys-II, is a nonce-based AEAD scheme that provides security even in the nonce-misuse setting. We refer to the Deoxys submission document [JNPS16] for full specification details.

With the recommended parameters given in [JNPS16], when instantiated with the Deoxys-BC-256 block cipher, the two AE modes lead to a 128-bit key version (denoted Deoxys-I-128-128 and Deoxys-II-128-128), while when using Deoxys-BC-384, they lead to a 256-bit key version (Deoxys-I-256-128 and Deoxys-II-256-128). We note that for all versions of Deoxys, a message cannot exceed 2^{60} blocks, while a maximum of 2^{64} messages can be ciphered under the same secret key. Therefore, when attacking one of the Deoxys versions, an adversary can obtain at most 2^{124} blocks of data under the same key.

3 Improved Security Bounds for Deoxys-BC

Proving the lower bound on the number of active S-boxes for Deoxys-BC in the single-key setting is straightforward: it is identical to the AES owing to the same round function. Proving bounds in the related-tweakey setting is more challenging. In [JNPS16], the designers evaluated lower bounds with Matsui's algorithm [BN10], split approach [BN11], and extended split approach [ELN+14]; these are shown in Table 1. They showed the number of rounds to activate at least 22 active S-boxes, which ensures the maximum probability of 2^{-132} and is thus unlikely to be satisfied even with the full codebook. However, Deoxys accepts a tweak which can be used by the attacker to increase the attack data resources. We note that trying 2^{132} plaintext/tweak pairs to satisfy the paths with probability 2^{-132} does not distinguish the cipher immediately, because we obtain 2^4 wrong pairs as well as 1 right pair. However, it is still possible to spend extra effort to analyze those $2^4 + 1$ candidates to identify the right pair, considering that $2^4 + 1$ candidates are significantly smaller than 2^{132} pairs. This fact motivates us to derive lower bounds for an even higher number of rounds.

In this section, we first briefly explain how to search for differential bounds with mixed integer linear programming (MILP). We then explain the simple application to Deoxys-BC in Section 3.2. The simple application gives us only rough bounds. The main contributions

of this section are the much tighter lower bounds obtained by incorporating degrees of freedom with respect to differences, which is explained in Section 3.3.

3.1 Brief Introduction of MILP for Differential Bound Search

Mixed integer linear programming (MILP) is a general mathematical tool, which takes an objective function and a system of linear inequalities with respect to real numbers as input, and searches for an optimal solution which minimises/maximises the objective function satisfying all the inequalities. Mouha et al. [MWGP11] showed that the problem of finding the optimal differential path can be converted to MILP.

Single-Key for AES. The internal state of AES is represented by 16 bytes per round. To find r-round truncated differential paths with the minimum number of active S-boxes, one defines $16r$ variables $x_i \in \{0,1\}$, in which $x_i = 1$ denotes that the ith byte is active and $x_i = 0$ denotes that the ith byte is inactive. The objective function becomes "minimise Σx_i." To exclude solutions with invalid differential propagation, the property of branch number 5 of MixColumns needs to be represented as a system of inequalities. By introducing another dummy variable, $d_j \in \{0,1\}$ for column j, Mouha et al. expressed the constraints of the branch number with nine inequalities per column. For example, suppose that 4 bytes (corresponding to x_0, x_5, x_{10}, x_{15}) are processed by MixColumns and are updated to 4 bytes (corresponding to $x_{16}, x_{17}, x_{18}, x_{19}$). Then, valid active patterns for $x_0, x_5, x_{10}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}$ can be expressed as

$$\begin{aligned} x_0 + x_5 + x_{10} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} \geq 5d_j, \\ d_j \geq x_0,\ d_j \geq x_5,\ d_j \geq x_{10},\ d_j \geq x_{15},\ d_j \geq x_{16},\ d_j \geq x_{17},\ d_j \geq x_{18},\ d_j \geq x_{19}. \end{aligned} \quad (1)$$

The constraints encoding valid differential propagations for the entire cipher (in the single-key setting) can be constructed by iterating the above nine inequalities for all columns and for all rounds, which results in $9 \cdot 4 \cdot r$ inequalities.

Related-Tweakey with TK^1. The extension to related-tweakey with one tweakey state is simple. One can define 16 binary variables $stk_0, stk_1, \cdots, stk_{15}$ to represent whether the corresponding subtweakey byte is active or not. Let $stk_{16r}, stk_{16r+1}, \cdots, stk_{16r+15}$ be the 16 variables for the subtweakey after r rounds. Activeness of $stk_{16r}, stk_{16r+1}, \cdots, stk_{16r+15}$ for $r \geq 1$ is uniquely determined accordingly to the tweakey permutation h and the active pattern for $r = 0$. Namely, $stk_{16r+j} = stk_{16(r-1)+h(j)}$ for $r \geq 1$ and $j = 0, 1, \cdots, 15$.

To model the AddRoundKey operation, one can introduce 16 further variables y_{16r+j} per round to denote the state after AddRoundKey. Suppose that the ith byte of the state (corresponding to x_i) and the ith byte of subtweakey (corresponding to stk_i) are xored to compute the new ith byte of the state (corresponding to y_i). Then, for (x_i, stk_i, y_i) where $i = 16r + j$, we need to exclude $(x_i, stk_i, y_i) \in \{(0,0,1), (0,1,0), (1,0,0)\}$ from the solution space, which can be done with one inequality per pattern:

$$x_i + stk_i - y_i \geq 0, \qquad x_i - stk_i + y_i \geq 0, \qquad -x_i + stk_i + y_i \geq 0. \quad (2)$$

Related-Tweakey with TK^2 and TK^3. Modelling multiple tweakey states is more complex, owing to the bit-level update in TK^2 and TK^3 by the LFSRs. The natural extension, making a bit-wise model, using $128 + 256$ variables and $128 + 384$ variables per round for Deoxys-BC-256 and Deoxys-BC-384 respectively, is however too expensive and the system soon becomes infeasible when r grows. The designers of SKINNY [BJK+16] showed that TK^2 and TK^3 can be efficiently modelled at the byte-level. Suppose that a single byte of TK_0^1 and TK_0^2 in the same position are active in the 256-bit tweakey, and let a and b be the differences of those bytes. Then ΔSTK_0 is $a \oplus b$ in this byte. In the next

round, the byte position changes with h, and then b is updated with $LFSR_2$, which makes the difference of the next round tweakey $a \oplus LFSR_2(b)$. Similarly, by ignoring the position update, the round tweakey difference is computed by $a \oplus (LFSR_2)^2(b), a \oplus (LFSR_2)^3(b)$ and so on. Given the fact that the minimum (non-trivial) cycle of $LFSR_2$ has length 15 (except the cycle length corresponding to the all-zero state), the number of cancellations between those two bytes is at most 1 in every 15 rounds. Let LANE_i, where $i = 0, \cdots, 15$, be 16 binary variables to represent that at least one among the ith bytes of TK_0^1 and TK_0^2 is active. Let h_{inv} be the inverse of h. We then obtain the following constraints for $r \leq 15$:

$$\begin{aligned} \mathrm{LANE}_i - stk_i \geq 0,\ \mathrm{LANE}_i - stk_{16+h_{inv}(i)} \geq 0,\ \cdots,\ \mathrm{LANE}_i - stk_{16(r-1)+h_{inv}^{r-1}(i)} \geq 0, \\ stk_i + stk_{16+h_{inv}(i)} + stk_{32+h_{inv}^2(i)} + \cdots + stk_{16(r-1)+h_{inv}^{r-1}(i)} \geq r \cdot \mathrm{LANE}_i - 1. \end{aligned} \tag{3}$$

The model can simply be extended to TK_3. The only difference from TK_2 is that the cancellation can occur up to twice for each LANE_i owing to additional degrees of freedom of differences in the state TK_0^3 (see [JNP14] for more details). As a result, the only difference will be the modifying of Eq. (3) as

$$stk_i + stk_{16+h_{inv}(i)} + stk_{32+h_{inv}^2(i)} + \cdots + stk_{16(r-1)+h_{inv}^{r-1}(i)} \geq r \cdot \mathrm{LANE}_i - 2. \tag{4}$$

3.2 Simple Application to Deoxys and Limitations

The MILP model for the related-tweakey setting in [BJK+16] can be simply applied to both `Deoxys-BC-256` and `Deoxys-BC-384`, which already provides better lower bounds than the ones in [JNPS16]. They are listed in Table 1 in the row "simple model."

Recall that the designers focused on the number of rounds to have at least 22 active S-boxes. For both of `Deoxys-BC-256` and `Deoxys-BC-384`, the new bounds ensure 22 active S-boxes with one fewer round than the designers' evaluation, which implies that the security of `Deoxys-BC-256` and `Deoxys-BC-384` against related-tweakey differential attacks is higher than the original expectation.

We note that the simple application of the previous method in [BJK+16] cannot be applied to 16-rounds of `Deoxys-BC-384` directly. Because the cycle of the LFSRs is 15, subtweakey differences in round 0 and round 15 become identical, namely $\Delta STK_0 = \Delta STK_{15}$. Then ΔSTK_{15} must be modelled by the following modifications:

1. Add an inequality for $\Delta STK_0 = \Delta STK_{15}$ per byte, namely $stk_i = stk_{16\cdot 15+h_{inv}^{15}(i)}$.
2. Replace the left-hand side of Eq. (4) by $\sum_{r=0}^{14} stk_{16r+h_{inv}^{r}(i)}$.

Linear Incompatibility. Those bounds are not tight, that is to say, they do not ensure truncated differential paths matching those bounds. Indeed, we tried to find differential paths satisfying the truncated differential paths, without success. In fact, we observed that all the truncated differential paths we tested included *linear incompatibility*, which was demonstrated by Fouque et al. against AES in [FJP13].

Intuitively, the observation by Fouque et al. is that the difference cancellation between the key state and the round state in some round, say round i, imposes some linear relationship between the key and state differences. Hence, difference cancellation in a different round, say round $i + 1$, cannot be independently simulated.

An example of linear incompatibility for 2-round AES is illustrated in Figure 2. This truncated differential path is correct if cancellation in round i, cancellation in round $i + 1$, and cancellation in the key schedule are independently considered. However, the cancellation in the key schedule determines that two bytes in each row of the key must have identical difference. This is incompatible with the cancellation in round $i + 1$ due to the property of MixColumns. We refer to [FJP13] for more details.

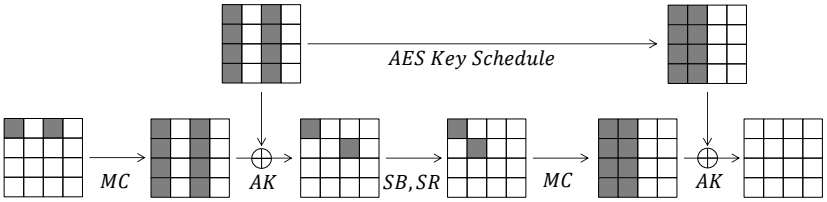

Figure 2: An example of linear incompatibility for 2-round AES in [FJP13, Figure 7].

3.3 Incorporating Degrees of Freedom and Consumption

In this section, we solve the problem of linear incompatibility. First, one may wonder whether the linear incompatibility could be solved by a two-stage search as in [SGL+17]. Namely, in the first stage, truncated differential paths with small number of active S-boxes (which may contain a linear incompatibility) are searched with a tool. Then in the second stage, differential paths with the highest probability are searched for each discovered truncated differential path until a feasible truncated differential path is found. This approach works well for a small number of rounds, indeed Sun et al. [SGL+17] found 6-round related-key differential paths for AES-128. However, the running time for each truncated differential path quickly increases when the number of rounds becomes large; moreover, the number of truncated differential paths to test is too high when the model of the first stage is loose.

The above discussion indicates the need of solving the linear incompatibility in the first stage. We explain how to model linear incompatibility for `Deoxys` in MILP.

3.3.1 Overall Idea

The overall idea is to include constraints stating that the degree of freedom of differences in the truncated differential paths is greater than or equal to the consumption of degrees of freedom.

Degrees of Freedom. Suppose that the size of the tweakey states is s words, namely $s = 2$ for `Deoxys-BC-256` and $s = 3$ for `Deoxys-BC-384`. Suppose that LANE_i is active, i.e. some of the s states in position i have non-zero difference. Those differences can be chosen independently, thus the total degrees of freedom of differences is $s \times \ell$ bytes, where ℓ is the number of active lanes in the key schedule.

Degrees of Consumption Type 1. Due to the property of the tweakey schedule, for each active lane, up to $s - 1$ subtweakey differences can be 0. Each cancellation of this type enforces a linear constraint of the form $TK^1[i] \oplus TK^2[i] = 0$ for `Deoxys-BC-256` and $TK^1[i] \oplus TK^2[i] \oplus TK^3[i] = 0$ for `Deoxys-BC-384`, in which '$[i]$' denotes the ith byte of the tweakey state. Consequently, every time a cancellation occurs to make a subtweakey difference 0, the path consumes one degree of freedom.

Degrees of Consumption Type 2. This is for the difference cancellation in the MixColumns and AddRoundKey operations, and is calculated column by column. After AddRoundKey and SubBytes, we assume that the state difference in each active byte is uniformly random. Then in the subsequent MixColumns (via ShiftRows), we count the number of inactive output bytes. The path consumes degrees of freedom for the number of inactive bytes. Subsequently, we count the number of difference cancellation in the next AddRoundKey operation. For each of the tweakey differences counted as degrees of freedom, the corresponding differences in subtweakeys are uniquely determined. Hence, to calculate degrees we regard

the subtweakey differences as fixed. Finally, the differential propagation consumes degrees of freedom for each of the cancellations during AddRoundKey.

In summary, let a, b, and c be the number of active bytes before MixColumns, the number of inactive bytes after MixColumns, and the number of cancellations during AddRoundKey, respectively, for each column. Consumption of degrees are calculated by $a - b - c$. If this is greater than or equal to 0, the path does not consume any degrees, while if it is below 0, the path consumes degrees by $-(a - b - c)$ bytes.

Example. We show an example of calculating degrees by using a 13-round truncated differential path for Deoxys-BC-384 ($s = 3$) shown in Figure 5 in Appendix A. The number of active lanes ℓ is 5, and thus the degrees of freedom of difference are $3 \times 5 = 15$ bytes. Cancellations in subtweakeys occur on 3 bytes in STK_1, 1 byte in STK_2, 2 bytes in STK_5, 1 byte in STK_6, and 1 byte in STK_{12}; hence degrees are consumed by 8 bytes due to type 1 consumption. Type 2 consumption first appears in the leftmost column in round 2. There are 2 active bytes before MixColumns, 1 inactive byte after MixColumns, and 2 bytes are cancelled for AddRoundKey. That is, $(a, b, c) = (2, 1, 2)$ and the degree is consumed by $-(2 - 1 - 2) = 1$ byte. The path consumes 7 bytes with type 2. In total, the degrees are $15 - 8 - 7 = 0$.

We would like to make two remarks about the relationship between remaining freedom degrees and validity of the truncated differential paths.

- Zero remaining degrees do not imply whether or not the truncated differential path can be instantiated with particular differences. Because we do not consider dependency between degrees of consumption in different columns, the path might be satisfied. Hence, we keep truncated differential paths with zero remaining degrees as candidates to achieve the minimum number of active S-boxes (and later explores the actual differences with the method that will be explained in Section 4.2).
- Even if the remaining degrees of freedom are greater than 0, it does not ensure that the truncated differential path can be instantiated with actual difference. For example, our experiment detected the case that the consumption of degrees are concentrated on a particular lane. Hence, even if the sum of degrees of freedom in all lanes are greater than the sum of degrees of consumption in all lanes, contradiction may occur for particular lanes.

3.3.2 Representing Degree Calculation in the MILP Model

Degrees of Freedom. Degrees of freedom are represented as $s \cdot \sum_{i=0}^{15} \mathrm{LANE}_i$.

Degrees of Consumption Type 1. Consumption of type 1 is calculated as the maximum number of active bytes in the subtweakey minus the number of actually activated bytes. The model is $r \cdot \sum_{i=0}^{15} \mathrm{LANE}_i - \sum_{i=0}^{16r-1} stk_i$, where r is the number of rounds to evaluate.

Degrees of Consumption Type 2. Type 2 consumption is more complicated. Suppose that y_0, y_5, y_{10}, y_{15} are input variables to MixColumns, $x_{16}, x_{17}, x_{18}, x_{19}$ are the corresponding output variables from MixColumns, $stk_{16}, stk_{17}, stk_{18}, stk_{19}$ are variables of STK_1 xored in the subsequent AddRoundKey, and $y_{16}, y_{17}, y_{18}, y_{19}$ are the outputs from AddRoundKey.

***a*: Number of active bytes before MixColumns.** This is represented by $y_0 + y_5 + y_{10} + y_{15}$.

***b*: Number of inactive bytes after MixColumns.** The natural model is $4 - x_{16} - x_{17} - x_{18} - x_{19}$. A large value of b implies high consumption of degrees in this column. However, if the column is inactive, this natural model sets b to 4, while the path actually does not consume any degree; thus in this case we need to set b to 0. Recall

that the previous model to describe MixColumns in Eq. (1) uses a dummy variable d_j for column j, which represents whether the column is active or not. Here, we introduce another dummy variable b_j and add an equality $b_j = 4d_j - x_{16} - x_{17} - x_{18} - x_{19}$. When the column is active ($d_j = 1$), b_j is set to $4 - x_{16} - x_{17} - x_{18} - x_{19}$ as the natural model, and when the column is inactive ($d_j = 0$), b_j becomes 0 since all x_i are inactive when the column is inactive.

***c*: Number of cancellations in AddRoundKey.** We introduce a variable c_i for each byte, and set $c_i = 1$ only if the difference cancellation occurs in AddRoundKey. In detail, we give additional constraints in the model of AddRoundKey in Eq. (2) so that c_i takes 1 only if non-zero x_i and non-zero stk_i result in zero differences on y_i, where $i = 16, 17, 18, 19$. Namely, when $x_i = 1$, $stk_i = 1$, and $y_i = 0$, we give a constraint that $c_i = 1$ is in the solution space while $c_i = 0$ is not. Similarly, when $(x_i, stk_i, y_i) \in \{(0,0,0), (0,1,1), (1,0,1), (1,1,1)\}$, we give a constraint that $c_i = 0$ is in the solution space while $c_i = 1$ is not. In the end, in addition to Eq. (2), we add the following 5 inequalities;

$$\begin{aligned} -x_i - stk_i + y_i + c_i &\geq -1, \\ x_i + stk_i + y_i - c_i &\geq 0, \\ x_i - stk_i - y_i - c_i &\geq -2, \\ -x_i + stk_i - y_i - c_i &\geq -2, \\ -x_i - stk_i - y_i - c_i &\geq -3. \end{aligned}$$

For example, $(x_i, stk_i, y_i) = (0,0,0)$ leads to $c_i = 0$ because the second inequality becomes $0 + 0 + 0 - c_i \geq 0$, and then $c_i = 1$ does not satisfy the system.

Recall that consuming degrees are calculated by $-(a - b - c)$. Let $\mathrm{TYPE2}_j$ be integer variables (not binary) to denote consuming degrees of type 2 for column j. Then

$$\mathrm{TYPE2}_j \geq -\big((y_0 + y_5 + y_{10} + y_{15}) - b_j - (\mathrm{cancel}_{16} + \mathrm{cancel}_{17} + \mathrm{cancel}_{18} + \mathrm{cancel}_{19})\big).$$

Finally, the constrains to avoid linear incompatibility can be modelled as "degrees of freedom is greater than or equal to consuming degrees", which is represented as

$$s \cdot \sum_{i=0}^{15} \mathrm{LANE}_i \geq \Big(r \cdot \sum_{i=0}^{15} \mathrm{LANE}_i - \sum_{i=0}^{16r-1} stk_i\Big) + \sum_{j=0}^{4r-1} \mathrm{TYPE2}_j.$$

3.3.3 Search Results and Discussion

The results of solving the models incorporating degrees are shown in Table 1 in the row of "linear incompatibility." The lower bounds become significantly tighter. It is interesting to notice that the difference in the bounds appears already for 5 rounds, which will give a significant impact on the evaluation of boomerang-type attacks. If we focus on the number of rounds to ensure 22 active S-boxes, the new bounds require only 8 (resp. 9) rounds, which improves from 10 (resp. 12) rounds proven by the designers in [JNPS16], and from 9 (resp. 11) rounds proven by the simple model in Section 3.2 for `Deoxys-BC-256` (resp. `Deoxys-BC-384`), respectively.

Note that dependencies among linear relations derived from Type 1 and Type 2 degrees of consumption are not considered in our tool. In general, the bounds are often established under some assumption e.g. Markov assumption, where the state is updated by independently chosen subkeys in every round. In our analysis, we assume that no path exists if the available degrees of freedom are smaller than the consumed degrees of freedom. This assumption is reasonable as demonstrated in Section 4.2 that when a

truncated differential path is given, the available degrees of freedom should be slightly larger than or equal to the degrees of consumption to suggest an exact differential path with a reasonable chance. In contrast, numbers with the simple model in Table 1 are lower bounds under exactly the same assumption as many previous works since there is no dependency among linear relations derived only from consumption degrees Type 1.

4 Boomerang and Rectangle Attacks against Deoxys-BC

Boomerang attacks and variants are typically useful for analysing schemes for which one can find good short differential paths and bad long ones. Indeed, the main strategy of these techniques is to combine short differential paths in order to attack a longer version of the cipher. As we can observe in Table 1, this seems to be exactly the case with Deoxys-BC (the number of active S-boxes remains rather small for a few rounds, but then quickly grows). Thus boomerang-like attacks are likely to be among the most powerful cryptanalytic techniques for this design.

4.1 Brief Introduction of the Attack Framework

Boomerang and Rectangle Attacks. Boomerang attack [Wag99] 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 $\alpha \to \beta$, and the second one to have a differential $\gamma \to \delta$, 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. Amplified boomerang attack works in a chosen-plaintext scenario and a right quartet is obtained with probability $p^2q^2 2^{-n}$ [KKS00]. Further, it was pointed out in [BDK01, BDK02] 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 $\hat{p} = \sqrt{\Sigma_i \mathrm{Pr}^2(\alpha \to \beta_i)}$ and $\hat{q} = \sqrt{\Sigma_j \mathrm{Pr}^2(\gamma_j \to \delta)}$. With this improvement, the attack was renamed as *rectangle attack*.

Boomerang and rectangle attacks under related-key setting were formulated in [BDK05]. Let ΔK and ∇K be the key differences for the first and second sub-cipher, respectively. The attack needs to access 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 $\hat{p}^2\hat{q}^2$. Related-key rectangle attacks can be similarly formulated.

Boomerang Switch. The boomerang switch was used to gain free rounds in the middle in the attacks against full AES-192 and AES-256 [BK09]. The idea was to optimise the transition between the sub-paths of E_0 and E_1 in order to minimise the overall complexity of the distinguisher. In [BK09], three types of switch were introduced. Two of them which are used in this paper are the *ladder switch* and the *S-box switch*.

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.

4.2 Search for Paths with High Probability

As mentioned before, the numbers of active S-boxes in Table 1 are lower bounds that do not ensure the existence of a path. For the first step to mount an attack, exact differential paths are searched given an active-byte pattern in truncated differential paths generated with methods in Section 3.3. In the literature, the MILP-based approach is widely used to search for exact differential paths. However, the 8-bit S-boxes used in `Deoxys-BC` are too heavy for MILP solvers. In [SGL+17], Sun et al. found 6-round related-key differential paths for AES-128 using a constraint programming (CP) solver. Thus, the CP-based approach seems to work for `Deoxys-BC`. However, the experiments show that the CP-based approach is applicable only when $r < 6$ (5) for `Deoxys-BC-384` (`Deoxys-BC-256`). In this subsection, an algorithm for searching exact related-key differential paths will be presented for `Deoxys-BC` and other block ciphers where the key schedule is linear. This algorithm applies even when the size of S-box is eight bits.

This algorithm exploits two observations on the generated active patterns. Firstly, the master tweakey difference is confined to a small set by linear equations derived from difference cancellations. Secondly, given an exact master tweakey difference, it is easy to find differential paths or verify there is no solution following the active pattern. With these two observations in mind, our algorithm is designed to proceed in two steps.

- Derive the space of the master tweakey difference. To do this, linear equations over the master tweakey difference are first extracted from the active pattern, and then the solution space is obtained by solving the system of linear equations.
- For each master tweakey difference in the solution space, search for differential paths following the active pattern.

Derive the Space of the Master Tweakey Difference. There are two types of linear equations over the master tweakey difference regarding the types of consumption degrees. The first comes from subtweakey difference cancellations. We follow the notation used in previous sections. Note that $\mathrm{LANE}_i = 1$ means the master tweakey differences $\Delta W_1[i], \Delta W_2[i]$ (and $\Delta W_3[i]$) for `Deoxys-BC-256` (`Deoxys-BC-384`) are active. If $\mathrm{LANE}_i = 1$ and after r rounds $stk_{16r+h^r_{inv}(i)} = 0$, then

$$\Delta W_1[i] \oplus LFSR_2^r(\Delta W_2[i]) = 0$$

for `Deoxys-BC-256`, and

$$\Delta W_1[i] \oplus LFSR_2^r(\Delta W_2[i]) \oplus LFSR_3^r(\Delta W_3[i]) = 0$$

for `Deoxys-BC-384`. For simplicity, we take `Deoxys-BC-256` as an example in the rest of this section.

The other type of linear equations comes from cancellations between subtweakey differences and state differences, i.e., consumption Type 2. An example of consumption degrees for 2-round Deoxys-BC is illustrated in Figure 3. As shown in the figure, two active bytes $\Delta X_r[4, 5]$ of the state at Round r are cancelled by the subtweakey difference. These two active state bytes are involved in the same MixColumns, and there is a linear relation between them, i.e. $c_1 \cdot \Delta X_r[4] \oplus c_2 \cdot \Delta X_r[5] = 0$ where c_1, c_2 are constants, due to the property of MixColumns that any four bytes of the input and output can be calculated from the remaining four bytes. Thus, there is also a linear relation between the two subtweakey difference bytes, namely

$$c_1 \cdot \Delta STK_r[4] \oplus c_2 \cdot \Delta STK_r[5] = 0.$$

Assume $\Delta STK_r[4] = \Delta W_1[i] \oplus LFSR_2^r(\Delta W_2[i])$, $\Delta STK_r[5] = \Delta W_1[j] \oplus LFSR_2^r(\Delta W_2[j])$ for some $i, j \in [0, 15]$. Then a linear equation over the master tweakey difference becomes

$$c_1 \cdot (\Delta W_1[i] \oplus LFSR_2^r(\Delta W_2[i])) \oplus c_2 \cdot (\Delta W_1[j] \oplus LFSR_2^r(\Delta W_2[j])) = 0.$$

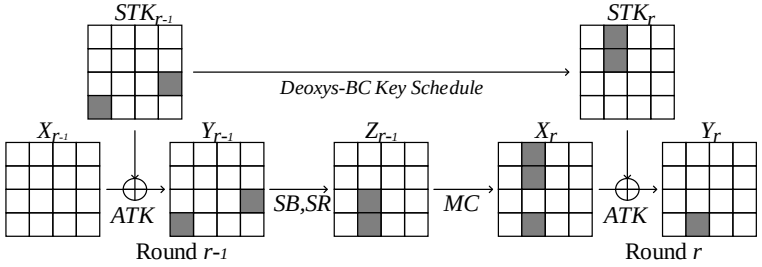

Figure 3: An example of consumption degrees Type 2.

Suppose the degrees of consumption of Type 1 and 2 are d_1 and d_2 respectively. Then there are $d_1 + d_2$ byte-wise linear equations in total. With all linear equations over the master tweakey difference extracted, the solution space can be calculated. Suppose all these linear equations are independent, then the size of solution space is $2^{(s\times\ell-d_1-d_2)\times 8}$. Note that s is the number of tweakey states and ℓ is the number of active lanes in the key schedule. Since the most available degrees of the master tweakey difference have been used to minimise the number of active S-boxes through MILP, the number of byte degree left, $s \times \ell - d_1 - d_2$ is small (usually 1 or even 0, and the actual bit degree of freedom may be slightly greater than $8 \times (s \times \ell - d_1 - d_2)$ due to bit-level dependencies), making the size of the resulting solution space small. Thus, traversing the solution space is practical.

Search for Differential Paths under Fixed Master Tweakey Difference. Given an exact master key difference, all subtweakey differences are determined accordingly since the key schedule is linear. With all subtweakey differences known, there are two types of S-boxes:

Type i the input and output differences are determined.

Type ii the input or output differences are not determined but some constraints are imposed by the subtweakey differences.

We again take Figure 3 as an example. At Round r, $\Delta X[4, 5]$ can be known from ΔSTK_r, and further $\Delta Z_{r-1}[6, 7]$ can be calculated through MixColumns. At Round $r - 1$, $\Delta Y[3, 14]$ are known from ΔSTK_{r-1}. Consequently, the input and output differences of the two active S-boxes at Round $r - 1$ are determined and thus these two S-boxes belong to Type i.

Active S-boxes at the first round and the last round belong to Type ii. There may exist Type ii S-boxes in the middle rounds which form into small groups. For this case, a local search for optimal differential path is needed for each group.

In short, the algorithm proceeds as follows.

1. Set $p_r = 0$
2. For each master tweakey difference from the solution space:
   (a) Compute the subtweakey differences.
   (b) Derive input/output differences of S-boxes from the subtweakey differences.

(c) For S-boxes of Type i, check whether the differentials are compatible or not. If not, go to Step 2.

(d) For S-boxes of Type ii, check whether compatible differentials among each group can be found or not.[3] If not, go to Step 2; otherwise, find the best local differentials for each group.

(e) Check if the probability p of the currently obtained path is greater than p_r. If yes, $p_r = p$ and save this path.

Given an active pattern, if there are t_s ($t_s > 0$) S-boxes of Type i, an exact differential path is not guaranteed, which can be seen from Step 2(c) of our algorithm. Assume the input and output differences of a Type ii S-box are random. Then the probability that the differential of this S-box is compatible is almost $\frac{1}{2}$. As a result, an exact differential path can be obtained with high chance only when $(s \times \ell - d_1 - d_2) \times 8 > t_s$. For the cases where $t_s > 0$, if the default constraint $s \times \ell - d_1 - d_2 \geq 0$ is imposed to the MILP model, then for generated active patterns it is unlikely to find an exact differential path. So setting aside 1/2-byte degrees in the MILP model is a way to increase the chance to find an exact differential path at a cost of more active S-boxes. Therefore, there is a tradeoff between $s \times \ell - d_1 - d_2$ and the number of Type ii S-boxes t_s.

Our algorithm for searching differential paths is implemented with SageMath. The main results are differential paths of `Deoxys-BC` that constitute the boomerang distinguishers, which are presented in the next section.

4.3 Boomerang Distinguisher of `Deoxys-BC`

In this section, we search for boomerang distinguishers of `Deoxys-BC` by exploiting the switching techniques. First, we incorporate the ladder switch into the MILP model, and generate truncated boomerang distinguishers. Then, given a truncated boomerang distinguisher, the upper path and the lower path are searched independently by applying the algorithm from Section 4.2.

Incorporate the Switching Techniques into the MILP Model. Suppose that the aim is to find a boomerang distinguisher over $R_1 + R_2$ rounds. First, we generate an MILP model for the first $R_1 + 1$ rounds and for the last $R_2 + 1$ rounds, respectively. Suppose binary variables $u_0, \cdots, u_{16 \cdot R_1 + 15}$ denote the activeness of S-boxes in the first $R_1 + 1$ rounds, and $l_0, \cdots, l_{16 \cdot R_2 + 15}$ denote the activeness of S-boxes in the last $R_2 + 1$ rounds. Then, we let the middle two rounds overlap by adding another 32 binary variable $y_0, \cdots, y_{31}$, and

$$\begin{aligned} u_{16\cdot(R_1-1)+i} - y_i &\geq 0, & u_{16\cdot R_1-1+i} - y_{16+i} &\geq 0, \\ l_i - y_i &\geq 0, & l_{16+i} - y_{16+i} &\geq 0, \\ -u_{16\cdot(R_1-1)+i} - l_i + y_i &\geq -1, & -u_{16\cdot R_1+i} - l_{16+i} + y_{16+i} &\geq -1, \end{aligned}$$

for $0 \leq i \leq 15$. In the inequalities, $y_i = 1$ if both of $u_{16\cdot(R_1-1)+i}$ and l_i are 1; otherwise, $y_i = 0$. Now the objective is to minimise

$$\sum_{i=0}^{16\cdot R_1-1} u_i + \sum_{j=0}^{31} y_j + \sum_{k=16}^{16\cdot R_2+15} l_k.$$

The boomerang distinguishers of `Deoxys-BC` obtained are listed in Table 4. Even though only a single path is considered for both E_0 and E_1, the notation $\hat{p}^2\hat{q}^2$ is still

[3] For the boomerang attacks explained later, we pick one of the best differential characteristics. As pointed out by a reviewer, multiple characteristics may contribute to increase the probability as demonstrated in the attack against MANTIS by Dobraunig et al. [DEKM16].

Table 4: Boomerang distinguishers

Deoxys-BC-256				Deoxys-BC-384			
R_1, R_2	#AS	pq	$\hat{p}^2\hat{q}^2$	R_1, R_2	#AS	pq	$\hat{p}^2\hat{q}^2$
4,4	6	2^{-36}	2^{-72}	5,5	4	2^{-24}	2^{-42}
5,4	9	2^{-61}	2^{-122}	6,5	9	2^{-60}	2^{-120}
5,5	16	2^{-106}	2^{-212}	6,6	15	2^{-98}	2^{-196}
6,5	20	2^{-136}	2^{-265}	7,6	20	2^{-134}	2^{-268}

used, because there may be improvements on the probabilities due to the S-box switch. Specifically, the 11-round distinguisher of Deoxys-BC-256 and the 10-round distinguisher of Deoxys-BC-384 utilise the S-box switch and save one active S-box.

4.4 Application to Deoxys-BC-384

We present a practical 10-round boomerang distinguisher with $4 \cdot 2^{42}$ data complexity and a 11-round boomerang distinguisher with $4 \cdot 2^{120}$ data complexity against Deoxys-BC-384. The data complexity is even lower if we attack a smaller number of rounds, e.g. $4 \cdot 2^6$ for 8-rounds and $4 \cdot 2^{18}$ for 9 rounds. We first focus on the 10-round distinguisher.

4.4.1 10-Round Distinguisher against Deoxys-BC-384

As summarised in Table 4, 10 rounds are divided into upper 5 rounds and lower 5 rounds. Table 1 shows that the number of active S-boxes for 5 rounds is at least 5. Hence the maximum probability of p and q is 2^{-30} for the straightforward evaluation, which requires $4 \cdot (pq)^{-2} = 2^{122}$ queries. This data complexity is already close to the full codebook. We found that, by using the ladder switch, the probabilities for 6 active S-boxes (out of 10) can be 1, which is the main reason that the complexity dropped into practical range.

The master tweakey difference is provided in Table 6 and the upper and lower 5-round paths are specified in Table 11 of Appendix. Hereafter the notation Δ and ∇ denote the difference for the upper path and lower path, respectively. In Table 11, the columns of X, K, Y and Z denote the initial state difference, subtweakey difference, state difference after AddRoundKey, and state difference after ShiftRows $\circ$ SubBytes, respectively.

Upper Path. According to Table 6, we use the following tweakey difference:

$$\Delta TK_0^1 = \begin{pmatrix} 00 & 00 & 90 & 00 \\ 00 & 00 & 00 & 1b \\ 8b & 00 & 00 & 00 \\ 00 & 90 & 00 & 00 \end{pmatrix}, \Delta TK_0^2 = \begin{pmatrix} 00 & 00 & 63 & 00 \\ 00 & 00 & 00 & 42 \\ 21 & 00 & 00 & 00 \\ 00 & 63 & 00 & 00 \end{pmatrix}, \Delta TK_0^3 = \begin{pmatrix} 00 & 00 & 7d & 00 \\ 00 & 00 & 00 & 49 \\ 34 & 00 & 00 & 00 \\ 00 & 7d & 00 & 00 \end{pmatrix}.$$

The values are hexadecimal numbers. These uniquely determine all subtweakey differences according to the key schedule. As shown in Table 11,

$$\Delta STK_0 = \begin{pmatrix} 00 & 00 & 8e & 00 \\ 00 & 00 & 00 & 10 \\ 9e & 00 & 00 & 00 \\ 00 & 8e & 00 & 00 \end{pmatrix}, \Delta STK_1 = \begin{pmatrix} 00 & 00 & 00 & bb \\ 00 & 00 & 00 & d2 \\ 00 & 00 & 00 & 69 \\ 00 & 00 & 00 & 69 \end{pmatrix}, \Delta STK_3 = 0, \Delta STK_4 = 0,$$

and so on. A notable feature is that the difference is chosen so that ΔSTK_3 and ΔSTK_4

are 0. The plaintext difference is

$$\Delta P = \begin{pmatrix} 00 & 00 & 8e & 00 \\ a3 & 00 & 00 & 10 \\ 9e & 00 & 00 & 00 \\ 00 & 8e & 00 & 00 \end{pmatrix},$$

which is chosen so that 4 byte differences are cancelled by the initial AddRoundKey with ΔSTK_0. Furthermore, the remaining 1-byte difference $a3$ will propagate to 69 with probability 2^{-6} through SubBytes, and cancel the 1-column difference in ΔSTK_1 in the second round. After this cancellation, zero-difference state continues until AddRoundKey in round 5, which activates 4 bytes in a diagonal position. However, as we explain below, the probability for those 4 active S-boxes can be 1 due to the ladder switch effect.

Lower Path. Similarly to the upper path, the master tweakey difference ∇K and differential propagation are defined in Table 6 and Table 11. To avoid redundancy, we omit re-defining the detailed data. The construction of the lower path is similar to the upper path but the propagation goes to the inverse direction according to the boomerang attack framework. The ciphertext difference ∇C is chosen to cancel the subtweakey difference ∇STK_9 by AddRoundKey in round 10, and then zero-difference state continues until AddRoundKey in round 7.

Switch in the Middle Two Rounds. In Table 11, the differential propagations are described in both of lower and upper paths for round 5 and round 6 to estimate the effect of ladder switch and S-box switch. Here, the goal is to divide the round function operation for rounds 5 and 6 into two independent parts, and assign one part to the upper path and the other part to the lower path in order to maximise the number of active S-boxes that are bypassed with probability 1.

Let $col(i)$ be the 4 bytes in column i and let $diag^{-1}(i)$ be 4 bytes that move to $col(i)$ by applying the map in ShiftRows. We denote by $\mathsf{OP}(col(i))$ and $\mathsf{OP}(diag^{-1}(i))$ the application of a OP operation only partially to $col(i)$ and $diag^{-1}(i)$. We also denote multiple columns or inverse diagonals by $col(i_1, i_2, \cdots)$ and $diag^{-1}(i_1, i_2, \cdots)$.

The upper path covers from plaintext to AddRoundKey in round 5. The lower path covers from ciphertext to round $\mathsf{ShiftRows}^{-1}$ in round 6. So, the remaining operations in the middle two rounds are SubBytes(SB), ShiftRows(SR), MixColumns(MC) in round 5 and AddRoundKey(AK), SubBytes(SB) in round 6. We divide those 5 operations as follows.

upper path				
$\mathsf{SB}(diag^{-1}(1,2,3))$	$\mathsf{SR}(diag^{-1}(1,2,3))$	$\mathsf{MC}(col(1,2,3))$	$\mathsf{AK}(col(1,2,3))$	$\mathsf{SB}(col(2,3))$
lower path				
$\mathsf{SB}(diag^{-1}(0))$	$\mathsf{SR}(diag^{-1}(0))$	$\mathsf{MC}(col(0))$	$\mathsf{AK}(col(0))$	$\mathsf{SB}(col(0,1))$

Then, the upper path does not include any active S-box during those 5 operations, and thus the probability of differential propagation is 1 owing to the ladder switch. The lower path contains one active S-box during the computation of $\mathsf{SubBytes}(\mathsf{col}(1))$ in round 6 (byte position 6), which lowers the probability of the lower path q by a factor of 2^{-6}. However, if analysed in detail, the input difference of this active S-box is fixed (9e) from the upper path, and the output difference of this active S-box is fixed (68) from the lower path. Then the S-box switch in [BK09] can be applied. Namely, to calculate the probability of forming a quartet (q^2), the probability for one of the pairs becomes 1, i.e. the probability stays q instead of q^2. The mechanism of this S-box switch is as follows. Suppose that a

paired input to the S-box i_1 and $i_2 = i_1 \oplus 9\mathrm{e}$ becomes $o_1 = \mathcal{S}(i_1)$ and $o_2 = \mathcal{S}(i_2)$, and $o_1 \oplus o_2 = 68$ by paying the cost of q. Then, values of the S-box output for the other pair are $o_3 = o_1 \oplus 68$ and $o_4 = o_2 \oplus 68$, which get back to o_2 and o_1. Hence, the corresponding i_3 and i_4 with probability 1 satisfy the difference 9e.

Complexity. All in all, there is one active S-box in round 1 of the upper path, which makes $p = 2^{-6}$ and there are two and one active S-boxes in round 10 and round 6 of the lower path, respectively, which makes $q = 2^{-18}$. When a quartet is constructed, we do not have to calculate the squared value for the S-box switch in round 6. Thus the probability of finding a right quartet is $(2^{-6})^2 \cdot 2^{-6} \cdot (2^{-12})^2 = 2^{-42}$. Considering that we make queries for 4 related-key oracles, the data complexity is $4 \cdot 2^{42} = 2^{44}$.

Experimental Verification. We experimentally verified the distinguisher for reduced-round variants. The first experiment is for 8 rounds. where round 1 and round 10 are dropped from Table 11. Then, the only probabilistic behaviour is the S-box switch in round 6, and the experiment clearly reflects the effect of the S-box switch (and the ladder switch). Moreover, we also need to take care of the observation by Murphy [Mur11] which pointed out that two independently chosen paths may not be connected.

The procedure in our experiment follows the related-key boomerang distinguisher framework introduced in Section 4.1. Namely, we define four related-key oracles, randomly generated many plaintext pairs (P_1, P_2), and check if the corresponding (P_3, P_4) returned by the boomerang structure satisfy the same plaintext difference. Among 2^{15} random pairs of (P_1, P_2), 546 pairs of (P_3, P_4) satisfy the correct plaintext difference. Therefore the probability to be a right quartet is $546/2^{15} \approx 2^{-5.91}$. This matches quite well the theoretically evaluated 2^{-6} of the probability of the S-box switch.

The second experiment is for 9 rounds, where only round 10 is dropped from Table 11. The theory expects that the probability for observing a right quartet lowers down by a factor of 2^{-12} compared to the 8-round attack, due to the additional active S-box in round 1. Among 2^{25} random pairs of (P_1, P_2), 133 pairs of (P_3, P_4) form a right quartet, thus the probability is $133/2^{25} \approx 2^{-17.94}$. This also matches well the theoretically evaluated $2^{-6} \cdot 2^{-12} = 2^{-18}$.

4.4.2 Rectangle Attack with Key Recovery against 13-Round Deoxys-BC-384

Similarly to the discussion above, as shown in Table 4, a distinguisher can be established against 11-round Deoxys-BC-384 with $4 \cdot 2^{120} = 2^{122}$ data complexity. In this section, we further extend the attack to 13 rounds by appending key recovery in a rectangle attack framework. The attack adds one round before and after the 11-round distinguisher as shown in Table 12.

We follow the notations and the generic key recovery algorithm for rectangle attacks from [LGS17], which provides formulae to estimate attack complexities. While Biham et al. original algorithm works under the single key setting, these formulae apply for related-key rectangle attacks as long as the key schedule is linear. In Round 0, let U_b be all the possible differences in the plaintext, and V_b be the space spanned by the values in U_b. Let $r_b = \log_2|V_b|$ and $t_b = \log_2|U_b|$. Let m_b be the STK_0 bits which are involved in the calculation of active S-boxes in Y_0. Similarly, we define U_f, V_f, r_f, t_f and m_f for Round 12. Note that in round 12, we can compute the difference in Z_{12} using the difference in the ciphertext and the known difference in STK_{13}. Hence, we can use Z_{12} to determine the values of U_f, V_f, r_f, t_f and m_f. The complexities of the key recovery attack are the following according to [LGS17].

- Data complexity: $D = 4M$ chosen plaintexts, where $M = \sqrt{g} \cdot 2^{n/2}/\hat{p}\hat{q}$ and g is the expected number of right quartets.

- Time complexity: $4M + 2 \cdot M^2 \cdot 2^{r_f - n} + 2 \cdot M^2 \cdot 2^{t_f - n} + M^2 \cdot 2^{2t_f + 2r_b - 2n}(1 + 2^{t_b - r_b}) + M^2 \cdot 2^{t_b + t_f - 2n + 1}(2^{m_b + t_f} + 2^{m_f + t_b})$ memory accesses.

- Memory complexity: $4M + 2^{t_b} + 2^{t_f} + 2^{m_b + m_f}$.

In the case of 13-round `Deoxys-BC-384` key recovery attack, $n = 128$ and set the number of right quartets $g = 4 \cdot \hat{p}\hat{q} = 2^{-60}$ which is from the 11-round boomerang distinguisher. Thus $M = \sqrt{4} \cdot 2^{64} \cdot 2^{60} = 2^{125}$. $r_b = \log_2(256^{12}) = 96$, $t_b = \log_2(127^{12}) = 83.8$, $m_b = 96$, $r_f = \log_2(256^6) = 48$, $t_f = \log_2(127^6) = 41.9$, $m_f = 48$. With these parameters, we can compute the complexities of the attack. The data complexity is $4M = 2^{127}$, the time complexity is $2^{269.8}$ memory access and the memory complexity is 2^{144}. Since the size of tweakey is 384 bits, the attack is better than the brute force attack as long as the key size is larger than 270 bits. If $t = 128$ as suggested by the designers, the attack covers 12 rounds: we utilise the 11-round distinguisher as shown in Table 12. Unlike the 13-round attack against `Deoxys-BC-384`, one round is added only before the distinguisher. Following the same key recovery algorithm, the data, time and memory complexities are 2^{127}, 2^{127} and 2^{125}, respectively.

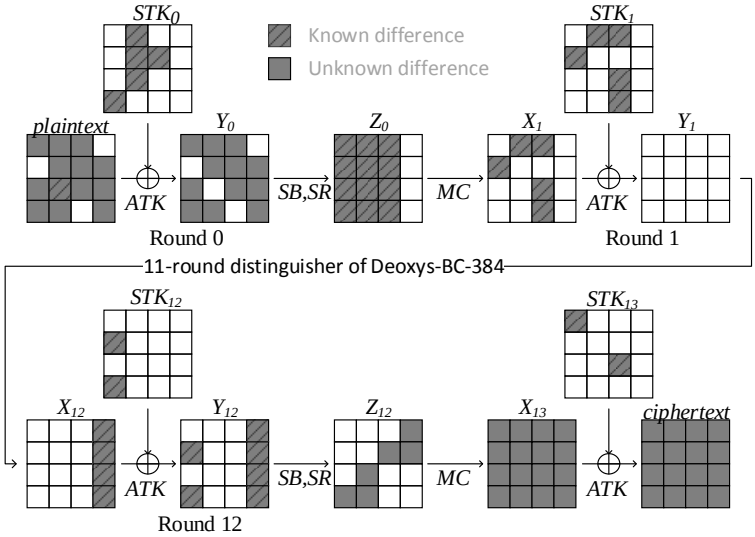

Figure 4: Key recovery attack against 13-round `Deoxys-BC-384` with data complexity below codebook.

4.5 Application to `Deoxys-BC-256`

An attack against `Deoxys-BC-256` can be mounted using similar techniques. We will therefore skip the details, and only present the main attack results for `Deoxys-BC-256`. Regarding distinguishers, as shown in Table 4, 8 rounds and 9 rounds can be distinguished with $4 \cdot 2^{72} = 2^{74}$ queries and $4 \cdot 2^{122} = 2^{124}$ queries, respectively.

For key recovery, by appending one round at the end of the 9-round distinguisher, 10-round `Deoxys-BC-256` can be attacked by rectangle attack with data complexity below codebook. The attack procedure is similar to the one for 13-round attack against `Deoxys-BC-384`. As a result, we can recover the key with data, time, memory complexities of $2^{127.58}$, 2^{204} and $2^{127.58}$, respectively. If $t = 128$, then 9 rounds of `Deoxys-BC-256` can be attacked: we utilise the 9-round distinguisher given in Table 8. The first 8 rounds are used as a distinguisher and the last round is considered as the appended round to the

distinguisher. Following the same key recovery algorithm in Section 4.4.2, the data, time and memory complexities are 2^{117}, 2^{118} and 2^{117}, respectively.

5 Beyond full-codebook for tweakable block ciphers

The complexity of cryptanalytic attacks against a block cipher is typically measured by the different resources required to successfully run the attack. The *time* complexity corresponds to the work effort required to mount the attack; the *data* complexity corresponds to the amount of data (e.g. plaintext-ciphertext pairs); finally the *memory* complexity denotes the amount of memory or storage required to run the attack. Regarding the data complexity, a reasonable assumption in the case of single-key attacks is that the amount of data that may potentially be available to an attacker is limited to the size of the message space, i.e. 2^n where n is the block length. An attack that requires 2^n plaintexts / ciphertexts is known as a *full-codebook* attack. While one could argue about the relevance or applicability of such an attack, a full-codebook attack may still play a role in the security analysis of a block cipher – for example, if the full domain encryption allows the recovery of the secret key, this may still indicate some structural weakness in the cipher construction.

However, when considering tweakable block ciphers, the data limit of 2^n message blocks may no longer need to apply. Recall that tweakable block ciphers take as input a plaintext (of length n) and a tweak (of length t), and so even in the single-key case, it is reasonable to assume that an attacker may have available an amount of data $D \gg 2^n$ to carry out an attack, as long as $D \leq 2^{n+t}$. In fact, these *beyond full-codebook* attacks have shown to be powerful and realistic against real-world tweakable block ciphers. For example, Bellare et al. [BHT16] describe an attack against the NIST standards for Format-Preserving Encryption (FPE) [Nat16] when they are used with small message spaces. They presented message-recovery, beyond full-codebook attacks that exploit the fact that the algorithms are Feistel-based tweakable block cipher constructions. For example, for 4-bit messages, the attacks fully recover the target message using between 2^{21} and 2^{25} ciphertexts. These require a large number of tweaks, but only three messages per tweak.

Ciphers adopting the `TWEAKEY` framework [JNP14], such as `Deoxys-BC`, offer further flexibility in setting the limit of data resources available for an attack. The construction allows one to add a tweak of (almost) any length to a key-alternating block cipher and/or to extend the key space of the block cipher to (almost) any size. A n-bit block cipher using the framework will take a k-bit key and a t-bit tweak, and a tweakey schedule will then take the $(k+t)$-bit tweakey to produce the n-bit round subtweakeys STK_i. Then for a fixed-size tweakey, the versatility of the `TWEAKEY` framework for setting the values of k and t provides attackers with a potentially optimal strategy to attack instances of `TWEAKEY` ciphers: select the key size k as large as possible – which results on a higher security claim – as long as the size of the tweak t is large enough to supply the required data to run the attack.

However, having sufficiently large plaintext/tweak space to satisfy the characteristic does not ensure the simple differential attack based on a differential characteristic with probability below 2^{-n}. This is because we will obtain too much wrong pairs that probabilistically satisfy the same input and output differences without following the characteristic. Here, with a careful analysis, we show that characteristics with probability below 2^{-n} can still be used to distinguish the cipher from the ideal one.

In general, our strategy is exploiting the small bias of the characteristic by iterating the bias many times. Assume that for an input difference Δ_i, the output difference Δ_j appears with probability 2^{-p}, where $p > n$. Also assume that for any output difference but Δ_j appears with probability 2^{-n}. Then, we consider asking 2^x pairs of (P, T) satisfying Δ_i where $x > p$, and count how many times each of output differences occur. The number of hits without satisfying the characteristic follows Poisson distribution with a

parameter $\lambda = 2^{x-n}$. When λ is big enough, Poisson distribution with parameter λ can be approximated to the normal distribution with average λ and variance λ. When a probabilistic variable X follows the normal distribution with average μ and variance σ^2, the probability that X is within $\mu \pm \sigma$ is 66.7%. (The probability that X is within $\mu \pm 3\sigma$ is 99.73%.) Hence, any output difference but Δ_j is observed about $2^{x-n} \pm \sigma$ times, while Δ_j is observed about $2^{x-n} \pm \sigma + 2^{x-p}$ times. σ is $2^{(x-n)/2}$. The condition to be a valid distinguisher is $2^{x-p} > 2^{(x-n)/2}$, which is $x > 2p - n$.

For example, let us consider the case with $n = 128$, $p = 140$ (and sufficiently large tweak size). The condition of the number of queries x is $x > 2 \cdot 140 - 128 = 152$. Indeed, when $x = 160$, random generations occur $2^{32} \pm 2^{16}$ times, while δ_o occurs $2^{32} \pm 2^{16} + 2^{20}$. Due to this difference, the cipher can be distinguished.

5.1 Rectangle Attacks on `Deoxys-BC`

Under classical settings where the available data is below full-codebook, it is pointed in [BDK02] that whenever the boomerang distinguisher succeeds then the key recovery attack also succeeds in boomerang and rectangle attacks. However, under the new setting where the available data can be more than 2^n, additional constraints should be satisfied to make the key recovery succeed.

Suppose the tweak size is t, the tweakey size is h, $\hat{p}\hat{q} = 2^{-w}$. In both boomerang attacks and rectangle attacks, there are two natural constraints: (1) the data complexity M under each related key should be less than 2^{n+t}; (2) the time complexity for processing data should be less than 2^{h-t}. Based on these two constraints, a rough bound for w is derived as follows. Note that the bound applies when the key schedule is linear.

In rectangle attacks, the data complexity under the related-key setting $D = 4M$. There are $M^2 = 2^x$ quartets. First, the data complexity under each related key should be less than the number of available data, i.e., $2^{\frac{x}{2}} < 2^{n+t}$. Second, the time complexity of analyzing 2^x quartets should be at least $2^x 2^{-n}$. This can be deduced from the extreme case where no round is appended after the distinguisher. In this extreme case, pairs of ciphertexts whose difference is not equal to the output difference of the distinguisher can be discarded immediately. The number of quartets remaining is $2^x 2^{-n}$ and the time complexity of the key recovery attack should be at least $2^x 2^{-n}$. Therefore, the second requirement is that $2^x 2^{-n} \leq 2^{h-t}$. Additionally, to distinguish the right key from wrong keys, $2^x 2^{-n-2w} > \sqrt{2^x 2^{-2n}}$ as demonstrated previously, where 2^{-n-2w} (2^{-2n}) is the probability of the right key (wrong keys) being suggested by a quartet. This implies that $x > 4w$. Consequently,

$$\begin{cases} 2w < n + t \\ 4w - n < h - t \\ t, w \geq 0 \end{cases}$$

should hold. If $h = 2n$, then $w \in [0, \frac{2}{3}n)$; if $h = 3n$, it follows that $w \in [0, \frac{5}{6}n)$.

This analysis provides an upper bound of w. Note that, in conventional rectangle attacks, w should be less than $\frac{n}{2}$. As can be seen, there exists a less tight bound of w in beyond code-book rectangle attacks.

We consider beyond code-book rectangle attacks of `Deoxys-BC` and check if it is possible to cover more rounds than conventional rectangle attacks. For `Deoxys-BC-384`, the 12-round distinguisher has $w = 98$. However, with the input and output differences being dense, a 14-round attack is not possible. Even though a 13-round attacks can be mounted, it requires higher complexities than the 13-round attacked described in Section 4.4.2. For `Deoxys-BC-256`, among the distinguisher we obtained (see Table 4), no one satisfies $64 < w < \frac{2}{3}n$). Although we do not find better rectangle attacks under the beyond code-book setting, this setting may be helpful under other cases where suitable distinguishers are found.

6 Impact on Deoxys Authenticated Encryption

In the previous sections, we studied the security of the Deoxys-BC tweakable block cipher, where an attacker can ask for encryption/decryption with any tweak value and even for related keys. However, the CAESAR submission Deoxys uses this primitive inside two operating modes, as described in Section 2. Here, we analyse to what extent an adversary is constrained by the fact that they only have access to the AE interface, and not the internal TBC directly.

First, a small restriction is due to the 4-bit encoding placed in the tweak input of the TBC in both Deoxys modes, in order to separate their various phases. The effect for an attacker is that 4 bits of the tweak input have to be fixed to the specific encoding value and thus cannot contain any difference. However so few bits are unlikely to represent a big challenge, and may well be overcome by choosing an appropriate differential paths that does not contain any difference on these 4 bits. We have verified that for the attacks given in the bottom subtable of Table 2, there is indeed no difference in these 4 bits.

Secondly, in Deoxys the maximum amount of data allowed for a given key is 2^{t-4}, which is equal to 2^{124} when using the recommended parameters given in [JNPS16]. Therefore, all attacks requiring more than this amount of resources per key should be discarded. We can note that if a rectangle attack requires a bit more data than this limit, it is possible to use less data by accepting a slightly lower success probability (or keeping the same success probability, but repeating the attack a few times and thus increasing the time complexity). Moreover, the claimed security of the Deoxys modes is obviously limited to the key size used. Thus, all attacks requiring more than 2^{128} time/data/memory for Deoxys-BC-256 or more than 2^{256} time/data/memory for Deoxys-BC-384 should be discarded as well.

Thirdly, a natural restriction when interacting with an AE scheme is that a null character is returned in case the tag is not valid during a decryption/verification (unless in the specific misuse setting where unverified plaintext is released [ABL+14]; however such a scenario is not claimed to be covered by the Deoxys designers). This of course prevents the classical boomerang attack to apply, as a decryption oracle to the internal TBC is required in this adaptive chosen plaintext ciphertext attack. However, this restriction is not problematic for the amplified boomerang variant, or for the rectangle attack, where only chosen plaintext is required. We also note a similar restriction due to the counter mode used for encryption in Deoxys-II: whatever the scenario, only the ciphering direction is computed and the attacker can never access the decryption primitive.

A final and potentially more problematic restriction is due to the nonce input of the AE mode. Indeed, in a nonce-respecting scenario, the attacker can only query a nonce once for a given key (a nonce can be queried several times only if a different key is used each time, which is allowed in a related-key setting). In the case of Deoxys-I, where the nonce is used as tweak input to the internal TBC together with a block counter, this restriction is not so problematic for the adversary. Indeed, the attacker can organise the queries in advance so that they can observe the encrypted data required to perform the amplified boomerang or rectangle attacks on the internal TBC (the attack just has to ensure that the proper difference is inserted in the tweak input). The case of Deoxys-II appears to be more problematic, as the tag value that is inserted in the tweak input of the internal TBC cannot be controlled, nor even predicted by the attacker (in contrast to Deoxys-I). This makes it very challenging for an adversary to organise the queries in advance to obtain the necessary data to run the amplified boomerang or rectangle attacks on the internal TBC. Moreover, the plaintext of the TBC is fixed to the nonce, which further restricts the attacker's abilities. Observing directly the tag to perform the attack will not work either as an extra TBC call is performed before outputting the tag value (thus two encryption layers have to be attacked). Even in the nonce-misuse scenario, the tag will be unpredictable to the attacker so this optimistic scenario does not seem to help the cryptanalysis.

In summary, from all the attacks against (reduced-round) Deoxys-BC described in

previous sections, the only ones that can be applied to the `Deoxys` AE modes are amplified boomerang or rectangle attacks against `Deoxys-I`, with a maximal data complexity of $\leq 2^{124}$ per key (or close to that threshold, with a tradeoff possible with the success probability and/or time complexity), and time complexity $\leq 2^{128}$ for `Deoxys-I-128-128`, and $\leq 2^{256}$ for `Deoxys-I-256-128`.

Acknowledgements

This research was initiated during the ASK 2016 workshop, held in Nagoya in September 2016. We thank all the group members at ASK 2016 for fruitful discussion. The authors would like to thank the anonymous referees for their helpful comments. The third author is supported by Temasek Labs (DSOCL16194). The fifth author is supported by the National Key Basic Research Program of China (2013CB834203) and the National Natural Science Foundation of China (Grants 61472417, 61472415, 61402469, and 61672516).

References

[ABL+14] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to Securely Release Unverified Plaintext in Authenticated Encryption. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 105–125. Springer, 2014.

[Ava17] Roberto Avanzi. The QARMA Block Cipher Family. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.

[BDK01] Eli Biham, Orr Dunkelman, and Nathan Keller. The Rectangle Attack - Rectangling the Serpent. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, volume 2045 of *Lecture Notes in Computer Science*, pages 340–357. Springer, 2001.

[BDK02] Eli Biham, Orr Dunkelman, and Nathan Keller. New Results on Boomerang and Rectangle Attacks. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, volume 2365 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2002.

[BDK05] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-Key Boomerang and Rectangle Attacks. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 507–525. Springer, 2005.

[BHT16] Mihir Bellare, Viet Tung Hoang, and Stefano Tessaro. Message-Recovery Attacks on Feistel-Based Format Preserving Encryption. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 444–455. ACM, 2016.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016. IACR Cryptology ePrint Archive 2016/625.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-Key Cryptanalysis of the Full AES-192 and AES-256. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[BN10] Alex Biryukov and Ivica Nikolic. Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer, 2010.

[BN11] Alex Biryukov and Ivica Nikolic. Search for Related-Key Differential Characteristics in DES-Like Ciphers. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 18–34. Springer, 2011.

[DEKM16] Christoph Dobraunig, Maria Eichlseder, Daniel Kales, and Florian Mendel. Practical key-recovery attack on MANTIS5. *IACR Trans. Symmetric Cryptol.*, 2016(2):248–260, 2016.

[DV17] F. Betül Durak and Serge Vaudenay. Breaking the FF3 format-preserving encryption standard over small domains. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 679–707. Springer, 2017.

[ELN+14] Sareh Emami, San Ling, Ivica Nikolic, Josef Pieprzyk, and Huaxiong Wang. The Resistance of PRESENT-80 against Related-Key Differential Attacks. *Cryptography and Communications*, 6(3):171–187, 2014.

[FJP13] Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 183–203. Springer, 2013.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International*

Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.

[JNPS16] Jérémy Jean, Ivica Nikolić, Thomas Peyrin, and Yannick Seurin. `Deoxys` v1.41. Submitted to CAESAR, October 2016.

[KKS00] John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 75–93. Springer, 2000.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song. Security Analysis of SKINNY under Related-Tweakey Settings. *to appear in IACR Trans. Symmetric Cryptol.*, 2017(3), 2017.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2002.

[Mur11] Sean Murphy. The Return of the Cryptographic Boomerang. *IEEE Trans. Information Theory*, 57(4):2517–2521, 2011.

[MV05] David A. McGrew and John Viega. The Galois/Counter Mode of Operation (GCM). Submission to NIST Modes of Operation Process, May 2005.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

[Nat01] National Institute of Standards and Technology. *Federal Information Processing Standards Publication 197: Advanced Encryption Standard (AES)*. NIST, November 2001.

[Nat07] National Institute of Standards and Technology. *NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. NIST, November 2007.

[Nat16] National Institute of Standards and Technology. *NIST Special Publication 800-38G: Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*. NIST, March 2016.

[SGL+17] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of AES, SKINNY, and Others with Constraint Programming. *IACR Trans. Symmetric Cryptol.*, 2017(1):281–306, 2017.

[Wag99] David Wagner. The Boomerang Attack. In Lars R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.

A A 13-Round Truncated Differential Path Considering Degrees

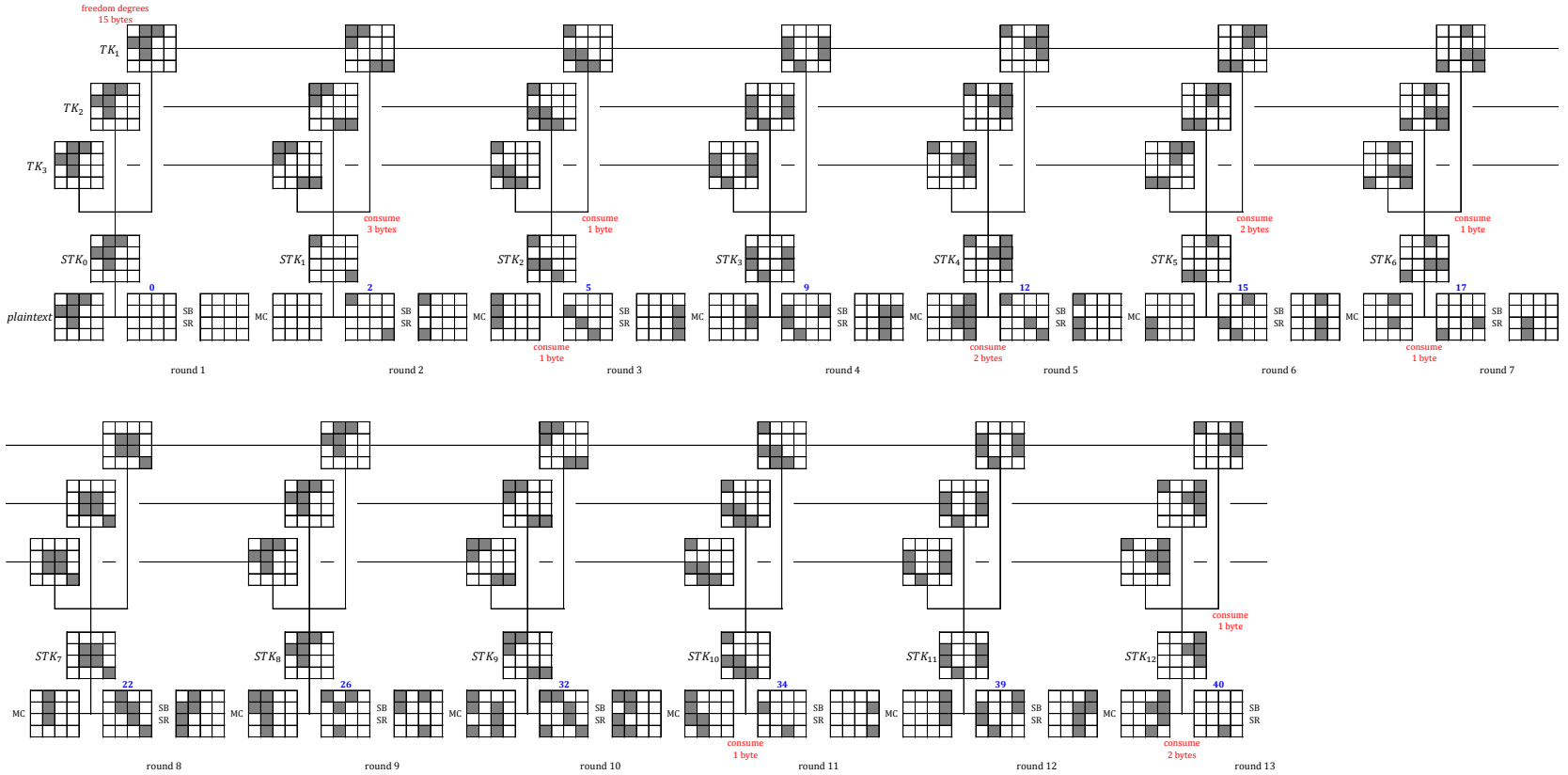

Figure 5: A 13-round truncated differential path considering degrees. Red comments explain the evaluation of degrees and blue numbers count the active S-boxes.

B Differential Paths and Boomerang Distinguishers

We give in this section the details of all boomerang distinguishers that are summarised in Table 4 of Section 4. The master tweakey differences are collected in Table 5 and Table 6. The differences are represented in hexadecimal and differences that are not crucial to the distinguishers are denoted with "**". At the meeting point of the upper path and the lower path, the grey colour is used to visualise the switching techniques.

Table 5: Master tweakey differences for distinguishers of `Deoxys-BC-256`

8 rounds	$\triangle K$	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 d1
	∇K	00 00 02 00 00 00 00 b3 00 00 00 00 00 00 00 00 00 00 a8 00 00 00 00 96 00 00 00 00 00 00 00 00
9 rounds	$\triangle K$	00 7f 00 00 00 ff 00 00 0b 00 f1 00 00 00 00 7c 00 cf 00 00 00 3f 00 00 70 00 5e 00 00 00 00 be
	∇K	00 00 00 00 00 a1 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 bf 00 a8 00 00 00 00 00 00 00 00
10 rounds	$\triangle K$	aa 71 c7 00 00 00 00 00 00 00 25 00 00 00 00 2a 38 98 00 00 00 00 00 00 00 12 00 00 00 00
	∇K	00 00 8d 00 00 00 00 00 61 00 00 00 00 00 00 08 00 00 83 00 00 00 00 00 e0 00 00 00 00 00 00 a8
11 rounds	$\triangle K$	00 3f 00 00 15 00 00 00 7f 00 00 00 00 00 00 07 00 cf 00 00 8a 00 00 00 9f 00 00 00 00 00 00 83
	∇K	00 00 00 00 00 00 00 00 52 61 fa 00 00 00 00 00 00 00 00 00 00 00 00 00 77 f0 66 00 00 00 00 00

Table 6: Master tweakey differences for distinguishers of `Deoxys-BC-384`

10 rounds	$\triangle K$	00 00 8b 00 00 00 00 90 90 00 00 00 00 1b 00 00 00 00 21 00 00 00 00 63 63 00 00 00 00 42 00 00 00 00 34 00 00 00 00 7d 7d 00 00 00 00 49 00 00
	∇K	00 00 00 00 00 00 00 6e 00 00 00 00 b1 00 00 00 00 00 00 00 00 00 00 42 00 00 00 00 f5 00 00 00 00 00 00 00 00 00 00 b3 00 00 00 00 d3 00 00 00
11 rounds	$\triangle K$	00 8b 00 00 c4 00 00 00 7a 00 c5 a6 00 00 00 00 00 ad 00 00 c4 00 00 00 73 00 21 d8 00 00 00 00 00 a3 00 00 9a 00 00 00 3b 00 0d 2e 00 00 00 00
	∇K	00 00 02 00 00 00 00 00 d7 00 00 00 00 00 00 00 00 00 99 00 00 00 00 00 bc 00 00 00 00 00 00 00 00 00 0c 00 00 00 00 00 f1 00 00 00 00 00 00 00
12 rounds	$\triangle K$	b8 00 7e 00 86 00 00 00 00 00 00 b8 00 00 d4 06 f5 00 c6 00 f5 00 00 00 00 00 00 f5 00 00 a6 3f a4 00 e6 00 48 00 00 00 00 00 00 a4 00 00 31 c0
	∇K	00 00 00 00 00 00 00 00 3d 00 00 00 00 00 00 58 00 00 00 00 00 00 00 00 c8 00 00 00 00 00 00 5b 00 00 00 00 00 00 00 00 8c 00 00 00 00 00 00 d0
13 rounds	$\triangle K$	dd 00 00 1b 00 00 00 00 00 00 00 f8 3f 00 00 e3 00 00 84 00 00 00 00 00 00 00 54 79 00 00 36 00 00 24 00 00 00 00 00 00 00 a0 99 00 00
	∇K	00 00 00 00 58 3d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2d e4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 a1 19 00 00 00 00 00 00 00 00 00 00

Table 7: 8-round distinguisher of Deoxys-BC-256

R	X	K	Y	Z	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	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	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 5f 00 00	00 00 00 00 00 00 00 00 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 05 00 00 00 00 00 00 00 00 00 42 00 00 00	00 ** ** ** ** 00 ** ** ** ** 00 ** ** ** ** 00	00 ** ** ** 00 ** ** ** 00 ** ** ** 00 ** ** **	1
5	00 7a 00 00 00 00 3e 00 00 00 82 ab 00 00 00 df	00 00 00 00 00 00 00 00 00 00 82 00 00 00 00 df	00 7a 00 00 00 00 3e 00 00 00 00 ab 00 00 00 00	00 b9 00 00 00 d1 00 00 00 01 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}

Table 8: 9-round distinguisher of `Deoxys-BC-256`

R	X	K	Y	Z	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 80 03 13 00 00 00 00 98 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 81 07 00 00 00 35 00 00 00 b4 00 1d 00 00	00 ** ** 07 00 ** ** 35 00 ** ** b4 00 ** ** 00	00 ** ** ** ** ** ** 00 ** ** 00 ** 00 00 ** **	1
5	** e4 00 ** ** ** 00 ** 00 ** ** 8f 5c 00 ** **	00 00 00 00 00 00 00 55 00 00 00 00 00 00 00 84	** e4 00 ** ** ** 00 00 00 ** ** 8f 5c 00 ** **	** e4 00 ** ** 00 00 ** ** 8f 00 ** ** 5c 00 **	1
6	b4 00 00 49 00 32 00 00 00 05 00 00 00 00 00 b5	00 00 00 49 00 00 00 00 00 05 00 00 00 00 00 00	b4 00 00 00 00 32 00 00 00 00 00 00 00 00 00 b5	ee 00 00 00 2f 00 00 00 00 00 00 00 b6 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 9: 10-round distinguisher of Deoxys-BC-256

R	X	K	Y	Z	p_r
1	80 00 00 00 49 00 00 00 5f 00 00 00 00 00 37 00	80 00 00 00 49 00 00 00 5f 00 00 00 00 00 37 00	00 00 00 00 00 00 00 00 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 f6 01 00 00 00 00 ff 00 00	00 00 00 00 00 00 00 f6 01 00 00 00 00 ff 00 00	00 00 00 00 00 00 15 00 00 00 15 00 00 00 15 00	2^{-21}
3	00 00 3f 00 00 00 00 00 00 00 00 00 00 00 2a 00	00 00 00 a4 00 00 00 9c 00 00 00 00 00 92 00 00	00 00 3f a4 00 00 00 9c 00 00 00 00 00 92 2a 00	00 00 16 a8 00 00 62 00 00 00 00 00 00 00 2c fc	2^{-33}
4	00 00 a6 b7 00 00 fe 54 00 00 00 b7 00 00 00 00	00 00 00 b7 00 00 fe 00 00 00 00 b7 00 00 00 00	00 00 a6 00 00 00 00 54 00 00 00 00 00 00 00 00	00 00 24 00 00 00 6c 00 00 00 00 00 00 00 00 00	2^{-14}
5	00 00 fc 00 00 00 fc 00 00 00 48 00 00 00 00 00	00 00 02 00 00 00 fc 00 00 00 48 00 00 00 00 00	00 00 fe 00 00 00 00 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 6a 00 00 d8 00 00 00 00 6e 00 00 00 00 fa	00 00 ** 00 00 d8 ** 00 00 00 ** 00 00 00 ** fa	00 00 ** 00 ** ** 00 00 ** 00 00 00 ** 00 00 **	1
5	** ** 00 ** ** ** ** 00 00 ** ** ** ** 88 ** **	67 00 00 00 00 00 00 00 00 00 b5 00 00 88 00 00	** ** 00 ** ** ** ** 00 00 ** ** ** ** 00 ** **	** ** 00 ** ** ** 00 ** ** ** 00 ** ** ** 00 **	1
6	00 ** 00 ** ** fc 00 ** ** ** 00 ** ** ** 00 00	00 00 00 00 00 fc 00 00 00 00 00 09 00 6d 00 00	00 ** 00 ** ** 00 00 ** ** ** 00 ** ** ** 00 00	00 73 00 f6 00 00 05 7b 00 f0 b8 7a 00 78 d3 00	1
7	00 6e 6e 00 00 00 0a 8e 00 00 00 79 00 95 00 00	00 6e 00 00 00 00 0a 00 00 00 00 79 00 00 00 00	00 00 6e 00 00 00 00 8e 00 00 00 00 00 95 00 00	00 00 20 00 00 00 3d 00 00 00 00 00 00 00 0b 00	2^{-20}
8	00 00 0c 00 00 00 51 00 00 00 00 00 00 00 4b 00	00 00 0c 00 00 00 51 00 00 00 00 00 00 00 4b 00	00 00 00 00 00 00 00 00 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 96 00 18 00 00 00 00 00 a2	00 00 00 00 00 00 00 96 00 18 00 00 00 00 00 a2	00 00 00 00 00 00 f3 00 00 00 00 ce 59 00 00 00	2^{-18}

Table 10: 11-round distinguisher of `Deoxys-BC-256`. The S-box switch is used in Round 7 (lower) for the S-box at position (3,3).

R	X	K	Y	Z	p_r
1	00 9f e0 00 f0 00 00 00 00 00 00 00 00 00 00 84	00 9f e0 00 f0 00 00 00 00 00 00 00 00 00 00 84	00 00 00 00 00 00 00 00 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	a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40	a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40	12 00 00 00 00 00 00 00 00 00 00 00 1b 00 00 00	2^{-14}
3	3f 00 00 00 09 00 00 00 3f 00 00 00 00 00 00 00	00 00 00 00 09 00 00 00 3f 00 00 00 00 00 00 00	3f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	2^{-7}
4	1b 00 00 00 80 00 00 00 80 00 00 00 9b 00 00 00	1b 00 00 00 80 00 00 40 00 00 00 40 00 00 00 00	00 00 00 00 00 00 00 40 80 00 00 40 9b 00 00 00	00 00 00 00 00 00 45 00 00 15 cf 00 00 15 00 00	2^{-28}
5	00 00 00 00 00 2a c0 00 00 15 c0 00 00 3f 8a 00	81 00 00 bf 00 00 c0 00 00 00 00 00 00 3f 00 00	81 00 00 bf 00 2a 00 00 00 15 c0 00 00 00 8a 00	3f 00 00 a0 3f 00 00 00 3f 00 00 3b 00 00 00 60	2^{-42}
6	00 00 00 00 00 00 00 8d 7e 00 00 76 41 00 00 00	00 00 c1 00 00 00 00 00 00 00 00 76 41 83 00 00	00 00 c1 00 00 00 00 8d 7e 00 00 00 00 83 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 e4 00 00 00 bd 87 00 00 00 c3	00 00 ** 00 00 00 ** 00 00 00 ** 87 00 00 ** c3	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	67 00 00 00 00 00 00 25 00 00 00 00 00 a7 00 00	** 00 ** ** ** ** 00 ** ** ** ** 00 00 ** ** **	** 00 ** ** ** 00 ** ** ** 00 ** ** ** 00 ** **	1
7	7c 00 00 e8 34 d6 00 54 dd 8e 00 b8 00 b2 00 c3	00 00 00 45 00 00 00 00 00 00 00 b8 00 6d 00 00	7c 00 00 ad 34 d6 00 54 dd 8e 00 00 00 df 00 c3	73 00 00 f6 95 00 43 7b 00 00 a2 7a 42 00 fd 00	2^{-7}
8	00 00 9a 00 00 00 86 8e 20 00 00 79 84 00 00 00	00 00 00 00 00 00 86 00 00 00 00 79 84 00 00 00	00 00 9a 00 00 00 00 8e 20 00 00 00 00 00 00 00	00 00 db 00 00 00 10 00 00 00 66 00 00 00 00 00	2^{-20}
9	00 00 fb 00 00 00 51 00 00 00 07 00 00 00 00 00	00 00 fb 00 00 00 51 00 00 00 07 00 00 00 00 00	00 00 00 00 00 00 00 00 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 0e 00 00 00 00 00 00 00 f6 00 00 00 00 00 a2	00 0e 00 00 00 00 00 00 00 f6 00 00 00 00 00 a2	00 c8 00 00 00 00 00 00 00 00 00 21 59 00 00 00	2^{-18}

Table 11: 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).

R	X	K	Y	Z	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	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	1
6	** 00 00 00 ** 00 00 00 ** 00 00 00 ** 00 00 00	00 10 00 00 00 9e 00 00 00 8e 00 00 00 8e 00 00	** 10 00 00 ** 9e 00 00 ** 8e 00 00 ** 8e 00 00	** ** 00 00 ** 00 00 ** 00 00 ** ** 00 ** ** 00	1
5	00 ** ** ** ** 00 ** ** ** ** 00 ** ** ** ** **	00 ee 00 00 00 00 00 00 00 00 00 00 00 00 00 11	00 ** ** ** ** 00 ** ** ** ** 00 ** ** ** ** 00	00 ** ** ** 00 ** ** ** 00 ** ** ** 00 ** ** **	1
6	00 00 00 00 00 9e 00 00 00 0a ab 00 00 00 93 7a	00 00 00 00 00 00 00 00 00 0a 00 00 00 00 93 00	00 00 00 00 00 9e 00 00 00 00 ab 00 00 00 00 7a	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 12: 11-round distinguisher of `Deoxys-BC-384`

R	X	K	Y	Z	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 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	1
7	00 ** ** 00 00 ** ** 00 00 ** ** 00 00 ** ** 00	00 00 00 08 00 50 00 00 00 00 13 09 00 00 00 1b	00 ** ** ** 00 ** ** 00 00 ** ** ** 00 ** ** **	00 ** ** ** ** ** 00 00 ** ** 00 ** ** 00 ** **	1
6	** cb 00 ** ** ** ff 00 00 ** ** 1a 00 ** ** **	00 00 00 00 00 ** 00 00 00 00 00 00 00 ** 00 00	** cb 00 ** ** ** ff 00 00 ** ** 1a 00 00 ** **	** cb 00 ** ** ff 00 ** ** 1a 00 ** ** 00 00 **	1
7	00 8d 00 00 00 00 00 e6 14 00 00 af 00 a3 00 00	00 8d 00 00 00 00 00 00 00 00 00 af 00 00 00 00	00 00 00 00 00 00 00 e6 14 00 00 00 00 a3 00 00	00 00 00 00 00 00 ed 00 00 00 99 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}

Table 13: 12-round distinguisher of Deoxys-BC-384

R	X	K	Y	Z	p_r
1	e9 3b 00 00 00 00 00 00 5e 00 00 43 00 00 e9 f9	e9 3b 00 00 00 00 00 00 5e 00 00 43 00 00 e9 f9	00 00 00 00 00 00 00 00 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 99 00 00 00 00 c8 00	00 00 00 00 00 00 00 00 00 99 00 00 00 00 c8 00	00 00 00 00 00 00 00 00 00 00 00 e1 00 00 00 fd	2^{-14}
3	00 00 00 1c 00 00 00 c5 00 00 00 c5 00 00 00 00	00 00 00 1c 09 00 00 c5 00 00 00 c5 00 00 00 00	00 00 00 00 09 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 43 00 00 00 00 00 00 00 00	2^{-7}
4	00 00 00 c5 00 00 00 86 00 00 00 43 00 00 00 43	00 00 00 c5 00 00 c5 86 00 00 00 00 74 00 00 43	00 00 00 00 00 00 c5 00 00 00 00 43 74 00 00 00	00 00 00 00 00 07 00 00 00 f8 00 00 00 f1 00 00	2^{-20}
5	00 00 00 00 00 ec 00 00 00 e4 00 00 00 06 00 00	00 00 00 00 00 00 00 00 00 e4 0d 00 00 06 00 00	00 00 00 00 00 ec 00 00 00 00 0d 00 00 00 00 00	00 00 00 00 79 00 00 00 b1 00 00 00 00 00 00 00	2^{-14}
6	3a 00 00 00 3a 00 00 00 00 00 00 00 c8 00 00 00	00 00 00 00 3a aa 00 00 00 00 e9 00 c8 00 00 e9	3a 00 00 00 00 aa 00 00 00 00 e9 00 00 00 00 e9	** 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	cf 4e 00 00 00 8a 09 00 00 8a 3b 00 00 00 00 00	** 4e 00 00 ** 8a 09 00 ** 8a 3b 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 40 00 94 00 00	00 ** ** ** ** 00 ** ** ** ** 00 ** ** ** ** 00	00 ** ** ** 00 ** ** ** 00 ** ** ** 00 ** ** **	1
7	00 00 ** 00 00 00 2a ** 00 00 00 5d 00 ** 00 00	00 00 00 00 00 00 2a 00 00 00 00 5d 00 00 00 00	00 00 ** 00 00 00 00 ** 00 00 00 00 00 ** 00 00	00 00 f5 00 00 00 fd 00 00 00 00 00 00 00 f1 00	1
8	00 00 1c 00 00 00 e5 00 00 00 00 00 00 00 00 00	00 00 1c 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 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 00 1c 00 00 00 00 e5 00 00 00 00 00 00	00 00 00 00 1c 00 00 00 00 e5 00 00 00 00 00 00	00 00 00 00 00 00 00 ff 00 00 00 aa 00 00 00 00	2^{-13}
12	00 00 00 b0 00 00 00 00 00 00 00 b0 00 00 00 55	2a 00 00 00 5d 00 00 00 00 00 00 00 00 00 00 00	2a 00 00 b0 5d 00 00 00 00 00 00 b0 00 00 00 55	86 00 00 84 00 00 00 2f 00 84 00 00 9f 00 00 00	2^{-30}

Table 14: 13-round distinguisher of `Deoxys-BC-384`

R	X	K	Y	Z	p_r
1	08 00 00 0c 00 00 00 df 00 00 00 00 bb 00 00 00	08 00 00 0c 00 00 00 df 00 00 00 00 bb 00 00 00	00 00 00 00 00 00 00 00 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 00 00 00 00 dd 3f 00 00 00	00 00 00 00 00 00 00 00 00 00 00 dd 3f 00 00 00	00 00 00 00 00 00 00 00 00 d6 00 00 00 6d 00 00	2^{-14}
4	00 bb 00 00 00 0c 00 00 00 00 00 00 00 0c 00 00	00 bb 00 00 00 0c 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 0c 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 00	2^{-7}
5	00 00 08 00 00 00 08 00 00 00 18 00 00 00 10 00	00 12 08 00 00 df 00 00 00 00 00 00 00 00 10 00	00 12 00 00 00 df 08 00 00 00 18 00 00 00 00 00	00 da 00 00 18 75 00 00 18 00 00 00 00 00 00 00	2^{-28}
6	30 30 00 00 18 30 00 00 28 af 00 00 00 00 00 00	00 30 00 00 00 00 00 00 28 00 00 00 00 00 2d c1	30 00 00 00 18 30 00 00 00 af 00 00 00 00 2d c1	b8 00 00 00 b8 00 00 77 00 00 00 11 b8 00 00 33	2^{-42}
7	00 00 00 bb 6b 00 00 ee d3 00 00 00 00 00 00 00	00 00 00 00 00 00 00 ee d3 3e 00 00 00 00 46 00	00 00 00 bb 6b 00 00 00 00 3e 00 00 00 00 46 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 b8 76 00 00 ef ff 00 00 00 00 00 00 00	00 00 00 ** 76 00 00 ** ff 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 40 00 94 00 00 00	** ** ** 00 00 ** ** ** ** 00 ** ** ** ** 00 **	** ** ** 00 ** ** ** 00 ** ** ** 00 ** ** ** 00	1
8	00 ** 00 00 00 2a ** 00 00 00 5d 00 ** 00 00 00	00 00 00 00 00 2a 00 00 00 00 5d 00 00 00 00 00	00 ** 00 00 00 00 ** 00 00 00 00 00 ** 00 00 00	00 f5 00 00 00 fd 00 00 00 00 00 00 00 f1 00 00	1
9	00 1c 00 00 00 e5 00 00 00 00 00 00 00 00 00 00	00 1c 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 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 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00 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
12	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 1c e5 00 00 00 00 00 00 00	00 00 00 00 00 00 00 1c e5 00 00 00 00 00 00 00	00 00 00 00 00 00 ff 00 00 00 aa 00 00 00 00 00	2^{-13}
13	00 00 b0 00 00 00 00 00 00 00 b0 00 00 00 55 00	00 00 00 2a 00 00 00 5d 00 00 00 00 00 00 00 00	00 00 b0 2a 00 00 00 5d 00 00 b0 00 00 00 55 00	00 00 84 86 00 00 2f 00 84 00 00 00 00 00 00 9f	2^{-30}