Boomerang Connectivity Table: A New Cryptanalysis Tool

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

Abstract. A boomerang attack is a cryptanalysis framework that regards a block cipher E as the composition of two sub-ciphers $E_1 \circ E_0$ and builds a particular characteristic for E with probability p^2q^2 by combining differential characteristics for E_0 and E_1 with probability p and q, respectively. Crucially the validity of this figure is under the assumption that the characteristics for E_0 and E_1 can be chosen independently. Indeed, Murphy has shown that independently chosen characteristics may turn out to be incompatible. On the other hand, several researchers observed that the probability can be improved to p or q around the boundary between E_0 and E_1 by considering a positive dependency of the two characteristics, e.g. the ladder switch and S-box switch by Biryukov and Khovratovich. This phenomenon was later formalised by Dunkelman et al. as a sandwich attack that regards E as $E_1 \circ E_m \circ E_0$, where E_m satisfies some differential propagation among four texts with probability r, and the entire probability is p^2q^2r. In this paper, we revisit the issue of dependency of two characteristics in E_m, and propose a new tool called *Boomerang Connectivity Table (BCT)*, which evaluates r in a systematic and easy-to-understand way when E_m is composed of a single S-box layer. With the BCT, previous observations on the S-box including the incompatibility, the ladder switch and the S-box switch are represented in a unified manner. Moreover, the BCT can detect a new switching effect, which shows that the probability around the boundary may be even higher than p or q. To illustrate the power of the BCT-based analysis, we improve boomerang attacks against `Deoxys-BC`, and disclose the mechanism behind an unsolved probability amplification for generating a quartet in `SKINNY`. Lastly, we discuss the issue of searching for S-boxes having good BCT and extending the analysis to modular addition.

keywords: boomerang attack, S-box, differential distribution table, incompatibility, ladder switch, S-box switch, `Deoxys`, `SKINNY`.

1 Introduction

Differential cryptanalysis, proposed by Biham and Shamir in the early 1990s [BS93], remains one of the most fundamental cryptanalytic approaches for assessing the security of block ciphers. For iterated ciphers based on predefined substitution tables (S-box), resistance against differential cryptanalysis is highly dependent on the non-linearity features of the S-box.

For an n-bit S-box $S : \{0,1\}^n \mapsto \{0,1\}^n$, the properties for differential propagations of S are typically represented in the $2^n \times 2^n$ table $\mathcal{T}$, called the *Difference Distribution Table (DDT)*. For any pair (Δ_i, Δ_o), the value

$$\#\{x \in \{0,1\}^n | S(x) \oplus S(x \oplus \Delta_i) = \Delta_o\}$$

is stored in the corresponding entry $\mathcal{T}(\Delta_i, \Delta_o)$ of the DDT, representing that the input difference Δ_i propagates to the output difference Δ_o with probability

$$\frac{\mathcal{T}(\Delta_i, \Delta_o)}{2^n}. \tag{1}$$

The maximum entry in the table $\mathcal{T}$ (outside the first row and column) is called the *differential uniformity* of S.

As an example, the DDT for the 4-bit S-box used in PRESENT [BKL+07] and LED [GPPR11] is shown in Table 1. We can observe that the differential uniformity of the S-box is 4.

Table 1. Difference Distribution Table (DDT) of the PRESENT S-box

Δ_i \ Δ_o	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
a	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
b	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
c	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
d	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
e	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
f	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

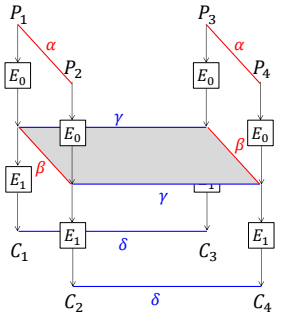

Fig. 1. Boomerang attack

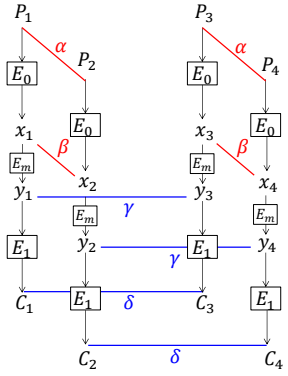

Fig. 2. Sandwich attack

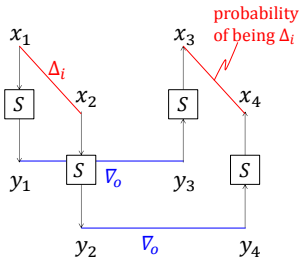

Fig. 3. Computation of r when E_m is an S-box layer

While Equation (1) represents the differential propagation property for a single S-box, in order to derive the differential properties of the entire cipher, a trail of high-probability differentials is searched through the cipher iteration, by assuming that the S-boxes and other operations applied in different rounds behave independently.

In many cases, it may not be possible to find a high-probability trail for the entire cipher. In such cases, the *Boomerang attack* framework, proposed by Wagner [Wag99], may be applied to exploit the differential properties of different segments of the cipher. In a boomerang attack, the target cipher E is regarded as a composition of two sub-ciphers E_0 and E_1, i.e. $E = E_1 \circ E_0$. Then suppose that the input difference α is propagated to the difference β by E_0 with probability p, while the difference γ is propagated to δ by E_1 with probability q. The boomerang attack exploits the expected probability of the following differential (depicted in Figure 1):

$$\Pr\left[E^{-1}\left(E(x) \oplus \delta\right) \oplus E^{-1}\left(E(x \oplus \alpha) \oplus \delta\right) = \alpha\right] = p^2 q^2. \tag{2}$$

Then, on making around $(pq)^{-2}$ adaptive chosen plaintext/ciphertext queries, E can be distinguished from an ideal cipher.

Variants of the boomerang attack were later proposed: the amplified boomerang attack (also called 'the rectangle attack') works in a chosen-plaintext scenario and a right quartet is expected to be obtained with probability $p^2 q^2 2^{-n}$ [KKS00]. Further, it was pointed out in [BDK01,BDK02] that any values of β and γ are allowed as long as $\beta \neq \gamma$. As a result, the probability of the right quartet increases to $2^{-n}\hat{p}^2\hat{q}^2$, where $\hat{p} = \sqrt{\Sigma_i \Pr^2(\alpha \to \beta_i)}$ and $\hat{q} = \sqrt{\Sigma_j \Pr^2(\gamma_j \to \delta)}$.

In boomerang-style attacks, the most important part of the attack is selecting suitable differential characteristics for E_0 and E_1. Initially, the standard assumption used in boomerang-style attacks was that two characteristics independently chosen for E_0 and E_1 could be used; as a result the typical attacker's strategy

was to optimise the best characteristics independently for the sub-ciphers E_0 and E_1. However, Murphy [Mur11] pointed out that, for S-box based ciphers, two independently chosen characteristics can be *incompatible*, thus the probability of generating a right quartet can be zero. He also showed that the dependency between two characteristics could give advantages for the attacker, giving an example that the probability of generating a quartet was pq instead of p^2q^2 when E_0 and E_1 are composed of a single S-box. The same phenomenon was observed by Biryukov et al. as the *middle round S-box trick* [BCD03].

Another improvement, proposed by Biryukov and Khovratovich [BK09], was named the *boomerang switch*. Suppose that the cipher state is composed of several words (typically 8 bits or 4 bits) and the round function applies S-boxes to each word in parallel. The main observation in [BK09] is that the boundary of E_0 and E_1 does not need to be defined on a state. Instead, a state can be further divided into words, and some words can be in E_0 and others can be in E_1. Suppose that half of the state is active only in E_0 and the other half is active only in E_1. Then, by regarding the former as a part of E_1 and the latter as a part of E_0, the probability on all the active S-boxes becomes 1. This technique is called *ladder switch*. Another switching technique in [BK09], is the *S-box switch*. When both the characteristics for E_0 and E_1 activate the same S-box with an identical input difference and an identical output difference, the probability of this S-box to generate a quartet is p instead of p^2.

Those observations were later formalised by Dunkelman et al. as *the sandwich attack* [DKS10,DKS14] depicted in Figure 2, that regards E as $E_1 \circ E_m \circ E_0$, where E_m is a relatively short operation satisfying some differential propagation among four texts with probability r, and the entire probability is p^2q^2r. Let (x_1, x_2, x_3, x_4) and (y_1, y_2, y_3, y_4) be input and output quartet values for E_m, where $y_i = E_m(x_i)$. The differential characteristics for E_0 specify the input differences α to E_m, namely $x_1 \oplus x_2 = x_3 \oplus x_4 = \alpha$, and E_1 specifies the output differences β to each S-box, namely $y_1 \oplus y_3 = y_2 \oplus y_4 = \beta$. Dunkelman et al. define r as follows [DKS10, Equation (4)].

$$r = \Pr\Big[(x_3 \oplus x_4) = \beta | (x_1 \oplus x_2 = \beta) \wedge (y_1 \oplus y_3 = \gamma) \wedge (y_2 \oplus y_4 = \gamma)\Big] \qquad (3)$$

Boomerang-style attacks have become an ever more popular cryptanalytic method for assessing the security of block ciphers. Yet, considering the research results above, we note the following questions that arise in their context:

- the probability r of the middle part E_m is for a quartet. Then, how can we evaluate r in an efficient and systematic way? The only known approach is to run experiments as in [DKS10,DKS14,BDK01,KHP+12].
- are there other switching techniques that can be used to improve boomerang-style attacks? In particular, can we find switching techniques that connect two characteristics with even higher probability than the S-box switch and Murphy's examples?

Answer to these questions would be of course of great interest to researchers working on block cipher cryptanalysis, but also significant to provide a deeper

understanding of the subtleties of boomerang-style attacks. Besides, it also contributes to block ciphers designers by taking into account this property as a criterion to choose a good S-box.

Our Contributions. This paper positively answers the above questions by proposing a new tool for evaluating the probability that boomerang-style quartets are generated. While we focus mainly on explaining the effects against ciphers employing S-boxes, we also present the extension to analyse ciphers based on modular addition.

Suppose that the middle layer E_m of the sandwich attack is composed of a single S-box layer. Then, for a given pair of (Δ_i, ∇_o), the probability that a right quartet is generated in each S-box in the middle S-box layer is given by:

$$\frac{\#\{x \in \{0,1\}^n | S^{-1}(S(x) \oplus \nabla_o) \oplus S^{-1}(S(x \oplus \Delta_i) \oplus \nabla_o) = \Delta_i\}}{2^n}, \tag{4}$$

where $S : \{0,1\}^n \mapsto \{0,1\}^n$ is an n-bit to n-bit S-box and S^{-1} is its inverse. What Equation (4) evaluates is illustrated in Figure 3, which is exactly r in Equation(3) when E_m is a single S-box layer. Note that the differences for E_0 and E_1 are defined between different paired values, thus we use Δ and ∇ to denote the differences of E_0 and E_1, respectively. We also note that in the figures we mainly use red and blue colours to describe Δ and ∇, respectively. The denominator is 2^n instead of 2^{2n}, which shows the implication of the sandwich attack that the probability r of generating a right quartet in E_m is at least 2^{-n} (if not 0).

Similar to the DDT, we can of course evaluate Equation (4) for all pairs of (Δ_i, ∇_o), storing the results (in fact the numerator) in a table. We call this table the *Boomerang Connectivity Table (BCT).* The BCT for the PRESENT S-box is shown in Table 2.

The BCT represents the observations by [Mur11] and [BK09] in a unified manner.

Incompatibility. (Δ_i, ∇_o) is incompatible when the corresponding entry in the BCT is 0.

Ladder switch. It corresponds to the first row and the first column of the BCT, in which either one of the input or output difference is zero, while the other is non-zero. As suggested by Table 2, equation (4) gives probability 1.

S-box switch. It corresponds to the claim that a DDT entry with non-zero value v would imply that the corresponding BCT entry is v. While this is correct in some cases, as it can be observed from the two tables, we show that the value of the BCT can in fact be larger than v owing to the new switching effect. However, at least the effect of S-box switch is always guaranteed.

Study of the BCT can also present more advantages to the attacker compared to the previously known switching techniques. With respect to versatility, the BCT shows that the switching effect can be applied even when Δ_i cannot be propagated to Δ_o in the DDT. With respect to strength, the maximum probability in the BCT is usually higher than that of the DDT. For example, the

Table 2. Boomerang Connectivity Table (BCT) of the PRESENT S-box

Δ_i \ ∇_o	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
1	16	0	4	4	0	16	4	4	4	4	0	0	4	4	0	0
2	16	0	0	6	0	4	6	0	0	0	2	0	2	2	2	0
3	16	2	0	6	2	4	4	2	0	0	2	2	0	0	0	0
4	16	0	0	0	0	4	2	2	0	6	2	0	6	0	2	0
5	16	2	0	0	2	4	0	0	0	6	2	2	4	2	0	0
6	16	4	2	0	4	0	2	0	2	0	0	4	2	0	4	8
7	16	4	2	0	4	0	2	0	2	0	0	4	2	0	4	8
8	16	4	0	2	4	0	0	2	0	2	0	4	0	2	4	8
9	16	4	2	0	4	0	2	0	2	0	0	4	2	0	4	8
a	16	0	2	2	0	4	0	0	6	0	2	0	0	6	2	0
b	16	2	0	0	2	4	0	0	4	2	2	2	0	6	0	0
c	16	0	6	0	0	4	0	6	2	2	2	0	0	0	2	0
d	16	2	4	2	2	4	0	6	0	0	2	2	0	0	0	0
e	16	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
f	16	8	0	0	8	0	0	0	0	0	0	8	0	0	8	16

DDT in Table 1 has entry 0 for $(\Delta_i, \Delta_o) = (1, 5)$, while the BCT in Table 2 for $(\Delta_i, \nabla_o) = (1, 5)$ gives us probability 1. As far as the authors are aware, such an event has never been pointed out in previous works, and we expect that many existing boomerang attacks can be improved by considering superior switching effects represented in the BCT. To illustrate this point, we show in this paper how to improve the boomerang attack against 10-round `Deoxys-BC-384` which was recently presented [CHP+17]. We also use the BCT with related-tweakey boomerang characteristics of `SKINNY-64` and `SKINNY-128` presented by [LGL17]. The BCT allows us to accurately evaluate the amplification of the probability of forming distinguishers. As a result, we detect flaws on the experimentally evaluated probability in [LGL17], and probabilities for `SKINNY-64` are improved.

To better understand the relationship between the DDT and the BCT, we consider the problem of finding an S-box such that the maximum probability in the BCT is the same as one in the DDT. We show that while 2-uniform DDT always derives 2-uniform BCT, finding such an S-box with 4-uniform DDT is hard especially when the size of the S-box increases. Finally, we discuss the application of our idea to the modular addition operation. We show that the ladder switch observed for the S-box based designs can be applied to the modular addition, while the S-box switch cannot be applied. We also find a new switching mechanism called *MSB-switch* for modular addition which generates a right quartet with probability 1.

Finally, we would like to emphasise that the BCT should not be considered only from the attackers' point-of-view. One major feature of our approach is

that the BCT can (and should) also be considered by designers. A block-cipher designer need to evaluate many S-box choices according to various criteria. The simple form of the BCT, which allows one to measure the strength of the S-box against boomerang-style attacks independently from the other components (not too dissimilar to the relation between differential cryptanalysis and an S-box DDT) will be of great benefit to designers as well.

Outline. In Section 2, we give a brief overview of related work. Section 3 introduces the boomerang connectivity table as a new method to evaluate the probability of two differential characteristics, and explains the mechanisms based on which our improved switching technique can work. The BCT is applied to `Deoxys` and `SKINNY` in Section 4 and Section 5. We then discuss difficulties in finding 4-uniform BCT and extends the analysis to modular addition in Section 6. We present our conclusions in Section 7.

2 Previous Work

The boomerang attack, originally proposed by Wagner [Wag99], was extended to the related-key setting and was formalised in [BDK05] by using four related-key oracles K_1, $K_2 = K_1 \oplus \Delta K$, $K_3 = K_1 \oplus \nabla K$ and $K_4 = K_1 \oplus \Delta K \oplus \nabla K$.

Let $E_K(P)$ and $D_K(C)$ denote the encryption of P and the decryption of C under a key K, respectively. In the framework, a pair (P_1, P_2) with plaintext difference Δ_i is first queried to E_{K_1} and E_{K_2} to produce (C_1, C_2). Then (C_3, C_4) is computed from (C_1, C_2) by xoring ∇_o, and queried to decryption oracles D_{K_3} and D_{K_4} to produce (P_3, P_4). With probability p^2q^2, where $p^2q^2 > 2^{-n}$, the pair (P_3, P_4) will have difference Δ_i, and the cipher may be distinguished. The pseudo-code of the related-key boomerang attack is given below.

1. $\kappa_1 \leftarrow random(), \kappa_2 \leftarrow \kappa_1 \oplus \Delta K, \kappa_3 \leftarrow \kappa_1 \oplus \nabla K, \kappa_4 \leftarrow \kappa_1 \oplus \Delta K \oplus \nabla K$.
2. Repeat the following steps $\mathcal{N}$ times, where $\mathcal{N} \geq (pq)^{-2}$.
3. $P_1 \leftarrow random()$ and $P_2 \leftarrow P_1 \oplus \Delta_P$.
4. $C_1 \leftarrow E_{\kappa_1}(P_1)$ and $C_2 \leftarrow E_{\kappa_2}(P_2)$.
5. $C_3 \leftarrow C_1 \oplus \nabla_C$ and $C_4 \leftarrow C_2 \oplus \nabla_C$.
6. $P_3 \leftarrow D_{\kappa_3}(C_3)$ and $P_4 \leftarrow D_{\kappa_4}(C_4)$.
7. Check if $P_3 \oplus P_4 = \Delta_P$.

Boomerang-style attacks have been widely considered in symmetric-key cryptanalysis, and thus we refrain from providing a complete list of previous works that apply the technique. A few noticeable examples of boomerang-style attacks against block ciphers include [BDK05,BK09,LGL17,CHP+17].

3 BCT – Boomerang Connectivity Table

In this section we introduce our novel idea, the Boomerang Connectivity Table (BCT), which can be used to more accurately evaluate the probability of generating a right quartet in boomerang-style attacks. As briefly explained in Section 1,

the BCT is constructed by directly computing the probabilities for generating boomerang quartets at the local level (Equation 4), and thus provides more useful information for boomerang attacks when compared to the DDT, which was typically used in previous works.

3.1 Definition of the BCT

As illustrated in Figure 3, we consider the case where the input difference to the S-box, Δ_i, is defined by the sub-cipher E_0 and the output difference from the S-box, ∇_o, is defined by E_1. The important observation is that when one of the input values to the S-box is fixed, all the values in the quartet are fixed. Hence, the generation of the right quartet is a probabilistic event, which we can compute as:

$$\frac{\#\{x \in \{0,1\}^n | S^{-1}(S(x) \oplus \nabla_o) \oplus S^{-1}(S(x \oplus \Delta_i) \oplus \nabla_o) = \Delta_i\}}{2^n}.$$

The table that stores the results of this equation for all (Δ_i, ∇_o) is useful in the analysis of the target cipher. We call it "Boomerang Connectivity Table (BCT)".

Definition 3.1 (Boomerang Connectivity Table). *Let $S : \{0,1\}^n \to \{0,1\}^n$ be an invertible function, and $\Delta_i, \nabla_o \in \{0,1\}^n$. The Boomerang Connectivity Table (BCT) of S is given by a $2^n \times 2^n$ table $\mathcal{T}$, in which the entry for the (Δ_i, ∇_o) position is given by*

$$\mathcal{T}(\Delta_i, \nabla_o) = \#\{x \in \{0,1\}^n | S^{-1}(S(x) \oplus \nabla_o) \oplus S^{-1}(S(x \oplus \Delta_i) \oplus \nabla_o) = \Delta_i\}.$$

The BCT for the PRESENT S-box is shown in Table 2. We note that the complexity for generating the BCT for an n-bit to n-bit S-box is $O(2^{3n})$, which is higher than $O(2^{2n})$ for the DDT.

The BCT provides a unified representation of existing observations on quartet generation/probabilities for boomerang-style attacks, which can be easily detected on analysis of the cipher's S-box BCT.

Incompatibility. In previous works, the compatibility or incompatibility of (Δ_i, ∇_o) noted in [Mur11] would be typically checked experimentally. This can however be observed directly in the BCT: the difference pair (Δ_i, ∇_o) is incompatible if the corresponding entry of the BCT is 0.

Ladder switch. The value in any entry in the first row and the first column of the BCT is 2^n. This corresponds to the ladder switch proposed in [BK09]. This probability 1 transition can also be explained in the way of Figure 3. The case with $\Delta_i \neq 0$ and $\nabla_o = 0$ is illustrated in Figure 4. As we can observe, for any choice of x_1 and $x_2 (= x_1 \oplus \Delta_i)$, we have their images y_1 and y_2 after the S-box application. Now since $\nabla_o = 0$, no modification is made to create y_3 and y_4, and thus after S^{-1} is applied, the paired values get back to (x_1, x_2) with probability 1, and the second pair will always satisfy Δ_i. The same holds when $\Delta_i = 0$ and $\nabla_o \neq 0$.

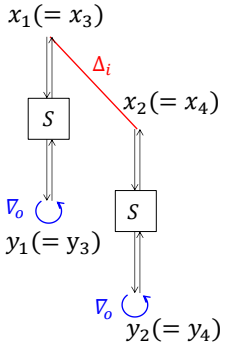

Fig. 4. Illustration of the ladder switch

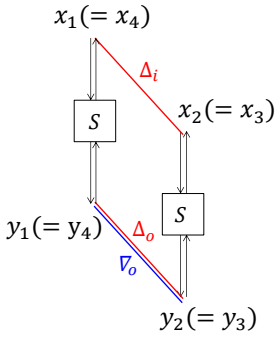

Fig. 5. Illustration of the S-box switch

S-box switch. The S-box switch can be explained in the context of the BCT as follows: if the DDT entry for (Δ_i, Δ_o) is non-zero, then by setting $\nabla_o = \Delta_o$, the BCT entry for (Δ_i, ∇_o) will take the same value. The mechanism of the S-box switch is the same as explained in [BK09], but here we explain it in the way of Figure 3, which will be useful to understand our new switching effects presented later. As illustrated in Figure 5, suppose that two input values x_1 and $x_2(= x_1 \oplus \Delta_i)$ are mapped to y_1 and y_2 satisfying $y_1 \oplus y_2 = \Delta_o$ with probability p. By setting $\nabla_o = \Delta_o$, y_3 and y_4 are computed by $y_1 \oplus \Delta_o$ and $y_2 \oplus \Delta_o$. This merely switches y_1 and y_2, and after S^{-1} is applied, the paired values become (x_2, x_1) with probability 1, and thus the second pair always satisfies Δ_i.

The above analysis, especially for the S-box switch, can be summarised as the following lemma about the relationship between the DDT and the BCT.

Lemma 1 *For any choice of (Δ_i, Δ_o), the value in the BCT is greater than or equal to the one in the DDT.*

Proof. The lemma is trivially valid when the value in the DDT is 0, or when $(\Delta_i, \Delta_o) = (0, 0)$. For the other non-zero DDT entries, the lemma follows from the discussion for the S-box switch above. □

BCT of the AES S-box. Because the PRESENT S-box does not offer the strongest resistance against maximum differential and linear probabilities, it may be interesting to study the properties of the BCT of the AES S-box, for example. The AES S-box is an 8-bit S-box, and thus the size of its DDT is 256×256. The properties of its DDT are well known: each column and row contain one entry with '4', 126 entries with '2', and the remaining is '0' (apart from the zero input and zero output differences). Hence in the entire DDT, the number of entries with '256', '4', '2' and '0' are 1, 255, 32130 and 33150, respectively.

In the BCT of the AES S-box, all entries for zero input difference (the first row) and zero output difference (the first column) are '256' owing to the ladder switch effect (similar to the BCT for the PRESENT S-box in Table 2). For the other entries, the maximum value of BCT is '6'. The number of entries with '256', '6', '4', '2' and '0' are 511, 510, 255, 31620 and 32640, respectively; these are summarised in Table 3. We also list the analysis of several other S-boxes having the same DDT structure in Table 3. Those include S-boxes of Camellia [AIK+00], TWINE [SMMK12], and Lilliput [BFMT16].

Table 3. Number of entries for each value for the DDT and BCT for the S-boxes in AES, Camellia, TWINE and Lilliput

Cipher	Table	256	6	4	2	0
AES	DDT	1	-	255	32130	33150
	BCT	511	510	255	31620	32640
Camellia	DDT	1	-	255	32130	33150
	BCT	511	510	255	31620	32640

Cipher	Table	16	6	4	2	0
TWINE	DDT	1	-	15	90	150
	BCT	31	30	15	60	120
Lilliput	DDT	1	-	15	90	150
	BCT	31	30	15	60	120

In Table 3, the following two facts deserve careful attention.

- The maximum non-trivial value in the BCT is '6', which is higher than the one in DDT. It means that for some Δ_i and $\Delta_o = \nabla_o$, generating a right quartet against an S-box can be easier than satisfying a differential transition for a pair.
- The number of zero entries in the BCT is smaller than in DDT. This means that even if DDT for (Δ_i, Δ_o) is 0, by setting $\nabla_o = \Delta_o$, a right quartet can be generated with (Δ_i, ∇_o).

The mechanisms behind these properties will be explained in the next subsection.

3.2 Increased Probability with Generalized Switching Effect

As shown in Lemma 1, each BCT entry may have a higher value than the corresponding entry in the DDT. This is caused by a new switching effect, but can be easily detected by considering the BCT.

Let us focus on the DDT entry for (Δ_i, Δ_o) whose value is '4,' namely Δ_i is propagated to Δ_o with probability 2^{-n+2}. In this case, there are two paired

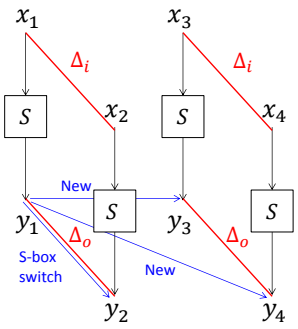

Fig. 6. Generalized switching effect: S-box switch and new switch

values such that the input difference is Δ_i and the difference after the S-box is Δ_o. This situation is illustrated in Figure 6.

Let $\mathcal{X}_{DDT}(\Delta_i, \Delta_o)$ and $\mathcal{Y}_{DDT}(\Delta_i, \Delta_o)$ be a set of paired values satisfying the differential transition from Δ_i to Δ_o.

$$\mathcal{X}_{\mathrm{DDT}}(\Delta_i, \Delta_o) \triangleq \{(a, b) \in \{0,1\}^n \times \{0,1\}^n : S(a) \oplus S(b) = \Delta_o, a \oplus b = \Delta_i\},$$
$$\mathcal{Y}_{\mathrm{DDT}}(\Delta_i, \Delta_o) \triangleq \{(S(a), S(b)) \in \{0,1\}^n \times \{0,1\}^n : S(a) \oplus S(b) = \Delta_o, a \oplus b = \Delta_i\}.$$

In the example in Figure 6, we have $\mathcal{X}_{\mathrm{DDT}}(\Delta_i, \Delta_o) = \{(x_1, x_2), (x_3, x_4)\}$ and $\mathcal{Y}_{\mathrm{DDT}}(\Delta_i, \Delta_o) = \{(y_1, y_2), (y_3, y_4)\}$.

Recall the strategy of the S-box switch, which sets $\nabla_o = \Delta_o$. Then for any $\mathcal{Y}_{\mathrm{DDT}}(\Delta_i, \Delta_o)$, $\mathcal{Y}_{\mathrm{DDT}} \oplus \nabla_o = \mathcal{Y}_{\mathrm{DDT}}$. Thus after the application of the inverse S-box, they will map back to $\mathcal{X}_{\mathrm{DDT}}(\Delta_i, \Delta_o)$. The essence of the S-box switch is finding a ∇_o for which $\mathcal{Y}_{\mathrm{DDT}} \oplus \nabla_o = \mathcal{Y}_{\mathrm{DDT}}$. Our observation of the generalized switching effect is that from two pairs in $\mathcal{Y}_{\mathrm{DDT}}(\Delta_i, \Delta_o)$, there are three ways to define such ∇_o:

$$\nabla_o \in \{y_1 \oplus y_2, y_1 \oplus y_3, y_1 \oplus y_4\}. \tag{5}$$

While one corresponds to the known S-box switch, the other two are new. Those choices of ∇_o are illustrated in Figure 6.

Thus, one entry of value '4' for Δ_i in the DDT will increase the value of two entries in the BCT, namely $(\Delta_i, y_1 \oplus y_3)$ and $(\Delta_i, y_1 \oplus y_4)$ by 4. Note that the BCT entry for $(\Delta_i, y_1 \oplus y_2)$ becomes '4', but the DDT of this entry is already '4' and we do not get an increase by 4. Let $y_{new} \in \{y_3, y_4\}$ and ℓ be a non-negative integer. The generalized switching effect can thus be summarised as follows:

$$\text{DDT for } (\Delta_i, y_1 \oplus y_{new}) \text{ is } 2\ell \Rightarrow \text{BCT for } (\Delta_i, y_1 \oplus y_{new}) \text{ is } 2\ell + 4.$$

From the above analysis, we obtain the following lemma about the relationship between the DDT and the BCT of an S-box.

Lemma 2 *For any fixed Δ_i, for each entry with '4' in the DDT, the value of two positions in the BCT will increase by 4.*

We omit the proof (it follows from the discussion above). We use instead the examples below to illustrate the lemma.

Example 1 *The row for $\Delta_i = \texttt{2}$ in the DDT in Table 1 contains an entry with '4.' This increases two entries of the BCT for $\Delta_i = \texttt{2}$. In fact, values for $\Delta_o = \texttt{3}$ and $\Delta_o = \texttt{6}$ in the BCT increase by 4 from the DDT, while the other non-trivial entries for $\Delta_i = \texttt{2}$ are exactly the same between the DDT and the BCT.*

Example 2 *The row for $\Delta_i = \texttt{9}$ in the DDT in Table 1 contains two entries with '4.' Values for $\Delta_o = \texttt{1}$ and $\Delta_o = \texttt{b}$ in the BCT increase by 4 from the DDT. The value for $\Delta_o = \texttt{f}$ is affected by both, thus increases by 8 from the DDT. The other non-trivial entries for $\Delta_i = \texttt{9}$ are exactly the same between the DDT and BCT.*

Note that Lemma 2 is about fixed Δ_i, but considering the symmetry, the same applies to any fixed ∇_o. In this paper, we omit lemmas for fixed ∇_o.

For 4-bit S-boxes, we propose a sufficient condition such that the S-box is free (has probability 1) with non-zero input and output differences using BCT.

Lemma 3 *For any 4-bit S-box, if the DDT has a row for some input difference Δ_i such that there are 4 entries of '4', then there exists an output difference ∇_o, such that (Δ_i, ∇_o) has probability 1 in the boomerang switch of this 4-bit S-box.*

Proof. Since the DDT has a row with 4 entries of '4' for some input difference Δ_i, we divide the input values into 4 sets $V_j = \{a_j, b_j, c_j, d_j\}$, s.t. $a_j \oplus b_j = c_j \oplus d_j = \Delta_i$ and $S(a_j) \oplus S(b_j) = S(c_j) \oplus S(d_j)$ for $j = 1, 2, 3, 4$. Each V_j corresponds to a boomerang quartet. Let $T_j = \{x_j, y_j, z_j, w_j\}$ for $j = 1, 2, 3, 4$ be the sets of output after S-box corresponding to V_j. Then $x_j \oplus y_j = z_j \oplus w_j = \Delta_{o,j}$ holds. Note that the row for Δ_i of DDT will have 4 non-zero entries in the columns for $\Delta_{o,j}$ for $j = 1, 2, 3, 4$. We define the 4 sets $D_j = \{x_j \oplus y_j, x_j \oplus z_j, x_j \oplus w_j\}$ for $j = 1, 2, 3, 4$ to store the XOR difference of T_j. Set $\nabla_o = D_1 \cap D_2 \cap D_3 \cap D_4$. Then if $\nabla_o \neq \emptyset$, (Δ_i, ∇_o) generates a quartet with probability 1. In fact, for any input value $x \in V_j$ with difference (Δ_i, ∇_o), we can verify that the second pair in the quartet will be exactly V_j. For example, suppose that the input pair $(x, x \oplus \Delta_i)$ is (a_1, b_1), the output will be (x_1, y_1). When ∇_o is applied, (x_1, y_1) must be changed to one of the $\{(y_1, x_1), (z_1, w_1), (w_1, z_1)\}$, which will have difference Δ_i after inverse S-box.

Then we only need to prove that $\nabla_o \neq \emptyset$ under the assumption that the DDT has 4 entries of '4' for Δ_i. Here we prove it experimentally as the mathematical proof is not trivial. While the number of all possible output of 4-bit bijective S-box is 16!, only those satisfying the condition imposed on T_j need to be checked. This greatly reduces the search space. We first choose 4 numbers from 0 to 15 as $\{x_1, y_1, z_1, w_1\}$. There are $Perm(16, 4) = 43680$ possible choices. But only 3360 choices satisfy $x_1 \oplus y_1 = z_1 \oplus w_1$, which are the valid choices of T_1. Similarly we can generate T_j for $j = 2, 3, 4$. The total number of valid $(T_1, ..., T_4)$ is around 2^{30}. Then we can compute D_j for $j = 1, 2, 3, 4$ and verify if ∇_o is empty. It takes less than 1 hour on a desktop to check all possible valid $(T_1, ..., T_4)$. The

result confirms that ∇_o is always non-empty. Therefore, we can conclude that the (Δ_i, ∇_o) has probability 1 with the generalized switching effect. □

Lemma 3 implies that having a row with four entries of '4' in the DDT may increase the power of the boomerang attack on those designs. This is an important observation since 4-bit S-boxes are widely used in lightweight designs.

Another observation is that the mechanism of the generalized switching effect requires the existence of a differential transition through the S-box with probability 2^{-n+2} or higher. In other words, the generalized switching effect does not exist in any 2-uniform DDT, which results in the following lemma.

Lemma 4 *For any S-box with 2-uniform DDT, the BCT is the same as the DDT but for the first row and the first column.*

We again omit the proof, and provide several examples.

Example 3 *The row for $\Delta_i =$ e in the DDT in Table 1 does not contain any entries with '4.' All the non-trivial entries for $\Delta_i =$ e are exactly the same between the DDT and BCT.*

Example 4 *When n is an odd number, n-bit S-boxes achieving 2-uniformity can be found easily. An example of such a 3-bit S-box is $S^{(3)} = [1, 7, 6, 3, 0, 2, 5, 4]$. The DDT and the BCT of $S^{(3)}$ are shown in Table 4 and Table 5, respectively, which clearly shows that besides the ladder switch, no generalized switching effect is available.*

Table 4. 2-uniform DDT of $S^{(3)}$

Δ_i \ Δ_o	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	2	0	0	2	2	0
2	0	0	0	0	2	2	2	2
3	0	2	2	0	2	0	0	2
4	0	2	0	2	0	2	0	2
5	0	0	2	2	0	0	2	2
6	0	2	0	2	2	0	2	0
7	0	0	2	2	2	2	0	0

Table 5. 2-uniform BCT of $S^{(3)}$

Δ_i \ ∇_o	0	1	2	3	4	5	6	7
0	8	8	8	8	8	8	8	8
1	8	2	2	0	0	2	2	0
2	8	0	0	0	2	2	2	2
3	8	2	2	0	2	0	0	2
4	8	2	0	2	0	2	0	2
5	8	0	2	2	0	0	2	2
6	8	2	0	2	2	0	2	0
7	8	0	2	2	2	2	0	0

3.3 Extension of Generalized Switching Effect to General DDT

The analysis in Section 3.2 applies only for the DDT whose maximum value is '4.' Although most of the existing S-boxes used in block ciphers were designed to satisfy this criterion, there has been a recent trend to weaken this criterion in

order to achieve higher efficiency. For example, the 4-bit S-box of GIFT [BPP+17] and the 8-bit S-box of SKINNY [BJK+16] have DDT whose maximum entry is higher than '4.' Motivated by these designs, we further extend the analysis in Section 3.2 to any 2ℓ-uniform DDT for a non-negative integer ℓ.

Recall that in the previous section, we explained that from one quartet there are two new ways to define ∇_o such that the BCT entry for (Δ_i, ∇_o) is higher than the corresponding one in the DDT by 4. When the DDT contains an entry of 2ℓ, where $\ell \geq 2$, there are ℓ paired values that satisfy the differential propagation. Then, $\binom{\ell}{2}$ distinct quartets can be constructed from ℓ paired values, which is illustrated in Figure 7 for $\ell = 3$.

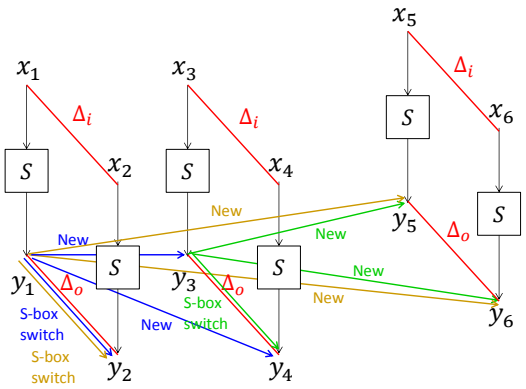

Fig. 7. Generalization of new switching effect. In total $\binom{3}{2} = 3$ distinct quartets are defined: $y_1y_2y_3y_4$ in blue, $y_1y_2y_5y_6$ in yellow, and $y_3y_4y_5y_6$ in green. Each quartet produces two new ways to define ∇_o.

Each of the $\binom{\ell}{2}$ quartets gives two new ways to define ∇_o such that the BCT entry for (Δ_i, ∇_o) is higher than the DDT by 4. Thus Lemma 2 is generalised as follows.

Lemma 5 *For any fixed Δ_i, for each entry with '2ℓ' in the DDT, the value of $2 \cdot \binom{\ell}{2}$ non-trivial positions in the BCT increase by 4.*

Example 5 *A single row $\Delta_i = 4$ of the DDT and BCT for the GIFT S-box is shown in Table 6. The DDT contains a single entry of '6' and '4'. Lemma 5 can be used to predict the sum of all the entries of the same row in the BCT. Namely, '6' in the DDT increases the sum in the BCT by $4 \cdot 2 \cdot \binom{3}{2} = 24$ and '4' in the DDT increases the sum in the BCT by $4 \cdot 2 \cdot \binom{2}{2} = 8$. Along with the ladder switch for the first column, the sum of entries for the BCT should be higher than the one in the DDT by $48 (= 24 + 8 + 16)$, which matches the actual BCT.*

Example 6 *Application of Lemma 5 to the 8-bit S-box of SKINNY-128 is discussed in Appendix A.*

Table 6. DDT and BCT of the GIFT S-box for $\Delta_i = 4$

Δ_o

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	sum
DDT	0	0	0	2	0	4	0	6	0	2	0	0	0	2	0	0	16
BCT	16	4	4	10	4	8	8	6	0	2	0	0	0	2	0	0	64

Lemma 5 shows that the impact from the DDT entry with 'x' to the BCT is large, on the order of x^2. Thus block-cipher designers adopting S-boxes with weak differential resistance need to be careful about how their choice will impact the corresponding BCT.

4 Applications to Deoxys-BC

In this section, we apply our BCT-based analysis to improve the recently proposed related-tweakey boomerang attacks against Deoxys-BC [CHP+17]. The specification of Deoxys-BC is briefly given in Section 4.1. The improved boomerang distinguishers are presented in Section 4.2, and the results of our experimental verification are reported in Section 4.3.

4.1 Specification

Deoxys-BC is an AES-based tweakable block cipher [JNPS16], which is based on the TWEAKEY framework [JNP14]. It is the underlying tweakable block cipher of the Deoxys authenticated encryption scheme submitted to the CAESAR competition (and one of the 15 candidates still being considered in the competition's third round). The Deoxys authenticated encryption scheme makes use of two versions of the cipher as its internal primitive: Deoxys-BC-256 and Deoxys-BC-384. Hereafter, we mainly focus on the specification of Deoxys-BC-384, which is a target in this paper. Deoxys-BC is a dedicated 128-bit tweakable block cipher which besides the two standard inputs, a plaintext P (or a ciphertext C) and a key K, also takes an additional input called a *tweak* T. The concatenation of the key and tweak states is called the *tweakey* state. For Deoxys-BC-384 the tweakey size is 384 bits. We assume that the reader is familiar with the AES block cipher [Nat01].

The round function of Deoxys-BC is exactly the same as that of the AES, except that the operation AddRoundKey is renamed as AddRoundTweakey. The internal state is viewed as a 4×4 matrix of bytes, and is updated by applying the following round function 14 times and 16 times for Deoxys-BC-256 and Deoxys-BC-384, respectively.

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

- MixColumns – Multiply the state by the 4×4 MDS matrix of AES.

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

Subtweakeys. The size of tweakey for Deoxys-BC-384 is 384 bits. Those are separated into three 128-bit words, and loaded into the initial tweakey states TK_0^1, TK_0^2, and TK_0^3. The 128-bit *subtweakey* used in the AddRoundTweakey operation is extracted from three tweakey states as $STK_i = TK_i^1 \oplus TK_i^2 \oplus TK_i^3 \oplus RC_i$, where RC_i is a round constant. Here, we omit the details of RC_i. Please refer to the original design document [JNPS16] for the exact specification.

In each round, the 128-bit words TK_i^1, TK_i^2, TK_i^3 are updated with the *tweakey schedule* algorithm, which is defined as

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

where the byte permutation h is

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

with the 16 bytes numbered by the usual AES byte ordering.

The $LFSR_2$ and $LFSR_3$ functions are simply the application of an LFSR to each on the 16 bytes of a 128-bit tweakey word. The two LFSRs used are given in Table 7 (x_0 stands for the LSB of the cell).

Table 7. The two LFSRs used in Deoxys-BC tweakey schedule

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

A schematic diagram of the instantiation of the TWEAKEY framework for Deoxys-BC-384 is shown in Figure 8.

4.2 Improved 10-Round Boomerang Attack

Cid et al. have recently presented in [CHP+17] several boomerang attacks in the related-tweakey setting, including 8-round, 9-round and 10-round boomerang distinguishers against Deoxys-BC-384 having probability 2^{-6}, 2^{-18}, and 2^{-42}, respectively. They proposed an MILP-based automated search method of differential characteristics that takes into account linear incompatibility in truncated differentials and the ladder switch effect in the boomerang attack. Among all

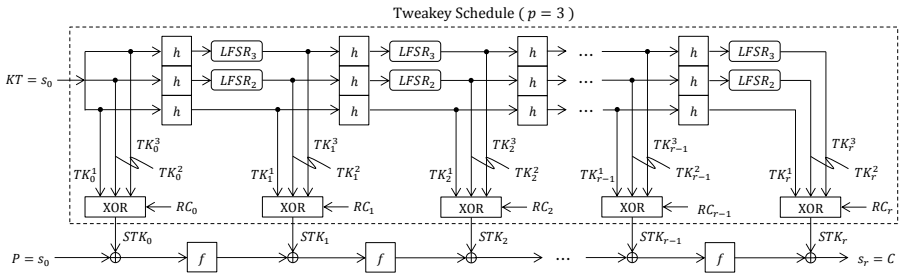

Fig. 8. A schematic diagram of `Deoxys-BC-384` with `TWEAKEY` framework

the possible differential characteristics, the authors chose the ones that exploit the S-box switch effect. Owing to the very detailed and careful optimisation, it seemed very unlikely that one could improve their proposed boomerang attacks; in other words, Cid et al. [CHP+17] picked the optimal choice under their assumptions on the search range.

However, our novel idea to use the BCT in boomerang-style attacks motivated us to improve their attacks by enlarging the search space when taking into account the generalised switching effect observed in the BCT of the AES S-box. In particular, their 8-round distinguisher includes only one active S-box that exploits the S-box switch effect, and hence an improvement by using BCT should be observed very clearly.

Our Goal. Recall that the maximum differential probability of the `AES` S-box is 2^{-6}, which is a reason why the probability of the 8-round distinguisher in [CHP+17] is 2^{-6}. As shown in Table 3, we observed in the BCT that the maximum probability of generating a quartet is $6/256 \approx 2^{-5.4}$ for the AES S-box. Hence, our goal here is to search for differential characteristics that achieve the probability of $2^{-5.4}$ and experimentally verify the correctness of the theory explained in Section 3.

In our analysis, we noticed that the authors of [CHP+17] interpreted the byte permutation h in the reverse order, thus their original analysis and results are in fact for a `Deoxys-BC` variant. Because our purpose here is demonstrate the possibility of improving existing attacks by use of the BCT, we analyse the same `Deoxys-BC` variant as in [CHP+17].

Searching for Differential Characteristics. We borrow the idea of the differential characteristic search proposed in [CHP+17]. Because the main focus of this paper is the generalised switching effect, we only briefly explain the search method.

The search in [CHP+17] is a two-stage approach. The first stage is searching for truncated differentials with the minimum number of active S-boxes using MILP. At this stage, there is no guarantee that each discovered truncated differ-

entials can be instantiated with actual differences. Here, the authors in [CHP+17] introduced two levels of tradeoff between the accuracy of truncated differentials and the assumption of the search range:

1. It only assumed independence between subtweakeys in different rounds, while the real different tweakeys are linearly related in the real cipher's algorithm, thus the truncated differentials detected in this approach may contain contradiction (often called "linear incompatibility").
2. Degrees of freedom (the number of differences that can be chosen independently of the other part of the trail) and the number of constraints for a valid trail (e.g. linear relations between subtweakeys mentioned above) were counted, and it was assumed that truncated differentials could be instantiated only if the degrees of freedom were higher than the degrees of consumption. Instead, truncated differentials that are detected in this way do not include contradiction about the linear incompatibility.

We refer to [CHP+17] for the exact MILP modelling for searching truncated differentials.

The second stage is searching for differences satisfying the given active-byte positions. This is done by listing all linear constraints in the truncated differential to build a system of linear equations, and by solving the system. We again refer to [CHP+17] for the exact method for generating the system.

The 10-round boomerang attack against `Deoxys-BC-384` uses 5-round differential characteristics for both E_0 and E_1. Active byte positions are chosen so that the ladder switch effect can be optimally exploited in the middle two rounds. Then the differential value is fixed to one of E_0 and E_1 and finally the differential value for the other half is fixed to exploit the S-box switch. The 10-round distinguisher [CHP+17] is given in Table 8. Cid *et al.* showed the differential propagation of E_0 in round 6 and of E_1 in round 5 to explicitly show that the ladder switch is applied. Both characteristics activate the S-box at position (1,1) in round 6 and both characteristics specify the same input and output difference (from `9e` to `68`), namely $\Delta_o = \nabla_o$, which is the condition to apply the S-box switch. The S-box is highlighted in red in Table 8. Note that in the DDT of the AES S-box, `9e` propagates to `68` with the highest probability of 2^{-6}.

We now replace the differential characteristic for the attack. Because of the optimisations done in [CHP+17], we use exactly the same differential characteristic for E_1, and only replace the difference of E_0. The characteristic for E_1 fixes the ∇_o of the target S-box to `68`. We confirmed that there exist two choices of Δ_i such that the BCT entry for $(\Delta_i, \texttt{68})$ is '6.' Those Δ_i are `2a` and `b4`. Hence, we added the linear equation $\Delta_i = \texttt{2a}$ or $\Delta_i = \texttt{b4}$ to the system of linear equations and solved the system to obtain the corresponding characteristics. The obtained differential characteristic for E_0 with $\Delta_i = \texttt{2a}$ is shown in Table 9.

4.3 Experimental Verification and Summary

As done in [CHP+17], we drop the first round and the last round of the 10-round boomerang characteristic, which leads to the 8-round boomerang characteristic

Table 8. 10-round distinguisher of `Deoxys-BC-384` [CHP+17]. † denotes the probability of the rounds that are evaluated for the boomerang switch. The probability is counted in the other half of the characteristic, thus the probability with † can be ignored.

rounds	initial Δ	tweakey Δ	before SB	after SR	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	$(2^{-6})^2$
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 68 00 00 ** 00 00 ** ** 00 ** ** 00	2^{-6}
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	$(2^{-12})^2$

Table 9. Improved differential characteristic for E_0 of `Deoxys-BC-384`.

rounds	initial Δ	tweakey Δ	before SB	after SR	p_r
1	00 00 15 00 b3 00 00 3f 2a 00 00 00 00 15 00 00	00 00 15 00 00 00 00 3f 2a 00 00 00 00 15 00 00	00 00 00 00 b3 00 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 00	$(2^{-6})^2$
2	00 00 00 12 00 00 00 1c 00 00 00 0e 00 00 00 0e	00 00 00 12 00 00 00 1c 00 00 00 0e 00 00 00 0e	00 00 00 00 00 00 00 00 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	0e 00 00 00 00 12 00 00 00 00 1c 00 00 00 00 0e	0e 00 00 00 00 12 00 00 00 00 1c 00 00 00 00 0e	** 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 3f 00 00 00 2a 00 00 00 15 00 00 00 15 00 00	** 3f 00 00 ** 2a 00 00 ** 15 00 00 ** 15 00 00	** ** 00 00 68 00 00 ** 00 00 ** ** 00 ** ** 00	$2^{-5.4}$
Master tweakey differences (ΔK) 00 00 ac 00 00 00 00 f4 58 00 00 00 00 ac 00 00 00 00 66 00 00 00 00 ab cd 00 00 00 00 66 00 00 00 00 df 00 00 00 00 60 bf 00 00 00 00 df 00 00					

only with a single active S-box now with the generalised switching effect. Our experiments clearly verify this effect.

Let κ_i, where $i \in \{1, 2, 3, 4\}$, be a 384-bit master tweakey for the first, second, third, and fourth oracles, respectively. Our experiments follow the pseudo-code in Section 2. The exact value of the master tweakey difference for E_1 denoted by ∇K is given in [CHP+17, Table 6]. We set $\mathcal{N}$ to 2^{15} and the number of attempts satisfying the last equation is counted. The test was iterated for 1,000 randomly chosen tweakeys; the average number of successes was 763. Hence, the probability of generating a right quartet is $763/2^{15} \approx 2^{-5.42}$, which closely matches and confirms the generalised switching effect.

We also derived the differential characteristic for $\Delta_i =$ `b4` and implemented the 8-round distinguisher for verification. In the experiments, the average number of successes over 1,000 different choices of keys was $775/2^{15} \approx 2^{-5.40}$, which again demonstrates the validity of the generalised switching effect.

Thus using the BCT for the AES S-box and the generalised switching effect, we were able to improve the probability of the boomerang distinguishers against `Deoxys-BC-384` by a factor of $2^{-0.6}$; namely to $2^{-5.4}$, $2^{-17.4}$, and $2^{-41.4}$ for 8 rounds, 9 rounds and 10 rounds, respectively. Although the improved factor in this particular case is small, the relevant point is that the effect of the generalised switch represented by the BCT could be experimentally verified against the AES

S-box. This indicates that the probability of boomerang distinguishers presented in previous works, which did not make use of the BCT, is unlikely to be optimal.

5 Applications to SKINNY

In [LGL17] Liu *et al.* proposed related-tweakey rectangle attacks against the SKINNY tweakable block cipher. The attacks evaluated the probability of generating a right quartet by taking into account the amplified probability, but did not consider the boomerang switch effect. In this section, we accurately evaluate the probability of generating the right quartet by applying the BCT. By doing so, we detect flaws in the experimentally evaluated probability in [LGL17] and show that the actual probabilities are higher than reported in [LGL17]. We first briefly review the specification of SKINNY in Section 5.1. The previous distinguishers and improved probabilities are then presented in Section 5.2 and Section 5.3, respectively.

5.1 Specification of SKINNY-128

SKINNY [BJK+16] is another family of lightweight tweakable block ciphers, based on the TWEAKEY framework [JNP14], which was introduced by Beierle *et al.* at CRYPTO 2016. The block size can be $n \in \{64, 128\}$ and the tweakey size can be $t \in \{n, 2n, 3n\}$. The 64-bit block version adopts a nibble-oriented SPN structure and is called SKINNY-64, while the 128-bit block version adopts a byte-oriented SPN structure and is called SKINNY-128.

An n-bit plaintext is loaded into the state represented by a 4×4-cell array, and the round function is then applied N_r times, where N_r is 40, 48 and 56 for n-bit, $2n$-bit and $3n$-bit tweakeys, respectively.

The round function consists of five operations: SubCells, AddRoundConstant, AddRoundTweakey, ShiftRowsand MixColumns.

SubCells. A 4-bit (resp. 8-bit) S-box whose maximum differential probability is 2^{-2} is applied to all cells in SKINNY-64 (resp. SKINNY-128).

AddRoundConstant. A 7-bit constant updated by an LFSR in every round is added to three cells of the state. Details of the LFSR can be found in [BJK+16].

AddRoundTweakey. A $n/2$-bit value is extracted from the n, $2n$ or $3n$-bit tweakey state, and is XORed to the upper half of the state. We omit the details of the tweakey schedule.

ShiftRows. Each cell in row j is rotated to the right (i.e. opposite to AES) by j positions.

MixColumns. Four cells in each column are multiplied by a binary matrix $\mathcal{M}$. When (i_0, i_1, i_2, i_3) is the 4-cell value input to $\mathcal{M}$, the output (o_0, o_1, o_2, o_3) is computed by $o_0 = i_0 \oplus i_2 \oplus i_3$, $o_1 = i_0$, $o_2 = i_1 \oplus i_2$, and $o_3 = i_0 \oplus i_2$. This is illustrated in Figure 9.

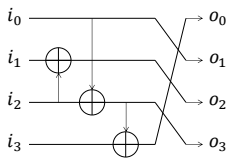

Fig. 9. A schematic representation of MixColumns of SKINNY

5.2 Previous Related-Tweakey Rectangle Attacks

Liu *et al.* [LGL17], among several cryptanalytic results, proposed 17-round, 18-round, 22-round and 23-round boomerang distinguishers against SKINNY-64-128, SKINNY-64-192, SKINNY-128-256, and SKINNY-128-384, respectively. The probabilities of those distinguishers are however too small to practically implement the verification experiments. Instead, the authors of [LGL17] implemented only the middle two rounds, the last round of E_0 and the first round of E_1, to experimentally verify that the proposed characteristics did not contain an incompatibility as pointed out by Murphy [Mur11]. If only the middle two rounds are evaluated, the probability including the amplified effect is calculated by $2^{-8.42}$, $2^{-16.30}$, $2^{-15.98}$ and $2^{-19.04}$ for the above four targets respectively, while their experimental verification implied that the probability should be $2^{-4.01}$, $2^{-7.53}$, $2^{-1.86}$, and $2^{-4.89}$, respectively. Those probabilities are summarised in Table 10.

Table 10. Previous boomerang distinguishers on SKINNY and our correction

Versions	$(\hat{p}\hat{q})^2$	Probability by Experiment	Our Corrected Probability
SKINNY-64-128	$2^{-8.42}$	$2^{-4.01}$	2^{-2}
SKINNY-64-192	$2^{-16.30}$	$2^{-7.53}$	$2^{-5.31}$
SKINNY-128-256	$2^{-15.98}$	$2^{-1.86}$	$2^{-1.86}$
SKINNY-128-384	$2^{-19.04}$	$2^{-4.89}$	0

Liu *et al.* mentioned in [LGL17] that one reason why the probabilities observed were higher than expected may be that some active Sboxes can be "saved", as the authors of [BK09] explained (the ladder switch and the S-box switch). They concluded that it is unlikely for the authors to overestimate the probability of the distinguishers.

This motivates us to apply the generalized switching effect of the BCT to explain the reasons behind their experimental results, and to improve their $\hat{p}^2\hat{q}^2$ probabilities to match the experimentally observed ones. We show that, while their experimental results cannot be explained only with the ladder switch and

S-box switch from [BK09], they can be explained rigorously by using the BCT along with the analysis for dependent S-boxes in [CLN+17].[7]

5.3 Precise Probability Evaluation of Boomerang Distinguishers

To explain the observed probabilities, we will use the attack against SKINNY-64-128. The last round (round 8) of E_0 and the first round (round 9) of E_1 are shown in Figure 10.

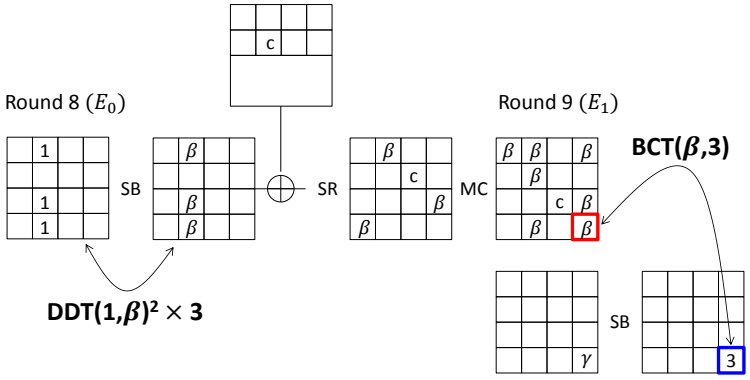

Fig. 10. Two rounds of 18-round distinguishers against SKINNY-64-128. Round 8 is covered by the characteristic in E_0 and round 9 is covered by the characteristic in E_1.

E_0 starts with three active nibbles with difference 1. Those will change into some difference in $\{0,1\}^4$ denoted by β. Then the difference c is introduced from the subtweakey difference.

In E_1, the differential propagation through the linear computations after the S-box is established with probability 1, thus omitted from Figure 10. In the end, E_1 consists only of a single S-box layer. It specifies that there is only one active S-box in round 9, with the output difference of the S-box 3 and the input difference that can be some value in $\{0,1\}^4$ denoted by γ.

In the straightforward evaluation with amplified probability, $\hat{p}$ is computed as $\left(4 \cdot (2^{-2})^2\right)^3 = 2^{-6}$, while $\hat{q}$ is calculated as $2 \cdot (2^{-2})^2 + 4 \cdot (2^{-3})^2 \approx 2^{-2.42}$. Thus $\hat{p}\hat{q} \approx 2^{-8.42}$ which matches the evaluation by the authors of [LGL17].

A careful analysis shows that the active S-boxes from E_0 in round 9 and the active S-boxes from E_1 in round 9 overlap each other in only one byte. E_0 specifies that the input difference Δ_i to the active S-box is β, while E_1 specifies

[7] Our experiments and theoretical explanation discovered different probabilities from the experiments by Liu *et al.* [LGL17]. We contacted the authors and confirmed that our evaluation is correct.

that the output difference ∇_o from the S-box is 3. This is exactly the situation in which the BCT can be applied to evaluate the probability of the active S-box and the other active S-boxes can be satisfied with probability 1 thanks to the ladder switch. Hence, we compute the probability of those two rounds as

$$\sum_{\beta\in\{0,1\}^4,\beta\neq 0}\Big(\frac{\mathcal{T}_{\mathrm{DDT}}(1,\beta)}{16}\Big)^2\cdot\frac{\mathcal{T}_{\mathrm{BCT}}(\beta,3)}{16}, \tag{6}$$

where $\mathcal{T}_{\mathrm{DDT}}(\Delta_i,\Delta_o)$ and $\mathcal{T}_{\mathrm{BCT}}(\Delta_i,\nabla_o)$ are the values of the DDT and the BCT for the input difference Δ_i and the output difference Δ_o or ∇_o, respectively. Those values for SKINNY's 4-bit S-box are summarised below.

β	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathcal{T}_{\mathrm{DDT}}(1,\beta)$	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0
$\mathcal{T}_{\mathrm{BCT}}(\beta,3)$	0	8	0	0	0	2	2	4	4	4	4	0	0	2	2

Hence, the probability can be calculated as $4\cdot(1/4)^2\cdot(1/4)=2^{-4}$.

The above evaluation using the BCT generally derives an approximated value under the assumption that the DDT and the BCT in consecutive two rounds can be evaluated independently. Given that the AddRoundTweakey operation updates only a half of the state, such an independent assumption cannot be established and the mechanism behind the experimental result is more complex.

Analysis including dependency of consecutive S-box applications. Here, an analysis involving several dependent S-boxes in [CLN+17] can be applied. By following [CLN+17] we introduce the notation:

$$\begin{aligned}\mathcal{X}_{\mathrm{DDT}}(\Delta_i,\Delta_o)&\triangleq\{x : S(x)\oplus S(x\oplus\Delta_i)=\Delta_o\},\\ \mathcal{Y}_{\mathrm{DDT}}(\Delta_i,\Delta_o)&\triangleq\{S(x) : S(x)\oplus S(x\oplus\Delta_i)=\Delta_o\}.\end{aligned}$$

And similarly for the BCT:

$$\begin{aligned}\mathcal{X}_{\mathrm{BCT}}(\Delta_i,\nabla_o)\triangleq&\{x : S^{-1}(S(x)\oplus\nabla_o)\oplus S^{-1}(S(x\oplus\nabla_i)\oplus\nabla_o)=\Delta_i\},\\ \mathcal{DX}_{\mathrm{BCT}}(\Delta_i,\nabla_o)\triangleq&\{x\oplus S^{-1}(S(x)\oplus\nabla_o) :\\ &\quad S^{-1}(S(x)\oplus\nabla_o)\oplus S^{-1}(S(x\oplus\Delta_i)\oplus\nabla_o)=\Delta_i\}.\end{aligned}$$

In the first S-box in round 8, the input difference 1 can change into one of {8, 9, a, b} with equal probability. We first consider the case for 8.

Case 1: $1\rightarrow 8$.

$$\mathcal{Y}_{\mathrm{DDT}}(1,8)=\{5,7,d,f\},\quad \mathcal{X}_{\mathrm{BCT}}(8,3)=\{4,6,c,e\},\quad \mathcal{DX}_{\mathrm{BCT}}(8,3)=\{2\}.$$

After the first S-box application in round 8, the paired values can take $\{5, 7, d, f\}$. They change to $\{4, 6, c, e\}$ with probability 2^{-2} after AddRoundTweakey and MixColumns. Here the source of randomness are subtweakey values xored to 8 nibbles of the state and other nibble values during the MixColumns operation. Then, after going through the propagation of the BCT with probability 1, the paired values $S^{-1}(C_3)$ and $S^{-1}(C_4)$ become $2 \oplus \{4, 6, c, e\} = \{4, 6, c, e\}$. This is a heavily S-box-dependent feature that the set of paired values does not change after the application of BCT.

During the backward computation for the second pair, ∇_o only impacts to one active nibble value, but as explained above, the set of possible values does not change. Thus during the inverse of MixColumns, the source of randomness does not change from the first pair. Hence all the values can return to the paired values with the same difference as the first pair with probability 1. In summary, in Case 1 a right quartet is generated with probability 2^{-2}.

Other cases. The analysis for the other three cases is similar.

Case 2 : $\mathcal{X}_{\mathrm{BCT}}(9, 3) = \{1, 3, 8, a\}, \mathcal{DX}_{\mathrm{BCT}}(9, 3) = \{b\}$.
Case 3 : $\mathcal{X}_{\mathrm{BCT}}(a, 3) = \{4, 6, c, e\}, \mathcal{DX}_{\mathrm{BCT}}(a, 3) = \{2\}$.
Case 4 : $\mathcal{X}_{\mathrm{BCT}}(b, 3) = \{1, 3, 8, a\}, \mathcal{DX}_{\mathrm{BCT}}(b, 3) = \{b\}$.

Thus, for any $\beta : \mathcal{T}(1, \beta) \neq 0$, $u \in \mathcal{X}_{\mathrm{BCT}}(\beta, 3)$ and $v \in \mathcal{DX}_{\mathrm{BCT}}(\beta, 3)$, the core property that $u \oplus v \in \mathcal{X}_{\mathrm{BCT}}(\beta, 3)$ is established. Hence after falling into each case, a right quartet is generated with probability 2^{-2}.

Finally, considering that each case occurs with probability $\frac{1}{4}$, the entire probability of generating a right quartet is $4 \cdot \frac{1}{4} \cdot 2^{-2} = 2^{-2}$. We implemented those two rounds and verified that the results match the above theory.

A similar analysis can be applied to other members of the SKINNY family to verify the experimental results in Table 10. We omit the details of those evaluation in this paper.

6 Discussion

The paper has so far mainly focused on the use of the BCT to improve previously proposed boomerang attacks. In this section, we considered further properties and aspects of the BCT. The hardness of finding S-boxes achieving 4-uniform BCT is explained in Section 6.1. Moreover, the boomerang switch for modular addition is discussed in Section 6.2, where we show that the switching effect is quite different for that operation.

6.1 Difficulties of Achieving 4-Uniform BCT

If the BCT provides the opportunity for attackers to improve their attack, as shown earlier in this paper, a natural question is therefore whether it is possible to find an S-box with minimum boomerang switching effect. As discussed in Example 4, finding such S-boxes for n-bit to n-bit S-box is easy when n is

odd, while in practice $n = 4$ and $n = 8$ are the most popular choices. In particular, most differentially strong S-boxes are designed to have 4-uniform DDT. Hence it is interesting to investigate whether 4-uniform DDT and BCT can be achieved simultaneously. Unfortunately, as we argue below, achieving 4-uniform BCT appears to be hard, especially as the size of the S-box increases, e.g. 8 bits.

Here, it is assumed that the differential spectrum of an AES-like S-box is used, i.e. the analysed S-box is an n-bit to n-bit S-box, and for each input and output difference of its DDT, there exist exactly one entry of '4' and $(2^{n/2}) - 2$ entries of '2.'

As in Lemma 2, each entry of '4' in the DDT increases two positions in the BCT for the same input or output difference by 4. To generate a 4-uniform BCT, the increased entries would have to have '0' in the DDT. Assume that the increased positions are chosen uniformly at random from all but zero. Then, the probability that the maximum value of the BCT in that row or column is '4' is

$$\frac{2^n/2}{2^n-1} \cdot \frac{2^n/2-1}{2^n-2} = \frac{2^{n-2}}{2^n-1}, \tag{7}$$

where the first term is the probability that the first increased position is chosen from '0' entries in the DDT, and the second term is for the second increased position. This must hold for $2^n - 1$ non-zero input or output differences, thus the probability is

$$\left(\frac{2^{n-2}}{2^n-1}\right)^{2^n-1}. \tag{8}$$

By setting $n = 4$ and 8, the probabilities that a randomly chosen S-box with a 4-uniform DDT simultaneously achieves 4-uniform BCT for 4-bit S-box and 8-bit S-box are $(4/15)^{15} \approx 2^{-28.6}$ and $(64/255)^{255} \approx 2^{-508.6}$, respectively.

For $n = 4$, if we consider the number of all 4-bit S-boxes with the optimal differential spectrum like the AES S-box, then it is unlikely that we find one that also achieves a 4-uniform BCT. Regarding $n = 8$, such an S-box may exist, but it is computationally hard to search for it.

6.2 Boomerang switch for modular addition

The early analysis in this paper considers the BCT for S-boxes. A natural extension is to study how to apply a BCT-type analysis to other non-linear operations. In this section, we consider the boomerang switch for modular addition.

While an S-box is an n-bit to n-bit mapping, modular addition maps $2n$-bit inputs to n-bit outputs. Thus the previous definition of BCT cannot be directly applied to modular addition, and we need a different way to define the BCT for modular addition.

Suppose that the target cipher is divided into E_0, a middle modular addition step, and E_1. Let $((x_1, x_1'), (x_2, x_2'), (x_3, x_3'), (x_4, x_4'))$ be a quartet of modular addition inputs, and (y_1, y_2, y_3, y_4) be the corresponding output quartet. In order to make the modular addition invertible, one of the addends needs to be fixed.

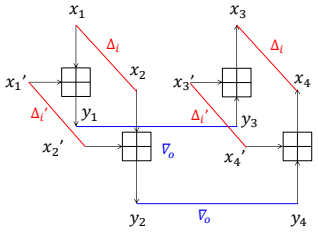

Fig. 11. A valid boomerang quartet for modular addition. Note that $x_3' = x_1'$, $x_4' = x_2'$.

Here we let x_i' for $i = 1, ..., 4$ be the fixed addends of the quartet. Thus $x_1' = x_3'$ and $x_2' = x_4'$. The input difference of modular addition specified by E_0 is (Δ_i, Δ_i'), namely $x_1 \oplus x_2 = x_3 \oplus x_4 = \Delta_i$ and $x_1' \oplus x_2' = x_3' \oplus x_4' = \Delta_i'$. The output difference specified by E_1 is ∇_o, namely $y_1 \oplus y_3 = y_2 \oplus y_4 = \nabla_o$. Figure 11 shows a valid boomerang quartet for modular addition.

The BCT for modular addition counts the number of inputs (x_i, x_i') such that the corresponding quartet with input difference (Δ_i, Δ_i') and output difference ∇_o is valid. Let '$\boxplus$' denote the modular addition and '$\boxminus$' the modular subtraction. The BCT for modular addition can then be defined in Equation (9). Table 11 and Table 12 give an example of the DDT and BCT for 3-bit modular addition when Δ_i is set to 0.

$$\mathcal{T}(\Delta_i, \Delta_i', \nabla_o) = \#\Big\{(x, x') \in \big(\{0,1\}^n, \{0,1\}^n\big) \,|\, \big((x \boxplus x') \oplus \nabla_o \boxminus x'\big) \oplus \Big(\big((x \oplus \Delta_i) \boxplus (x' \oplus \Delta_i') \oplus \nabla_o\big) \boxminus (x' \oplus \Delta_i')\Big) = \Delta_i\Big\} \tag{9}$$

Like the BCT for an S-box, it is easy to verify that the BCT for modular addition has a similar property in representing the *ladder switch* (see the first row and the first column in Table 12). Moreover, another interesting property, which we call *most significant bit (MSB) switch*, can also be observed for modular addition.

MSB switch. Suppose the output difference ∇_o specified by E_1 is on the most significant bit. Then the modular addition, with probability 1, generates a right boomerang quartet. This property can be derived by replacing the 'xor' of ∇_o with 'modular addition' in Equation (9). It can be observed in the column $\nabla_o = 4$ in Table 12.

Table 11. DDT of 3-bit modular addition with $\Delta_i = 0$

Δ_i' \ Δ_o	0	1	2	3	4	5	6	7
0	64	0	0	0	0	0	0	0
1	0	32	0	16	0	0	0	16
2	0	0	32	0	0	0	32	0
3	0	16	0	16	0	16	0	16
4	0	0	0	0	64	0	0	0
5	0	0	0	16	0	32	0	16
6	0	0	32	0	0	0	32	0
7	0	16	0	16	0	16	0	16

Table 12. BCT of 3-bit modular addition with $\Delta_i = 0$

Δ_i' \ ∇_o	0	1	2	3	4	5	6	7
0	64	64	64	64	64	64	64	64
1	64	0	32	0	64	0	32	0
2	64	64	0	0	64	64	0	0
3	64	0	32	0	64	0	32	0
4	64	64	64	64	64	64	64	64
5	64	0	32	0	64	0	32	0
6	64	64	0	0	64	64	0	0
7	64	0	32	0	64	0	32	0

On the other hand, the *S-box switch* does not work for modular addition. We can observe that in Table 11 the entry $(1, 1)$ is 32 while the corresponding entry in Table 12 is 0, which contradicts the result of S-box switch (Lemma 1).

The reason is that in the S-box switch, when the first pair of values are (x_1, x_2), the condition $\nabla_o = \Delta_o$ implies the S-box output (y_1, y_2) are swapped to (y_2, y_1). The paired output (y_2, y_1) are exactly the input of the inverse S-box to compute the second pair. However, for the modular addition with first pair of input $((x_1, x_1'), (x_2, x_2'))$, although the output of modular addition (y_1, y_2) are swapped to (y_2, y_1) under the condition $\nabla_o = \Delta_o$, the values x_1' and x_2' are not swapped. Thus, $((y_2, x_1'), (y_1, x_2'))$ will be the input of the inverse modular addition. Since y_2 and x_1' are not related, the original input difference is not guaranteed by the S-box switch.

Applications in Actual Ciphers. The analysis in Figure 11 can be directly applied to particular differential trails in ARX ciphers. As an example, we show the application in the SPECK32/64 cipher [BSS+13], in which the internal state in round i is composed of two 16-bit words l_{i-1} and r_{i-1} and the round function updates those values as $l_i \leftarrow (l_{i-1} \ggg 7) \boxplus r_{i-1} \oplus k_i$ and $r_i \leftarrow (r_{i-1} \lll 2) \oplus l_i$. Then, the above BCT corresponds to the probability of the modular addition in a single round of SPECK with $\Delta l_{i-1} = 0$, $\Delta r_{i-1} = \Delta_i'$, and $\Delta l_i = \nabla_o$. Note that the ladder switch can be applied to the right word as long as active bit positions in $\Delta r_{i-1} \lll 2$ and $\Delta r_i \oplus \Delta l_i$ do not overlap.

For example, with $\Delta r_{i-1} = \texttt{8000}$ and with any choice of Δl_i and Δr_i such that $(\Delta l_i \oplus \Delta r_i) \wedge \texttt{0002} = 0$, the MSB switch is applied to the modular addition and the ladder switch is applied to the right word. Hence, the probability r for one middle round is 1. Similarly, incompatible choices of $(\Delta r_{i-1}, \Delta l_i)$ with respect to the modular addition can be easily checked by using the BCT.

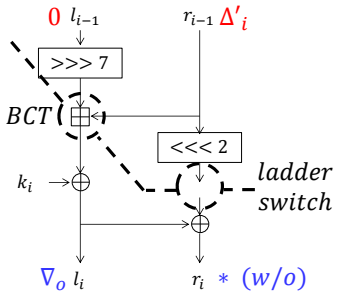

Fig. 12. Application of BCT for SPECK

7 Concluding Remarks

In this paper, we introduced the BCT as a generalised method to analyse the dependency of two differential characteristics in boomerang distinguishers. The BCT includes the existing observations of incompatibility between two characteristics, as well as the ladder switch and the S-box switch. Moreover, the BCT offers stronger switching effect than previous ones, and we analysed the mechanism why such an effect is generated. The larger the bias in the DDT becomes, the more advantages the BCT provides. Future primitive designers who wish to adopt differentially weak S-boxes should take into account the impact of their choices on the BCT.

The effect of the BCT-based analysis was demonstrated by improving the boomerang attacks against `Deoxys-BC` and by precisely evaluating the probability of previous boomerang distinguishers against `SKINNY`.

We also discussed the issue of searching for S-boxes having good BCT, and showed that the S-boxes having 2-uniform DDT always have 2-uniform BCT, while S-boxes having 4-uniform DDT usually cannot ensure 4-uniform BCT. Lastly, we extended the analysis to modular addition along with an application to SPECK, and explained the different behaviours between the BCT for a S-box and the BCT for the modular addition.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. We also thank attendees of the 2018 Dagstuhl seminar for Symmetric Cryptography, who provided us with various comments. The last author is supported by the Fundamental Theory and Cutting Edge Technology Research Program of Institute of Information Engineering, CAS (Grant No. Y7Z0341103), Youth Innovation Promotion Association CAS and the National Natural Science Foundation of China (Grants No. 61472415, 61732021 and 61772519).

A Demonstration of Lemma 5 for SKINNY-128

The size of the DDT for the 8-bit S-box of SKINNY-128 is 256×256. Each entry can take one of 13 different values but for 0 and 256: 2, 4, 6, 8, 12, 16, 20, 24, 28, 32, 40, 48 and 64. Hence, the impact to the BCT is much bigger than for many other S-boxes, making it a good target for verifying the correctness of Lemma 5.

Each row of Table 13 shows the number of entries with the designated value in the DDT. For example, when $\Delta_i = 01$, there are 6, 3 and 1 entries that take 16, 32 and 64, respectively. The column of "sum" shows the sum of the values of the BCT entries that were computed experimentally. The column of "Lem. 5" shows that value calculated by applying Lemma 5. Due to the limited space we only list the data for $\Delta_i = 1$ to 100.

As in Table 13, for any Δ_i, the relationships between the DDT and the BCT are correctly simulated by Lemma 5.

Table 13. Relationships between the DDT and the BCT simulated by Lemma 5

Δ_i	2	4	6	8	12	16	20	24	28	32	40	48	64	sum	Lem. 5	Δ_i	2	4	6	8	12	16	20	24	28	32	40	48	64	sum	Lem. 5
01	0	0	0	0	0	6	0	0	0	3	0	0	1	8704	8704	33	0	24	0	14	0	3	0	0	0	0	0	0	0	2048	2048
02	0	0	0	0	0	0	0	0	0	4	0	0	2	12288	12288	34	0	16	0	16	0	4	0	0	0	0	0	0	0	2304	2304
03	0	0	0	18	0	5	0	0	0	1	0	0	0	3456	3456	35	0	0	0	16	0	8	0	0	0	0	0	0	0	3072	3072
04	0	0	0	12	0	2	0	0	0	2	0	0	1	7424	7424	36	0	24	0	14	0	3	0	0	0	0	0	0	0	2048	2048
05	0	0	0	0	0	8	0	0	0	2	0	0	1	8192	8192	37	0	32	0	16	0	0	0	0	0	0	0	0	0	1536	1536
06	0	0	0	6	0	7	0	0	0	3	0	0	0	5248	5248	38	0	12	0	10	0	8	0	0	0	0	0	0	0	2880	2880
07	0	0	0	12	0	6	0	0	0	2	0	0	0	4352	4352	39	0	12	0	10	0	8	0	0	0	0	0	0	0	2880	2880
08	0	0	0	6	0	3	0	0	0	3	0	0	1	8320	8320	3a	0	12	0	16	0	1	0	0	0	2	0	0	0	3520	3520
09	0	0	0	6	0	3	0	0	0	3	0	0	1	8320	8320	3b	0	12	0	14	0	4	0	0	0	1	0	0	0	3136	3136
0a	0	0	0	12	0	2	0	0	0	2	0	0	1	7424	7424	3c	16	16	0	20	0	0	0	0	0	0	0	0	0	1600	1600
0b	0	0	0	6	0	9	0	0	0	2	0	0	0	4736	4736	3d	16	16	0	20	0	0	0	0	0	0	0	0	0	1600	1600
0c	0	12	0	12	0	5	0	0	0	1	0	0	0	3264	3264	3e	16	32	0	4	0	4	0	0	0	0	0	0	0	1856	1856
0d	0	12	0	12	0	5	0	0	0	1	0	0	0	3264	3264	3f	16	32	0	4	0	4	0	0	0	0	0	0	0	1856	1856
0e	0	12	0	12	0	5	0	0	0	1	0	0	0	3264	3264	40	0	0	0	4	0	4	0	0	0	3	0	0	1	8448	8448
0f	0	12	0	12	0	5	0	0	0	1	0	0	0	3264	3264	41	0	8	0	16	0	6	0	0	0	0	0	0	0	2688	2688
10	0	0	0	0	0	4	0	0	0	2	0	0	2	11264	11264	42	0	8	0	8	0	5	0	0	0	1	0	1	0	5248	5248
11	0	0	0	12	0	6	0	0	0	2	0	0	0	4352	4352	43	16	27	0	11	1	1	0	0	0	0	0	0	0	1600	1600
12	0	0	0	16	0	8	0	0	0	0	0	0	0	3072	3072	44	0	27	0	9	1	2	0	0	0	1	0	0	0	2688	2688
13	0	24	0	14	0	3	0	0	0	0	0	0	0	2048	2048	45	0	16	0	16	0	2	0	0	0	1	0	0	0	2816	2816
14	0	16	0	16	0	4	0	0	0	0	0	0	0	2304	2304	46	16	20	0	11	0	2	0	1	0	0	0	0	0	2176	2176
15	0	0	0	16	0	8	0	0	0	0	0	0	0	3072	3072	47	16	23	0	10	1	1	0	1	0	0	0	0	0	2048	2048
16	0	24	0	14	0	3	0	0	0	0	0	0	0	2048	2048	48	8	20	0	14	0	3	0	0	0	0	0	0	0	2016	2016
17	0	32	0	16	0	0	0	0	0	0	0	0	0	1536	1536	49	8	20	0	14	0	3	0	0	0	0	0	0	0	2016	2016
18	0	12	0	14	0	2	0	0	0	2	0	0	0	3648	3648	4a	8	15	0	13	1	4	0	0	0	0	0	0	0	2272	2272
19	0	12	0	14	0	2	0	0	0	2	0	0	0	3648	3648	4b	8	20	0	11	0	1	0	1	0	1	0	0	0	2912	2912
1a	0	12	0	12	0	7	0	0	0	0	0	0	0	2752	2752	4c	35	22	1	8	1	1	0	0	0	0	0	0	0	1440	1440
1b	0	12	0	10	0	8	0	0	0	0	0	0	0	2880	2880	4d	35	22	1	8	1	1	0	0	0	0	0	0	0	1440	1440
1c	16	32	0	4	0	4	0	0	0	0	0	0	0	1856	1856	4e	27	30	1	6	1	1	0	0	0	0	0	0	0	1408	1408
1d	16	32	0	4	0	4	0	0	0	0	0	0	0	1856	1856	4f	27	30	1	6	1	1	0	0	0	0	0	0	0	1408	1408
1e	16	16	0	20	0	0	0	0	0	0	0	0	0	1600	1600	50	0	0	0	4	0	4	0	0	0	3	0	0	1	8448	8448
1f	16	16	0	20	0	0	0	0	0	0	0	0	0	1600	1600	51	0	8	0	16	0	6	0	0	0	0	0	0	0	2688	2688
20	0	0	0	0	0	0	0	0	0	4	0	0	2	12288	12288	52	0	8	0	8	0	5	0	0	0	1	0	1	0	5248	5248
21	0	0	0	0	0	6	0	0	0	3	0	0	1	8704	8704	53	16	27	0	11	1	1	0	0	0	0	0	0	0	1600	1600
22	0	0	0	0	0	16	0	0	0	0	0	0	0	4096	4096	54	0	27	0	9	1	2	0	0	0	1	0	0	0	2688	2688
23	0	0	0	18	0	5	0	0	0	1	0	0	0	3456	3456	55	0	16	0	16	0	2	0	0	0	1	0	0	0	2816	2816
24	0	0	0	12	0	10	0	0	0	0	0	0	0	3328	3328	56	16	20	0	11	0	2	0	1	0	0	0	0	0	2176	2176
25	0	0	0	8	0	10	0	0	0	1	0	0	0	4096	4096	57	16	23	0	10	1	1	0	1	0	0	0	0	0	2048	2048
26	0	0	0	22	0	3	0	0	0	1	0	0	0	3200	3200	58	8	20	0	14	0	3	0	0	0	0	0	0	0	2016	2016
27	0	0	0	28	0	2	0	0	0	0	0	0	0	2304	2304	59	8	20	0	14	0	3	0	0	0	0	0	0	0	2016	2016
28	0	0	0	18	0	5	0	0	0	1	0	0	0	3456	3456	5a	8	15	0	13	1	4	0	0	0	0	0	0	0	2272	2272
29	0	0	0	18	0	5	0	0	0	1	0	0	0	3456	3456	5b	8	20	0	11	0	1	0	1	0	1	0	0	0	2912	2912
2a	0	0	0	24	0	4	0	0	0	0	0	0	0	2560	2560	5c	35	22	1	8	1	1	0	0	0	0	0	0	0	1440	1440
2b	0	0	0	18	0	5	0	0	0	1	0	0	0	3456	3456	5d	35	22	1	8	1	1	0	0	0	0	0	0	0	1440	1440
2c	0	28	0	16	0	1	0	0	0	0	0	0	0	1728	1728	5e	27	30	1	6	1	1	0	0	0	0	0	0	0	1408	1408
2d	0	28	0	16	0	1	0	0	0	0	0	0	0	1728	1728	5f	27	30	1	6	1	1	0	0	0	0	0	0	0	1408	1408
2e	0	28	0	16	0	1	0	0	0	0	0	0	0	1728	1728	60	0	0	0	4	0	8	0	0	0	3	0	0	0	5376	5376
2f	0	28	0	16	0	1	0	0	0	0	0	0	0	1728	1728	61	0	8	0	16	0	6	0	0	0	0	0	0	0	2688	2688
30	0	0	0	0	0	4	0	0	0	2	0	0	2	11264	11264	62	0	8	0	16	0	6	0	0	0	0	0	0	0	2688	2688
31	0	0	0	12	0	6	0	0	0	2	0	0	0	4352	4352	63	16	27	0	11	1	1	0	0	0	0	0	0	0	1600	1600
32	0	0	0	16	0	8	0	0	0	0	0	0	0	3072	3072	64	0	27	0	17	1	0	0	0	0	0	0	0	0	1664	1664

References

AIK+00. Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita. Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In Douglas R. Stinson and Stafford E. Tavares, editors, *SAC 2000*, volume 2012 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2000.

BCD03. Alex Biryukov, Christophe De Cannière, and Gustaf Dellkrantz. Cryptanalysis of SAFER++. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2003.

BDK01. Eli Biham, Orr Dunkelman, and Nathan Keller. The Rectangle Attack - Rectangling the Serpent. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, 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, *FSE 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, *EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 507–525. Springer, 2005.

BFMT16. Thierry P. Berger, Julien Francq, Marine Minier, and Gaël Thomas. Extended generalized feistel networks using matrix representation to propose a new lightweight block cipher: Lilliput. *IEEE Trans. Computers*, 65(7):2074–2089, 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, *CRYPTO 2016, 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, *ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

BKL+07. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

BPP+17. Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption, 2017. Cryptology ePrint Archive, Report 2017/622.

BS93. Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer, 1993.

BSS+13. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404, 2013.

CHP+17. Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. A security analysis of deoxys and its internal tweakable block ciphers. *IACR Trans. Symmetric Cryptol.*, 2017(3):73–107, 2017.

CLN+17. Anne Canteaut, Eran Lambooij, Samuel Neves, Shahram Rasoolzadeh, Yu Sasaki, and Marc Stevens. Refined probability of differential characteristics including dependency between multiple rounds. *IACR Trans. Symmetric Cryptol.*, 2017(2):203–227, 2017.

DKS10. Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2010.

DKS14. Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. *J. Cryptology*, 27(4):824–849, 2014.

GPPR11. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011.

JNP14. Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, 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.

KHP+12. Jongsung Kim, Seokhie Hong, Bart Preneel, Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key boomerang and rectangle attacks: Theory and experimental analysis. *IEEE Trans. Information Theory*, 58(7):4948–4966, 2012.

KKS00. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 75–93. Springer, 2000.

LGL17. Guozhen Liu, Mohona Ghosh, and Song Ling. Security analysis of SKINNY under related-tweakey settings (long paper). *IACR Trans. Symmetric Cryptol.*, 2017(3):37–72, 2017.

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

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

SMMK12. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE : A lightweight block cipher for multiple platforms. In Lars R. Knudsen and Huapeng Wu, editors, *SAC 2012*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer, 2012.

Wag99. David Wagner. The Boomerang Attack. In Lars R. Knudsen, editor, *FSE '99*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.