New Properties of Double Boomerang Connectivity Table

Qianqian Yang\textsuperscript{1,3}, Ling Song\textsuperscript{2,4,5}, Siwei Sun\textsuperscript{5,6}, Danping Shi\textsuperscript{1,3} and Lei Hu\textsuperscript{1,3}

\textsuperscript{1} State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
\texttt{yangqianqian@iie.ac.cn,shidanping@iie.ac.cn,hulei@iie.ac.cn}

\textsuperscript{2} College of Cyber Security, Jinan University, Guangzhou, China
\texttt{songling.qs@gmail.com}

\textsuperscript{3} School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

\textsuperscript{4} National Joint Engineering Research Center of Network Security Detection and Protection Technology, Jinan University, Guangzhou, China

\textsuperscript{5} School of Cryptology, University of Chinese Academy of Sciences, Beijing, China
\texttt{sunsiwei@ucas.ac.cn}

\textsuperscript{6} State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

\texttt{shidanping@iie.ac.cn,hulei@iie.ac.cn}

\texttt{songling.qs@gmail.com}

\texttt{sunsiwei@ucas.ac.cn}

\texttt{hulei@iie.ac.cn}

\texttt{yangqianqian@iie.ac.cn}

\texttt{shidanping@iie.ac.cn}

\texttt{hulei@iie.ac.cn}

\texttt{songling.qs@gmail.com}

\texttt{sunsiwei@ucas.ac.cn}

\texttt{hulei@iie.ac.cn}

\texttt{yangqianqian@iie.ac.cn,shidanping@iie.ac.cn,hulei@iie.ac.cn}

Abstract. The double boomerang connectivity table (DBCT) is a new table proposed recently to capture the behavior of two consecutive S-boxes in boomerang attacks. In this paper, we observe an interesting property of DBCT of S-box that the ladder switch and the S-box switch happen in most cases for two continuous S-boxes, and for some S-boxes only S-box switch and ladder switch are possible. This property implies an additional criterion for S-boxes to resist the boomerang attacks and provides as well a new evaluation direction for an S-box. Using an extension of the DBCT, we verify that some boomerang distinguishers of TweAES and Deoxys are flawed. On the other hand, inspired by the property, we put forward a formula for estimating boomerang cluster probabilities. Furthermore, we introduce the first model to search for boomerang distinguishers with good cluster probabilities. Applying the model to CRAFT, we obtain 9-round and 10-round boomerang distinguishers with a higher probability than that of previous works.

Keywords: boomerang attack · DBCT · cluster · CRAFT · TweAES · Deoxys-BC

1 Introduction

Differential cryptanalysis, proposed by Biham and Shamir [BS91], is one of the most powerful techniques to assess the security of block ciphers. The main idea is to search for non-random pairs of input and output differences of the cipher with high probability. In many cases, it is hard to find long differentials. To overcome the restriction, Wagner introduced the boomerang attack in [Wag99], which is a development of differential cryptanalysis. The main idea of boomerang attacks is to combine two short differentials with high probabilities to get a long one. In boomerang attacks, a cipher $E$ is regarded as the composition of two sub-ciphers, i.e., $E = E_1 \circ E_0$. Suppose there exists two differentials $\Delta_1 \rightarrow \Delta_2$ for $E_0$ and $\nabla_2 \rightarrow \nabla_3$ for $E_1$ with probabilities $p$ and $q$. Then it is a boomerang distinguisher of probability:

$$
P(E^{-1}(E(P) \oplus \nabla_3) \oplus E^{-1}(E(P \oplus \Delta_1) \oplus \nabla_3) = \Delta_1) = P_{E_0} \cdot P_{E_1} = p^2 \cdot q^2,$$

when the two differentials are independent.

Licensed under Creative Commons License CC-BY 4.0.
Later, researchers focused on the dependence and the connectivity of two differentials. At Asiacrypt 2009, Biryukov et al. proposed three types of switches to evaluate the transition between the differentials of $E_0$ and $E_1$ more exactly in [BK09]. Then Dunkelman et al. proposed the sandwich attack [DKS10, DKS14], which divides the cipher into three parts, i.e., $E = E_1 \circ E_m \circ E_0$ where $E_m$ contains dependency. Further, Cid et al. gave a new tool named Boomerang Connectivity Table (BCT) to calculate the probability of one middle round theoretically [CHP+18]. Expanding one middle round to multiple middle rounds, Song et al. gave a generalized framework for the BCT and introduced a method to precisely calculate the probability of boomerang distinguishers [SQH19].

As shown in [SQH19], the clustering effect, which raises the probability of a boomerang distinguisher of SKINNY block cipher from $2^{-163.84}$ to $2^{-77.83}$, is significant for word-oriented block ciphers in boomerang attacks. In fact, the clustering effect of $E_m$ can be considered only when dependency in $E_m$ is well handled. This encourages a line of research on boomerang attacks that searches for good boomerang distinguishers with dependencies being taken into account. In [CHP+18], Cid et al. used an MILP model to study the ladder switch for boomerang attacks on Deoxys. In [LS19], Liu et al. proposed a generic approach searching for boomerang distinguishers on GIFT, where there is one middle round. Furthermore, Delaune et al. [DDV20] introduced a new approach to search for boomerang distinguishers by carefully dealing with dependencies without the need of specifying the middle rounds. Typically, these works search for good truncated boomerang characteristics first and then instantiate them.

In the most recent work [DDV20, HBS21], we observe that to handle dependency intermediate differences are treated as random for simplicity. However, this may be not true in certain cases. Besides, the clustering effect of the two outer parts $E_0$ and $E_1$ are neglected in the search for a truncated boomerang. So there needs a more careful treatment of dependency in $E_m$ and the cluster effect over two outer parts.

**Contribution.** In this paper, we look into the double boomerang connectivity table (DBCT) which captures the properties of two continuous S-boxes in boomerang attacks and find new properties of DBCT, showing that the relation between neighbouring rounds cannot be ignored. We first observe that besides the common ladder switch and S-box switch, few other cases exist for two continuous active S-boxes and even for certain S-boxes, no other cases exist. When the cases besides the ladder switch and the S-box switch are rarer, it is more desirable for a cipher against the boomerang attack. Thus, the uniformity of DBCT for an S-box can be defined as an additional criterion for resisting the boomerang attack. The DBCT uniformity matters, which can be confirmed by comparing probabilities of the same boomerang distinguisher furnished with different S-boxes.

Further, we extend DBCT to the case of multiple active S-boxes in a row and the case where there is a complex linear layer in between. Applying extensions of DBCT to TweAES [CDJ+20] and Deoxys [JNPS16], we verify that a boomerang distinguisher of TweAES proposed by the designers is flawed, as well as two boomerang distinguishers of Deoxys–BC proposed in [BL22]. Once again, it demonstrates that the interactions between two S-box layers matter and should be treated carefully.

On the other hand, the properties of DBCT remind us to count only once instead of twice when there are two (or multiple) continuous active S-boxes. Based on this, we put forward a formula for evaluating the probability of the full boomerang cluster. We divide the cipher $E$ into three sub-ciphers $E = E_1 \circ E_m \circ E_0$. Inspired by the method for evaluating the probability of truncated differentials, proposed by Moriai et al. in [MSAK99], we give the cluster probability of $E_0$ and $E_1$ similarly. The same technique applies to $E_m$ due to the property of DBCT. Furthermore, we propose the first MILP model for searching boomerang clusters. What’s more, it keeps the feature that there is no need for specifying the middle rounds. Applying our MILP model to CRAFT, we obtain good boomerang clusters for
6–14 rounds of CRAFT. Among them, the probabilities of 9-round and 10-round boomerang clusters are higher than those proposed in [HBS21].

**Organization of the paper.** The rest of the paper is organized as follows: in Section 2, we present the preliminaries for the boomerang attack and sandwich attack, and recall the previous definitions of tables used in boomerang attack. In Section 3, we propose some new properties for double boomerang connectivity table (DBCT) of S-box and introduce extensions of DBCT. In Section 4, we revisit some existing boomerang distinguishers of CRAFT, TweAES and Deoxys, respectively, and verify that the boomerang distinguisher of TweAES and Deoxys are flawed. Using the property of DBCT, we give a new approach to simplify the formula for the probability of boomerang clusters and propose an MILP model to search for boomerangs with good cluster probability. Applying the model to CRAFT, we obtain 9-round and 10-round boomerang distinguisher with higher probability than previous works in Section 5. Finally, we conclude the paper in Section 6.

# 2 Preliminaries

## 2.1 Boomerang Attack

In [Wag99], Wagner proposed the boomerang attack which is grounded in the idea that combining two short differentials may lead to a good long one. In the left of Fig. 1 where a block cipher $E$ is treated as the composition of two sub-ciphers $E_0$ and $E_1$, we suppose there exist two short differentials $\Delta_1 \rightarrow \Delta_2$ and $\nabla_3 \rightarrow \nabla_2$ with high probability $p$ and $q$. Under the assumption that the two differentials are independent, the probability of the boomerang distinguisher of $E$ is

$$P(E^{-1}(E(P) \oplus \nabla_3) \oplus E^{-1}(E(P \oplus \Delta_1) \oplus \nabla_3) = \Delta_1) = P_{E_0} \cdot P_{E_1} = p^2 \cdot q^2.$$  

The sandwich attack, proposed by Dunkelman et al. in [DKS10, DKS14], is an improvement to the boomerang attack. Instead of assuming the two trails are independent, it takes into account the dependency between the two differentials and handles it in a middle part $E_m$, as shown in the right of Fig. 1. Thus, the sandwich attack regards cipher $E$ as the composition of three sub-ciphers $E = E_1 \circ E_m \circ E_0$, where $E_m$ usually contains a small number of rounds. If the probability of a boomerang coming back over $E_m$ for random inputs $x$ is $P(E_m^{-1}(E_m(x) \oplus \nabla_2) \oplus E_m^{-1}(E_m(x \oplus \Delta_2) \oplus \nabla_2) = \Delta_2) = r$, then the probability of the whole boomerang distinguisher is

$$P(E^{-1}(E(P) \oplus \nabla_3) \oplus E^{-1}(E(P \oplus \Delta_1) \oplus \nabla_3) = \Delta_1) = P_{E_0} \cdot P_{E_m} \cdot P_{E_1} = p^2 \cdot r \cdot q^2.$$  

## 2.2 Tables

First, let’s recall the definitions of some tables of an S-box.

**Definition 1.** Let $S$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. Given $\alpha_i, \beta_i \in \mathbb{F}_2^n, i \in \{1,2\}$, the difference distribution table (DDT) and the boomerang connectivity table (BCT [CHP+18]) are two-dimensional tables defined as

$$\text{DDT}(\alpha_1, \alpha_2) = \# \{x \in \mathbb{F}_2^n | S(x) \oplus S(x \oplus \alpha_1) = \alpha_2 \}.$$  

$$\text{BCT}(\alpha_1, \beta_2) = \# \{x \in \mathbb{F}_2^n | S^{-1}(S(x) \oplus \beta_2) \oplus S^{-1}(S(x \oplus \alpha_1) \oplus \beta_2) = \alpha_1 \}.$$  

For DDT, we introduce two sets as follows.

$$\mathcal{X}_{\text{DDT}}(\alpha, \beta) \triangleq \{ x \in \mathbb{F}_2^n : S(x) \oplus S(x \oplus \alpha) = \beta \},$$  

$$\mathcal{Y}_{\text{DDT}}(\alpha, \beta) \triangleq \{ x \in \mathbb{F}_2^n : x \in \mathbb{F}_2^n, S(x) \oplus S(x \oplus \alpha) = \beta \}. $$
**Definition 2** ([DR07]). Let $S$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. $S$ is called planar if and only if for all $\alpha, \beta \in \mathbb{F}_2^n$, both sets $X_{\text{DDT}}(\alpha, \beta)$, $Y_{\text{DDT}}(\alpha, \beta)$ are affine subspaces.

**Definition 3.** Let $S$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. The Upper BCT ($\text{UBCT}^1$ [WP19, DDV20]) and the Lower BCT ($\text{LBCT}^2$ [SQH19, DDV20]) are three-dimensional tables defined as

\[
\text{UBCT}(\alpha_1, \alpha_2, \beta_2) = \# \left\{ x \in \mathbb{F}_2^n \mid S(x) \oplus S(x \oplus \alpha_1) = \alpha_2, S^{-1}(S(x) \oplus \beta_2) \oplus S^{-1}(S(x \oplus \alpha_1) \oplus \beta_2) = \alpha_1 \right\},
\]

\[
= \#(X_{\text{UBCT}}(\alpha_1, \alpha_2) \cap (X_{\text{UBCT}}(\alpha_1, \alpha_2) \oplus \beta_2)),
\]

\[
\text{LBCT}(\alpha_1, \beta_1, \beta_2) = \# \left\{ x \in \mathbb{F}_2^n \mid S(x) \oplus S(x \oplus \beta_1) = \beta_2, S^{-1}(S(x) \oplus \beta_2) \oplus S^{-1}(S(x \oplus \alpha_1) \oplus \beta_2) = \alpha_1 \right\},
\]

\[
= \#(X_{\text{LBCT}}(\beta_1, \beta_2) \cap (X_{\text{LBCT}}(\beta_1, \beta_2) \oplus \alpha_1)).
\]

**Definition 4.** Let $S$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. The Extended BCT ($\text{EBCT}$ [BHL+20],

---

1$\text{UBCT}$ is called Boomerang Difference Table BDT in [WP19], renamed as $\text{UBCT}$ in [DDV20].
2$\text{LBCT}$ is denoted by $D_{\text{BDT}}$ in [SQH19], renamed as $\text{LBCT}$ in [DDV20].
et al. proposed a new approach to search for boomerang characteristics with high probability and then combine them. In [CHP17], Cid et al. proposed an MILP model for searching boomerang distinguishers on Deoxys, which employs the ladder switch in the combination. Later, Liu et al. gave a more generic MILP model for the block cipher GIFT in [LS19]. Note that in these two works the target cipher is subdivided into three parts $E = E_1 \circ E_m \circ E_0$, where $E_m$ is restricted to a single round. Then in [SQH19], Song et al. provided a new tool to compute the probability of boomerang distinguishers and showed that the dependency may exist in multiple rounds. One limitation of this work is that $E_m$ is determined when two trails are given. In [DDV20], Deaume et al. proposed a new approach to search for boomerang characteristics with $E_m$ being identified automatically.

**4** New Properties of Double Boomerang Connectivity Table

**Definition 5.** Let $S$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. Given $\alpha, \beta \in \mathbb{F}_2^n, i \in \{1, 2\}$, we define for each table the transition probability as

$$
P_{\text{DDT}}(\alpha_1, \alpha_2) = \text{DDT}(\alpha_1, \alpha_2)/2^n, \quad P_{\text{UBCT}}(\alpha_1, \alpha_2, \beta_2) = \text{UBCT}(\alpha_1, \alpha_2, \beta_2)/2^n,$$

$$P_{\text{LBCD}}(\alpha_1, \alpha_2) = \text{LBCD}(\alpha_1, \alpha_2, \beta_2)/2^n, \quad P_{\text{EBCT}}(\alpha_1, \alpha_2, \beta_2) = \text{EBCT}(\alpha_1, \alpha_2, \beta_2)/2^n.$$

**2.3 Properties of the Tables**

1. **S-box switch:** $\forall \alpha_i, \beta_i \in F_2^n, i \in \{1, 2\}$

$$\text{DDT}(\alpha_1, \alpha_2) = \text{UBCT}(\alpha_1, \alpha_2, \alpha_2) = \text{EBCT}(\alpha_1, \alpha_1, \alpha_2, \alpha_2),$$

$$\text{DDT}(\beta_1, \beta_2) = \text{LBCD}(\beta_1, \beta_2, \beta_2) = \text{EBCT}(\beta_1, \beta_1, \beta_2, \beta_2). \quad (1)$$

2. **Ladder switch:** $\forall \alpha_1, \beta_2 \in F_2^n$

$$\text{BCT}(0, \beta_2) = \text{UBCT}(0, 0, \beta_2) = 2^n, \quad \text{BCT}(\alpha_1, 0) = \text{LBCD}(\alpha_1, 0, 0) = 2^n.$$

3. $\forall \alpha_i, \beta_i \in F_2^n, i \in \{1, 2\}$

$$\text{UBCT}(\alpha_1, \alpha_2, \beta_2) = \sum_{\beta_1 \in F_2^n} \text{EBCT}(\alpha_1, \beta_1, \alpha_2, \beta_2),$$

$$\text{LBCD}(\alpha_1, \beta_1, \beta_2) = \sum_{\alpha_2 \in F_2^n} \text{EBCT}(\alpha_1, \beta_1, \alpha_2, \beta_2),$$

$$\text{BCT}(\alpha_1, \beta_2) = \sum_{\alpha_2 \in F_2^n} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) = \sum_{\beta_1 \in F_2^n} \text{LBCD}(\alpha_1, \beta_1, \beta_2)$$

$$= \sum_{\beta_1, \alpha_2 \in F_2^n} \text{EBCT}(\alpha_1, \beta_1, \alpha_2, \beta_2).$$

**2.4 Previous Methods to Search for Boomerang Distinguishers**

To find boomerang distinguishers, the classical approach is to search for two short differential characteristics with high probability and then combine them. In [CHP17], Cid et al. proposed an MILP model for searching boomerang distinguishers on Deoxys, which employs the ladder switch in the combination. Later, Liu et al. gave a more generic MILP model for the block cipher GIFT in [LS19]. Note that in these two works the target cipher is subdivided into three parts $E = E_1 \circ E_m \circ E_0$, where $E_m$ is restricted to a single round. Then in [SQH19], Song et al. provided a new tool to compute the probability of boomerang distinguishers and showed that the dependency may exist in multiple rounds. One limitation of this work is that $E_m$ is determined when two trails are given. In [DDV20], Deaume et al. proposed a new approach to search for boomerang characteristics with $E_m$ being identified automatically.

---

3In [Nyb19], a three-dimensional table, also named EBCT, was defined as

$$\text{EBCT}(a; b, c) = \# \{(x, y) \in \mathbb{F}_2^n \times \mathbb{F}_2^n | S(x) \oplus S(y) = b \text{ and } S(x \oplus a) \oplus S(y \oplus a) = c\},$$

where the output difference is allowed to be different from the difference on the opposite face.
Remark. All previous works search for boomerang distinguishers in two steps. The first step finds good truncated characteristics and the second step searches for good instantiations following the obtained truncated characteristics. The clustering effect is significant especially for word-oriented block ciphers. At present, all previous works consider the cluster effect when actual characteristics are obtained in the second step. As far as we know, no method is available in the literature to reflect the clustering effect in the first step.

3 New Properties of Double Boomerang Connectivity Table

In this section, we define the Double Boomerang Connectivity Table (DBCT) and present new properties of it. Then we discuss the extensions of DBCT.

3.1 Double Boomerang Connectivity Table (DBCT)

DBCT as defined below captures the properties of two S-boxes in a row in boomerang attacks, as depicted in Fig. 3 (left).

![Figure 3: DBCT of general S-box (left) and DBCT of hard S-box in the sense of Definition 8 (right)](image)

Definition 6. Let $S$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. The double boomerang connectivity table (DBCT)$^4$ is defined as

$$\text{DBCT}(\alpha_1, \beta_3) = \sum_{\alpha_2, \beta_2} \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3),$$

where $\text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) = \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)$.

Like the differential uniformity, a new uniformity can be defined similarly.

Definition 7 (Double Boomerang Uniformity). The double boomerang uniformity of $S$ is the largest value in the DBCT except for the first row and the first column:

$$U = \max_{\alpha_1, \beta_3 \neq 0} \text{DBCT}(\alpha_1, \beta_3).$$

Note we could represent $\text{DBCT}(\alpha_1, \beta_3)$ as the sum of two parts:

$$\text{DBCT}(\alpha_1, \beta_3) = \sum_{\alpha_2 = \beta_2} \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) + \sum_{\alpha_2 \neq \beta_2} \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3).$$

$^4$DBCT was first introduced in [HBS21] and defined in an algorithmic way. In fact, the DBCT notation used in this paper is the same as in [HBS21] but we use a more succinct definition.
For several 4-bit S-boxes (CRAFT [BLMR19], QARMA [Ava17], PRESENT [BKL+07], LBlock-s0 [WZ11], LBlock-s1 [WZ11], MIBS [ISSK09], and TWNIE [SMMK12]), we calculate their \( \text{dbct} \). When \( \alpha_1, \beta_3 \neq 0 \), there are 704, 738, 620, 608, 608, 735, and 735 nonzero values in total, respectively. Considering \( \alpha_2 \neq \beta_2 \), there are only 72, 36, 0, 0, 0, 0, and 0 nonzero values, respectively. We observe that the nonzero \( \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) \) occurs mainly when \( \alpha_2 = \beta_2 \). This means in most cases the \( UBCT \) and \( LBCT \) for computing \( DBCT \) degenerate to \( DDT \), as shown in Equation (1). Thus, the entries of \( DBCT \) can be lower-bounded by a value computed from \( DDT \) entries, as formalized in Property 1.

**Property 1.** Let \( S \) be a function from \( \mathbb{F}_2^2 \) to \( \mathbb{F}_2^2 \). For all \( \alpha_1, \alpha_2, \beta_2, \beta_3 \in \mathbb{F}_2^2 \setminus \{0\} \), nonzero \( \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) \) occurs mainly when \( \alpha_2 = \beta_2 \). Consequently,

\[
\text{DBCT}(\alpha_1, \beta_3) = \sum_{\alpha_2} \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) = \sum_{\alpha_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)
\geq \sum_{\alpha_2=\beta_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)
= \sum_{\alpha_2} \text{DDT}(\alpha_1, \alpha_2) \cdot \text{DDT}(\alpha_2, \beta_3).
\]

For \( \alpha_1 = 0 \) or \( \beta_3 = 0 \), it corresponds to the ladder-switch [DKS10] and

\[
\text{DBCT}(0, \beta_3) = \sum_{\beta_2} \text{UBCT}(0, 0, \beta_2) \cdot \text{LBCT}(0, \beta_2, \beta_3) = 2^{2n},
\]

\[
\text{DBCT}(\alpha_1, 0) = \sum_{\alpha_2} \text{UBCT}(\alpha_1, \alpha_2, 0) \cdot \text{LBCT}(\alpha_2, 0, 0) = 2^{2n}.
\]

It is also observed that for certain S-boxes, the set of nonzero \( \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) \) always satisfies \( \alpha_2 = \beta_2 \). This is an exciting property as it turns the “\( \geq \)” in Property 1 into a more desirable “\( = \)”. If such a property holds for an S-box, we call it a **hard S-box**.

**Definition (Hard S-box).** Let \( S \) be a function from \( \mathbb{F}_2^2 \) to \( \mathbb{F}_2^2 \). \( S \) is hard if the following holds. For all \( \alpha_1, \beta_3 \neq 0 \),

\[
\text{DBCT}(\alpha_1, \beta_3) = \sum_{\alpha_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)
= \sum_{\alpha_2=\beta_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)
= \sum_{\alpha_2} \text{DDT}(\alpha_1, \alpha_2) \cdot \text{DDT}(\alpha_2, \beta_3).
\]

**Remark.** For a cipher employing hard S-boxes, it only allows two typical switch effects in two continuous S-boxes, i.e., the S-box switch and the ladder switch. In other words, a right quartet \((x_1, x_2, x_3, x_4)\) for the two continuous S-box is always composed of two pairs of the same value, i.e., \( x_1 = x_4, x_2 = x_3 \), as illustrated in Fig. 3 (right).

**Example 1.** A good example of hard S-boxes is PRESENT’s S-box. Table 1 and Table 2 display PRESENT’s \( \text{DBCT} \) and \( DDT \). It can be seen that for all \( i, j \neq 0 \), the entry at position \((i, j)\) in \( \text{DBCT} \) equals the dot products between the \( i \)-th row and the \( j \)-th column of \( DDT \).
Then under what circumstances is an S-box hard? According to Definition 3, for \( \alpha_2, \beta_2 \in F_2^n \setminus 0 \), a hard S-box requires

\[
\sum_{\alpha_2 \neq \beta_2} dbct(\alpha_1, \alpha_2, \beta_2, \beta_3) = \sum_{\alpha_2 \neq \beta_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)
\]

\[
= \sum_{\alpha_2 \neq \beta_2} \#(\mathcal{Y}_{\text{DBCT}}(\alpha_1, \alpha_2) \cap (\mathcal{Y}_{\text{DBCT}}(\alpha_1, \alpha_2) \oplus \beta_2)) \cdot \#(\mathcal{X}_{\text{DBCT}}(\beta_2, \beta_3) \cap (\mathcal{X}_{\text{DBCT}}(\beta_2, \beta_3) \oplus \alpha_2)) = 0.
\]

**Proposition 1.** Let \( S \) be a planar S-box from \( F_2^n \) to \( F_2^n \). For \( \alpha_2, \beta_2 \in F_2^n \setminus 0, \alpha_2 \neq \beta_2 \), and \( \alpha_1, \beta_3 \in F_2^n \), if

\[
\begin{cases}
\text{span}(\alpha_2, \beta_2) \subseteq Y_{\text{DBCT}}(\alpha_1, \alpha_2), \\
\text{span}(\alpha_2, \beta_2) \subseteq X_{\text{DBCT}}(\beta_2, \beta_3)
\end{cases}
\]

never holds, then \( S \) is hard. If \( S \) has differential uniformity 4, then the S-box is hard when \( \text{DDT}(\alpha_1, \alpha_2) = \text{DDT}(\beta_2, \beta_3) = 4 \), \( Y_{\text{DBCT}}(\alpha_1, \alpha_2) = X_{\text{DBCT}}(\beta_2, \beta_3) \) never holds.

**Proof.** If the S-box is planar, for \( \alpha, \beta \in F_2^n \), we can write \( X_{\text{DBCT}}(\alpha, \beta) \) and \( Y_{\text{DBCT}}(\alpha, \beta) \) as

\[
X_{\text{DBCT}}(\alpha, \beta) = x_0 + X_{\text{DBCT}}(\alpha, \beta),
\]

\[
Y_{\text{DBCT}}(\alpha, \beta) = y_0 + Y_{\text{DBCT}}(\alpha, \beta),
\]

where \( X_{\text{DBCT}}(\alpha, \beta) \) and \( Y_{\text{DBCT}}(\alpha, \beta) \) are linear subspaces and \( x_0 \) and \( y_0 \) are elements of \( X_{\text{DBCT}}(\alpha, \beta) \) and \( Y_{\text{DBCT}}(\alpha, \beta) \), respectively. Consider the cardinality of the intersection \( \mathcal{X}_{\text{DBCT}}(\alpha_1, \alpha_2) \cap (\mathcal{X}_{\text{DBCT}}(\alpha_1, \alpha_2) \oplus \beta_2) \). There are two possible cases.

- \( \beta_2 \in Y_{\text{DBCT}}(\alpha_1, \alpha_2) \), which equivalently means that \( \mathcal{Y}_{\text{DBCT}}(\alpha_1, \alpha_2) = \mathcal{Y}_{\text{DBCT}}(\alpha_1, \alpha_2) \oplus \beta_2 \).
- \( \beta_2 \notin Y_{\text{DBCT}}(\alpha_1, \alpha_2) \), which means that \( \mathcal{Y}_{\text{DBCT}}(\alpha_1, \alpha_2) \oplus \beta_2 \) is a coset of \( Y_{\text{DBCT}}(\alpha_1, \alpha_2) \) different from \( Y_{\text{DBCT}}(\alpha_1, \alpha_2) \). In this case, the intersection between the two cosets is empty.

The same applies to the cardinality of the intersection \( \mathcal{X}_{\text{DBCT}}(\beta_2, \beta_3) \cap (\mathcal{X}_{\text{DBCT}}(\beta_2, \beta_3) \oplus \alpha_2) \). Then, a planar S-box is hard if

\[
\sum_{\alpha_2 \neq \beta_2} dbct(\alpha_1, \alpha_2, \beta_2, \beta_3) = \sum_{\alpha_2 \neq \beta_2, \beta_2 \in Y_{\text{DBCT}}(\alpha_1, \alpha_2), \alpha_2 \in \mathcal{X}_{\text{DBCT}}(\beta_2, \beta_3)} \#Y_{\text{DBCT}}(\alpha_1, \alpha_2) \cdot \#X_{\text{DBCT}}(\beta_2, \beta_3) = 0.
\]

**Example 2.** The S-box of CRAFT, whose DDT is displayed in Table 12, is not hard as \( Y_{\text{DBCT}}(10, 5) = X_{\text{DBCT}}(15, 10) = \{0, 5, 10, 15\} \) and \( \text{DBCT}(10, 10) = \sum_{\alpha_2 = \beta_2} dbct(10, \alpha_2, \beta_2, 10) + \sum_{\alpha_2 \neq \beta_2} dbct(10, \alpha_2, \beta_2, 10) = 64 + 64 = 128 \). The DBCT of CRAFT is displayed in Table 13.

### 3.2 Extensions

One may wonder under what circumstances the DBCT is applicable, i.e., there are two active S-boxes in a row. Basically, the linear layer of the round function needs to be simple so that the output difference of one S-box may exactly be the input difference of another S-box. Indeed, this may happen when the linear layer can be represented with a binary matrix. A natural question would be: what if the linear layer is extremely simple or complex?

In this subsection, we first discuss extensions of DBCT in the case where the linear layer is extremely simple. In this case, multiple active S-boxes in a row are possible. Then we discuss the extension in the case where the linear layer is relatively complex.
Table 1: DBCT of PRESENT’s 4-bit S-box

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>1</td>
<td>256</td>
<td>32</td>
<td>32</td>
<td>16</td>
<td>32</td>
<td>-</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>256</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>16</td>
<td>24</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>256</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>256</td>
<td>24</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>256</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>8</td>
<td>24</td>
<td>8</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>24</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>40</td>
<td>16</td>
<td>8</td>
<td>40</td>
<td>-</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>256</td>
<td>-</td>
<td>16</td>
<td>16</td>
<td>40</td>
<td>24</td>
<td>24</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>a</td>
<td>256</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>b</td>
<td>256</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>c</td>
<td>256</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td>d</td>
<td>256</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td>e</td>
<td>256</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td>f</td>
<td>256</td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>-</td>
<td>16</td>
<td>24</td>
</tr>
</tbody>
</table>

Multiple S-boxes. In the case of \( t > 2 \) consecutive S-boxes, we could define as well a similar table which is called \( t \)-BCT. If the S-box is hard, then we have the following proposition for \( t \)-BCT.

Proposition 2. Let \( S \) be a function from \( \mathbb{F}_2^n \) to \( \mathbb{F}_2^n \). If \( S \) is hard, then for \( \forall \alpha, \beta \in \mathbb{F}_2^n \setminus 0 \) and \( t > 2 \),

Table 2: DDT of PRESENT’s 4-bit S-box

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>a</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>b</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>c</td>
<td>-</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>d</td>
<td>-</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>e</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>f</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
3-BCT(\(\alpha, \beta\)) = \sum_{\alpha_2, \beta_2, \alpha_3, \beta_3} \text{UBCT}(\alpha, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3) \cdot \text{LBCT}(\alpha_3, \beta_3, \beta)

= \sum_{\alpha_2, \alpha_3} \text{DDT}(\alpha, \alpha_2) \cdot \text{DDT}(\alpha_2, \alpha_3) \cdot \text{DDT}(\alpha_3, \beta)

\(t\)-BCT(\(\alpha, \beta\)) = \sum_{\alpha_2, \beta_2, \ldots, \alpha_t, \beta_t} \text{UBCT}(\alpha, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3) \cdot \ldots \cdot \text{LBCT}(\alpha_t, \beta_t, \beta)

= \sum_{\alpha_2, \ldots, \alpha_t} \text{DDT}(\alpha, \alpha_2) \cdot \text{DDT}(\alpha_2, \alpha_3) \cdot \ldots \cdot \text{DDT}(\alpha_t, \beta).

The proof of Proposition 2 is postponed to Appendix A.1.

**Complex linear layer.** Suppose the the output differences of the first S-box is \(\alpha_2, \beta_2\) and input differences of the second S-box is \(\alpha'_2, \beta'_2\), as depicted in Fig. 4. Unlike the case \(\alpha_2 = \alpha'_2, \beta_2 = \beta'_2\) captured by the original DBCT, there is a complex linear mapping \(M\) between \(\alpha_2, \beta_2\) and \(\alpha'_2, \beta'_2\). As long as \(\alpha'_2\) (resp. \(\beta_2\)) can be computed from \(\alpha_2\) (resp. \(\beta'_2\)) through \(M\), a DBCT can be defined similarly as follows.

\[\text{DBCT}(\alpha_1, \beta_3) = \sum_{\alpha_2, \beta_2} \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) = \sum_{\alpha_2, \beta_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha'_2, \beta'_2, \beta_3).\]

Figure 4: General DBCT with a complex linear layer in between

**Example 3.** Consider four bytes passing through \(\text{SB} \circ \text{MC} \circ \text{SB}\) where \(\text{SB}\) and \(\text{MC}\) are the same as defined in AES. Without loss of generality, we consider \(\text{DBCT}^{0,j}(0 \leq j \leq 3)\) for the first input byte and the \(j\)-th output byte. Let’s assume that the input difference state for upper characteristic is \(\Delta = [\alpha_1, 0, 0, 0]^T\) and consider four situations for the output difference state for lower characteristic as shown in Fig. 5. For case 1 where \(\nabla = [\beta_3, 0, 0, 0]^T\), for \(\forall \alpha_1, \beta_3 \in \mathbb{F}_{2^8}\):

\[\alpha'_2 = (\text{MC} \cdot [\alpha_2, 0, 0, 0]^T)[0], \beta_2 = (\text{MC}^{-1} \cdot [\beta'_2, 0, 0, 0]^T)[0]\]

\[\text{DBCT}^{0,0}(\alpha_1, \beta_3) = \sum_{\alpha_2, \beta_2} \text{dbct}(\alpha_1, \alpha_2, \beta_2, \beta_3) = \sum_{\alpha_2, \beta_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha'_2, \beta'_2, \beta_3).\]

The AES S-box is an 8-bit S-box, and thus the size of its DBCT is \(256 \times 256\). In the \(\text{DBCT}^{0,0}\) of the AES S-box, all entries for zero input difference (the first row) and zero output difference (the first column) are 65536 owing to the ladder switch effect. For the other
New Properties of Double Boomerang Connectivity Table

Figure 5: Four cases of DBCT of AES block cipher: DBCT^{0,0}, DBCT^{0,1}, DBCT^{0,2} and DBCT^{0,3}

Table 3: Number of entries for each value for the DBCT^{i,j} and the basic DBCT for the AES S-box

<table>
<thead>
<tr>
<th>M</th>
<th>Table</th>
<th>65536</th>
<th>16</th>
<th>8</th>
<th>0</th>
<th>192-332</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>DBCT^{0,0}, DBCT^{1,0}, DBCT^{2,0}, DBCT^{3,0}</td>
<td>511</td>
<td>8</td>
<td>882</td>
<td>64135</td>
<td>-</td>
</tr>
<tr>
<td>MC</td>
<td>DBCT^{0,1}, DBCT^{1,1}, DBCT^{2,1}, DBCT^{3,1}</td>
<td>511</td>
<td>3</td>
<td>252</td>
<td>64770</td>
<td>-</td>
</tr>
<tr>
<td>MC</td>
<td>DBCT^{0,2}, DBCT^{1,2}, DBCT^{2,2}, DBCT^{3,2}</td>
<td>511</td>
<td>1</td>
<td>-</td>
<td>65024</td>
<td>-</td>
</tr>
<tr>
<td>MC</td>
<td>DBCT^{0,3}, DBCT^{1,3}, DBCT^{2,3}, DBCT^{3,3}</td>
<td>511</td>
<td>3</td>
<td>126</td>
<td>64896</td>
<td>-</td>
</tr>
<tr>
<td>XOR</td>
<td>basic DBCT</td>
<td>511</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>65025</td>
</tr>
</tbody>
</table>

entries, the maximum value is 16. The number of entries with 65536, 16, 8, and 0 are 511, 8, 882, and 64135. For the all 16 DBCT^{i,j} (0 \leq i, j \leq 3), there are four cases of the number of entries, the details are shown in Table 3. We also list the basic DBCT for the AES S-box in Table 3. Note that, the DBCT is related to the linear layer and the S-box. For the basic DBCT with simple XOR operations, the AES S-box is hard without zero entries. For DBCT^{i,j} (0 \leq i, j \leq 3) with complex linear layer, most entries are zero.

4 Revisiting Boomerang Attacks on CRAFT, TweAES and Deoxys-BC

In this section, we revisit some existing boomerang distinguishers of CRAFT, TweAES and Deoxys-BC, respectively. Through the boomerang distinguisher of CRAFT, we demonstrate how DBCT uniformity and hard S-box matter. For the boomerang distinguishers of TweAES and Deoxys-BC, inspired by the property of AES S-boxes with a complex linear layer in between, we verify that two published boomerang distinguishers are flawed using extended DBCT.

4.1 Effect of Different S-boxes for Boomerang Distinguishers

CRAFT is a lightweight tweakable block cipher introduced by Beierle et al. [BLMR19] at FSE 2019. For more details of the cipher, please refer to Appendix B.1 and [BLMR19]. In [HBS21], a 13-round boomerang distinguisher of CRAFT is presented and there are 7 rounds in the middle to handle the dependency. We redraw the 7-round middle part \( E_m \) in Fig. 6, where White cells are zero differences, Yellow cells are nonzero differences and Green cells are unknown differences. The input difference of the upper characteristics is \( \Delta = [0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \) and the output difference of the lower characteristics is \( \nabla = [0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \). Note that \( \{A, ..., H\} \) are the states for upper characteristics and \( \{a, ..., h\} \) are the states rounds for lower characteristics. The symbols follow those in the original work, details can be referred to [HBS21]. A detailed analysis in [HBS21] showed that the boomerang distinguisher of the 7-round \( E_m \) involves four DBCTs and its probability is \( 2^{-10.39} \).
For a boomerang distinguisher, its probability is closely related to the S-box being used. Specifically, the probability of the 7-round distinguisher is related to the DBCT, BCT and DDT of the S-box. To demonstrate the effect of the S-box on the probability of the boomerang distinguisher, we replace CRAFT’s S-box with different S-boxes and then compute the probability under all possible values for the active cells in the input and output differences ($\Delta$, $\nabla$). The results are summarized in Table 4.

Table 4: Probability of the 7-round distinguisher with different S-boxes

<table>
<thead>
<tr>
<th>S-box</th>
<th>DDT uni.</th>
<th>BCT uni.</th>
<th>DBCT uni.</th>
<th>Hard</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Max</td>
</tr>
<tr>
<td>CRAFT [BLMR19]</td>
<td>4</td>
<td>16</td>
<td>128</td>
<td>$\times$</td>
<td>$2^{-10.39}$</td>
</tr>
<tr>
<td>QARMA [Ava17]</td>
<td>4</td>
<td>10</td>
<td>48</td>
<td>$\times$</td>
<td>$2^{-13.99}$</td>
</tr>
<tr>
<td>PRESENT [BKL’07]</td>
<td>4</td>
<td>16</td>
<td>40</td>
<td>$\checkmark$</td>
<td>$2^{-15.47}$</td>
</tr>
<tr>
<td>LBlock-s0 [WZ11]</td>
<td>4</td>
<td>16</td>
<td>40</td>
<td>$\checkmark$</td>
<td>$2^{-15.61}$</td>
</tr>
<tr>
<td>LBlock-s1 [WZ11]</td>
<td>4</td>
<td>16</td>
<td>40</td>
<td>$\checkmark$</td>
<td>$2^{-15.41}$</td>
</tr>
<tr>
<td>MIBS [ISSK09]</td>
<td>4</td>
<td>6</td>
<td>32</td>
<td>$\checkmark$</td>
<td>$2^{-15.59}$</td>
</tr>
<tr>
<td>TWNIE [SMMK12]</td>
<td>4</td>
<td>6</td>
<td>28</td>
<td>$\checkmark$</td>
<td>$2^{-15.58}$</td>
</tr>
</tbody>
</table>

Table 4 compares the uniformity of DDT, BCT and DBCT for different S-boxes, lists whether these S-boxes are hard or not, and gives the probabilities for the 7-round distinguisher under different S-boxes. From Table 4, We have the following observations.

- Even though CRAFT’s S-box and PRESENT’s S-box share the same DDT and BCT uniformity, the probability of the 7-round distinguisher differs for these two S-boxes. A possible reason for this is that they have different DBCT uniformity.

- Although QARMA’s S-box has better BCT uniformity than PRESENT’s S-box, it results in a higher probability. We note QARMA’s S-box has a higher DBCT uniformity.

- PRESENT’s S-box, LBlock-s0 and LBlock-s1 share the same DDT, BCT, and DBCT uniformity; they also lead to almost the same probability for the 7-round distinguisher.
The S-boxes of MIBS and TWINE have small BCT and small DBCT uniformity; at the same time, the probabilities of the 7-round distinguisher are low and stable for different input and output differences ($\Delta, \nabla$).

These observations indicate that, apart from the uniformity of BCT and DDT, the uniformity of DBCT is a new measure criterion to evaluate the performance of S-box for resisting boomerang attacks. Therefore, the DBCT uniformity should be used together with the BCT uniformity to have a better evaluation of the S-box against the boomerang attack.

As the DBCT is equivalent to the S-box switch in most cases, i.e., a quartet is formed by two pairs of the same value, it is interesting to see what happens when we force the ciphertexts to form such quartets. It is expected that the probability will increase. An experiment on the 7-round boomerang distinguisher of CRAFT confirm this, as shown in Appendix C.1.

4.2 Flawed Boomerang Distinguisher of TweAES and Deoxys-BC

For the S-box of AES, the DBCT with complexity linear layer has too many zero values. It will easily invalidate boomerang characteristics. Inspired by this, we focus on TweAES and Deoxys-BC and revisit some boomerang distinguishers, finding them flawed.

4.2.1 Recompute the Probability of the Boomerang Distinguisher in [CDJ + 20] with DBCT

The tweakable block cipher TweAES is one of the underlying primitives of Authenticated Encryption with Associated Data (AEAD) scheme ESTATE [CDJ + 20], which is a second-round candidate of the NIST Lightweight Cryptography Standardization project. For more details of the cipher, please refer to Appendix B.2 and [CDJ + 20].

In [CDJ + 20], Chakraborti et al. introduced a 7-round boomerang distinguisher with probability $2^{-123}$, as illustrated in Fig 7. It can be seen that only the third, the fourth and the last rounds are critical to the probability of the distinguisher. Since the differential propagation of the last round is simple, we mainly detail the first six rounds and Table 11 gives their setting. We recompute the probability of the two middle rounds and find it 0 rather than $2^{-75}$ as reported in [CDJ + 20]. This means the probability of the full distinguisher is not $2^{-75}$ multiplied by the probability of the last round $2^{-48}$.

To compute the probability $P_{E_m}$ of the two middle rounds, we divide the state into four columns and calculate them separately. So

$$P_{E_m} = P_0 \cdot P_1 \cdot P_2 \cdot P_3.$$  

Without loss of generality, we compute the probability $P_1$ of the second column in detail. As shown in Fig. 7, $[*, *, \alpha'_1, \alpha'_2]^T = MC \times [\alpha, 0, 0, 0]^T$ and $[\beta, *, *, s]^T = MC^{-1} \times [0, 0, \beta'_1, \beta'_2]^T$,

$$P_1 = \frac{1}{2^{8 \cdot 3}} \cdot \sum_{\alpha, \beta_1', \beta_2' \neq 0} UBCT(1, \alpha, \beta) \cdot LBCT(\alpha'_1, \beta'_1, 4) \cdot LBCT(\alpha'_2, \beta'_2, 6).$$

It is easy to compute $P_1$ by trying all possible $\alpha, \beta'_1$, and $\beta'_2$. We then obtain $P_1 = 0$. Hence, the differential propagation for the two middle rounds is impossible, i.e.,

$$P_{E_m}(\Delta_1 \rightarrow \nabla_7) = P_0 \cdot P_1 \cdot P_2 \cdot P_3 = 0.$$  

Our computation shows that the probability of the 7-round boomerang distinguisher proposed in [CDJ + 20] is not correct.
4.2.2 Recompute the Probability of the Boomerang Distinguishers of Deoxys-BC in [BL22] with DBCT

Deoxys-BC is an AES-based tweakable block cipher [JNPS16], based on the tweakey framework [JNP14]. For more details of the cipher, please refer to Appendix B.3 and [JNP14].

In [BL22], Bariant et al. proposed some boomerang attacks on Deoxys-BC, as illustrated in Fig. 14 and Fig. 15 which are taken from [BL22]. Fig. 14 is an 8-round boomerang attack on Deoxys-BC in the model RTK1 and Fig. 15 is a 10-round boomerang attack on Deoxys-BC in the model RTK2, where RTK1 denotes single-key attacks on any variant with at least 128 bits of tweak and RTK2 denotes single-key attacks on Deoxys-BC-384 with 256 bits of tweak, or related-key attacks on Deoxys-BC-256. For more details of the attack, please refer to [BL22]. We recompute the probability for the middle part of the cipher in the two boomerang attacks and find it 0 rather than high probabilities.

8-round boomerang attack in the model RTK1. We compute the probability for the differential transition over the red boxes in the three middle rounds, as illustrated in Fig. 8. The detailed formula for computing the probability is

\[ P = \frac{1}{2^{38}} \cdot \sum_{\alpha_1, \beta_1, \alpha_2, \beta_2} \text{UBCT}(01, \alpha_1, \beta_1) \cdot \text{EBCT}(\alpha'_1, \beta'_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha'_2, \beta'_2, c8), \]

where \([*, \alpha'_1, *, *] = MC \times [0, \alpha_1, 0, 0], [*, \alpha'_2, *, *] = MC \times [0, \alpha_2, 0, 0], [*, \beta_2, *, *] = MC^{-1} \times [0, \beta'_2, 0, 0] \) and \([*, \beta_1, *, *] = MC^{-1} \times [0, \beta'_1, 0, 0] \). We verify that no matter what \( \beta_2 \) is, the differential transition from 01 to \( \beta_2 \) over two S-box layers is incompatible as \( \forall \beta_2 \in \mathbb{F}_2^8 \setminus 0 \)

\[ P' = \frac{1}{2^{28}} \cdot \sum_{\alpha_1, \beta'_1} \text{UBCT}(01, \alpha_1, \beta_1) \cdot \text{LBCT}(\alpha'_1, \beta'_1, \beta_2) = 0. \]
So the probability of $P$ must be 0, which means the characteristic over the three middle rounds is incompatible.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example8.png}
\caption{The three middle rounds of the 8-round boomerang attack in the model RTK1.}
\end{figure}

10-round boomerang attack in the model RTK2. We compute the probability for the differential transition over the red boxes in the two middle rounds, as depicted in Fig. 9. For $\forall \alpha \in \mathbb{F}_2^8 \setminus 0$, the probability is

\begin{equation}
P = \frac{1}{2^{2s}} \sum_{\alpha_1, \beta_1'} \text{UBCT}(\alpha, \alpha_1, \beta_1') \cdot \text{LBCT}(\alpha_1', \beta_1', 7f) = 0,
\end{equation}

where $[*,*] = \text{MC} \times [0, 0, 0, \alpha_1]$ and $[*,*] = \text{MC}^{-1} \times [0, 0, \beta_1', 0]$. By trying all possible $\alpha, \alpha_1$ and $\beta_1'$, we get a zero probability. Therefore, the middle part of the 10-round attack is also incompatible.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example9.png}
\caption{The two middle rounds of the 10-round boomerang attack in the model RTK2.}
\end{figure}

4.2.3 Discussion

Even though the basic DBCT cannot be directly applied to the boomerang distinguisher of TweAES and Deoxys–BC, employing the extensions, we do confirm that the interactions between two S-box layers matter and should be treated carefully. The source codes for computing the probabilities in subsection 4.2 are available via the link https://www.jianguoyun.com/p/DTV20E4QiPTMChiVlNQEIAA.
5 MILP Model to Search for Boomerangs with Cluster Probability

It is shown in [SQH19] the probability of a boomerang distinguisher of SKINNY is significantly increased from $2^{-103.84}$ to $2^{-77.83}$ when the clustering effect is considered. Later, better boomerang distinguishers of SKINNY were proposed by exploiting the clustering effect in [DDV20, HBS21]. Generally, the search for boomerang distinguishers proceeds in two steps. The first one is to search for good truncated boomerang characteristics with the least active S-boxes, and the second one is to search for the best instantiations. Although the cluster effect is very significant for word-oriented block ciphers, it is hard to be well considered in the above two steps. In fact, in the previous works [DDV20, HBS21] multiple boomerang characteristics are counted only when a good boomerang characteristic is given. In other words, multiple characteristics are searched under the fixed input and the output difference of a given boomerang characteristic. Due to the limitation of computing and storage capacity, there is no guarantee that the search will lead to boomerang clusters with sufficiently good probability.

To partially overcome the drawbacks, we propose a new strategy to search for boomerang distinguishers. Note that, for a boomerang distinguisher, only the input difference of the upper characteristic and the output difference of the lower characteristics are fixed while the difference of the intermediate state can vary. This motivates us to borrow the methods for calculating the probability of truncated differentials and provide a formula for estimating the probability of a boomerang cluster. In particular, Property 1 is used to simplify the computation of the probability of the middle part $E_m$. In this section, this formula is presented by taking a boomerang distinguisher of the block cipher CRAFT as an example. With this formula, we then propose a new MILP model to search for truncated characteristics with good cluster probability as the objective. The efficiency of the formula and the new model is demonstrated by its application to CRAFT, where better 9-round and 10-round boomerang distinguishers are obtained.

5.1 Formula for the Probability of Boomerang Clusters

The existing strategy for searching for good boomerang distinguishers is to search for a single boomerang characteristic with minimal active S-boxes as the objective at first. Our basic idea is that if we could replace the objective function with the cluster probability, it is more likely to obtain good boomerang clusters.

In the following, we formulate the boomerang cluster probability for SPN ciphers via an example of CRAFT under a common assumption used in truncated differential cryptanalysis and then show how to model the probability of clusters with MILP. Note that we consider SPN ciphers with $n$ parallel S-boxes of $s$ bits each in the nonlinear layer.

5.1.1 The previous formula for the probability of boomerang clusters

Suppose we have a boomerang distinguisher of $E = E_1 \circ E_m \circ E_0$. Following the work of Song et al. from [SQH19] and Delaune from [DDV20], the probability of the distinguisher is

$$P_E(\Delta \iff \nabla) = \sum_{\Delta_1, \nabla_1} P_{E_0}(\Delta \iff \Delta_1) \cdot P_{E_m}(\Delta_1 \iff \nabla_1) \cdot P_{E_1}(\nabla_1 \iff \nabla).$$

We assume $E_m$ contains dependency, i.e., the differential probability of its active S-boxes cannot be evaluated only by DDT. We denote two DDTs by UDDT2 and LDDT2, where L and U denote whether the two DDTs belong to lower path or upper path, and the probability transition formulas are $P_{UDDT2}(\alpha_1, \alpha_2) = (P_{DDT}(\alpha_1, \alpha_2))^2$ and $P_{LDDT2}(\beta_1, \beta_2) = (P_{DDT}(\beta_1, \beta_2))^2$ where $(\alpha_1, \alpha_2)$ is the input difference and output difference of the S-box in upper path and $(\beta_1, \beta_2)$ is the input difference and output difference of the S-box in lower path.
Definition 9 (Upper Boundary/Lower Boundary of $E_m$ [SQH19]). The upper boundary ($B^u$) of $E_m$ is delineated by the round, which only has $UDDT2$. The lower boundary ($B^l$) of $E_m$ is delineated by the round which only has $LDDT2$.

Taking CRAFT as an example, we give the previous formula for the probability of boomerang clusters.

Example 4. Fig. 10 shows a 13-round boomerang characteristic from [HBS21], which is obtained by extending the 7-round boomerang distinguisher as in Fig. 6, by three rounds on both sides. The input difference of the upper characteristics is $\Delta = [0, 0, a, a, 0, 0, 0, a, a, 0, 0, 0, 0, a]$ and the output difference of the lower characteristics is $\nabla = [0, a, 0, 0, 0, 0, 0, 0, a, a, 0, 0, 0, 0, a]$. The symbols follow those in the original work, details can be referred to [HBS21].

- $E_0/E_1$: There are 3 rounds for $E_0$ and $E_1$, respectively. The probability is

$$P_{E_0}(\Delta \rightsquigarrow \Delta_1) = (P_{E_0}(\Delta \rightarrow \Delta_1))^2 = p^2,$$

$$P_{E_1}(\nabla_1 \rightsquigarrow \nabla) = (P_{E_1}(\nabla_1 \leftarrow \nabla))^2 = q^2,$$

where $\Delta_1 = [0, 0, 0, 0, 0, A_5, 0, 0, 0, 0, 0, 0, 0, 0, 0]$, $\nabla_1 = [0, 0, 0, 0, 0, 0, h_5, 0, 0, 0, 0, 0, 0, 0, 0]$. Given $A_5 \neq 0$, $h_5 \neq 0$, the probability $p$ and $q$ can be computed as follows

$$p = \sum_{x_{12}, y_9} (P_{\text{DBCT}}(a, x_{12}))^3 \cdot P_{\text{DBCT}}(x_{12}, y_9) \cdot P_{\text{DBCT}}(y_9, A_5) \cdot Pr(a \xrightarrow{\text{DDT}^2} y_9),$$

$$q = \sum_{z_9, w_{12}} P_{\text{DBCT}}(h_5, z_9) \cdot P_{\text{DBCT}}(z_9, w_{12}) \cdot (P_{\text{DBCT}}(w_{12}, a))^3 \cdot Pr(z_9 \xleftarrow{\text{DDT}^2} a),$$

where $x_{12}, y_9, z_9, w_{12}$ are intermediate differences.

- $E_m$: There are 7 rounds, consisting of 4 DBCTs, the probability is $P_{E_m}(A_5 \rightsquigarrow h_5) = r$,

$$r = \sum_{B_9, b_9, c_5, C_{12}, c_{12}, d_1, E_1', F_1, f_{12}, G_9, g_9} P_{\text{DBCT}}(A_5, B_9, b_9) \cdot P_{\text{DBCT}}(B_9, b_9, c_5) \cdot P_{\text{DBCT}}(B_9, C_{12}, c_{12}) \cdot P_{\text{DBCT}}(C_{12}, c_{12}, d_1)
\cdot Pr(d_1 \xrightarrow{\text{DDT}^2} f_{12}) \cdot Pr(c_5 \xrightarrow{\text{DDT}^2} f_{12}) \cdot Pr(C_{12} \xrightarrow{\text{DDT}^2} E_1') \cdot Pr(C_{12} \xrightarrow{\text{DDT}^2} F_1').$$

$$P_{\text{DBCT}}(E_1', F_1, f_{12}) \cdot P_{\text{DBCT}}(F_{12}, f_{12}, g_9) \cdot P_{\text{DBCT}}(F_9', G_9, g_9) \cdot P_{\text{DBCT}}(G_9, g_9, h_5).$$

Figure 10: Example: a boomerang distinguisher for 13 rounds of CRAFT.
Considering the clusters, the final formula of the probability for $E$ is

$$P_r = \sum_{\Delta_1, \nabla_1 \neq 0} p^2 \cdot q^2 \cdot r.$$

### 5.1.2 Our new formula for the probability of boomerang clusters

Next, taking CRAFT as an example, we propose a high-level procedure to generate a formula that approximates the probability of its best boomerang cluster without focusing on a single characteristic.

In [MSAK99], Moriai et al. proposed a method to calculate the truncated differential probability for word-oriented SPN block ciphers. Typically, for two $s$-bit cells $a$ and $b$ which are independently uniformly distributed on $\mathbb{F}_{2^s} \setminus \{0\}$, the probability distribution of $a \oplus b$ is:

$$\begin{align*}
\frac{1}{2^s - 1} & \quad a \oplus b = 0, \\
\frac{1}{2^s - 1} & \quad a \oplus b \neq 0.
\end{align*}$$

In the boomerang clusters, the intermediate differences can take many possible values as in the truncated differentials, so the above probability distribution also applies here. For $E_0$ (resp. $E_1$), only UDDT2 (resp. LDDT2) is used for computing the probability. Actually, the probability for $E_0$ (resp. $E_1$) is equivalent to the differential probability given fixed input (resp. output) difference. Thus we could directly compute the probability using the method common to the one used in the truncated differential analysis.

1. Inspired by the idea of truncated differential, we transform the computation of $p$ and $q$ into counting the equality conditions of XOR operations by $\hat{p}$ and $\hat{q}$. As shown on the left of Fig. 11, there are 3 cells need to be 0 and the last cell need to be the fixed value, thus the probability is $\hat{p} = \frac{1}{15} = 2^{-15.63}$ on average if each difference distributed uniformly on $\mathbb{F}_{2^s} \setminus \{0\}$. Given a specific $\Delta$ as above, the probability $\hat{p}$ can be further adjusted to $\hat{p} = \frac{1}{15} = 2^{-11.81}$ on average by taking into account $\text{DDT}(0xa, \ast) = 4$. Similarly, the probability $\hat{q}$ is $2^{-11.81}$ on average, for $\forall h_5 \neq 0$.

2. Computing $r$ is complex. For example, in Equation 2, there are 12 variables to be traversed, which is very computationally intensive. We try to convert $r$ to the case containing only DDT, further simplifying the evaluation by the idea of truncated differentials. We obtain its lower bound by simplifying the computation using Property 1,

$$\hat{r} = \sum_{B_9, C_{12}, f_{12}, g_9} P_{\text{DDT}}(A_5, B_9) \cdot \text{Pr}(B_9 \xrightarrow{4 \text{ DDT}} f_{12}) \cdot \text{Pr}(B_9, C_{12}) \cdot \text{Pr}(C_{12} \xrightarrow{3 \text{ DDT}} f_{12}) \cdot$$

$$\text{Pr}(C_{12} \xrightarrow{3 \text{ DDT}} f_{12}) \cdot P_{\text{DDT}}(f_{12}, g_9) \cdot \text{Pr}(C_{12} \xrightarrow{4 \text{ DDT}} g_9) \cdot P_{\text{DDT}}(g_9, h_5)$$

where $B_9 = b_9, C_{12} = c_{12}$. Because of the nature of DBCT, the formula is simplified to the model only with DDT, and the probability can be calculated by using the technique of truncated differential evaluation. As shown in the middle of Fig. 11, there are 2 cell-wise conditions consumed in $f_{12}$, 1 cell condition consumed in $g_9$, and 1 cell condition consumed in $h_5$.Essentially because there are 4 UBCST · LBCTs, 4 connections are established and 4 cells condition consumed are created. Therefore, the probability of $E_m$ is $\hat{r} = \frac{1}{15} = 2^{-15.63}$ on average for any $A_5, h_5 \neq 0 \in \mathbb{F}_{2^s}$.

3. With the above techniques, the calculation of the probability of $E$ will become very simple:

$$\hat{P}_r = \sum_{\Delta_1, \nabla_1 \neq 0} \hat{p}^2 \cdot \hat{q}^2 \cdot \hat{r} = 15^2 \cdot 2^{-11.81 + 2} \cdot 2^{-11.81 + 2} \cdot 2^{-15.63} = 2^{-55.06}.$$
It can be inferred from DBCT and the borrowed technique from truncated differential analysis, that the stronger the S-box is, the better our computation approximates. We then replace CRAFT’s S-box with other S-boxes and then compute the probability under all possible input and output differences \((\Delta, \nabla)\). The results are summarized in Table 5. It can be seen that the probability by our formula is closer to the actual optimal probability when a stronger S-box is used.

Table 5: The probability of 13-round boomerang distinguishers for different S-boxes

<table>
<thead>
<tr>
<th>S-box</th>
<th>The maximum probability by formula (P_r)</th>
<th>Our result by formula (\hat{P}_r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRAFT</td>
<td>(2^{-52.59})</td>
<td>(2^{-58.59})</td>
</tr>
<tr>
<td>QARMA</td>
<td>(2^{-52.79})</td>
<td>(2^{-58.89})</td>
</tr>
<tr>
<td>PRESENT</td>
<td>(2^{-52.99})</td>
<td>(2^{-55.06})</td>
</tr>
<tr>
<td>TWNIE</td>
<td>(2^{-58.49})</td>
<td>(2^{-59.84})</td>
</tr>
<tr>
<td>MIBS</td>
<td>(2^{-58.44})</td>
<td>(2^{-59.84})</td>
</tr>
</tbody>
</table>

Remark 1. Note that our formula is valid only if the characteristics are the same in both faces at \(E_m\) of the boomerang. Actually, the two faces of the boomerang could have completely different differential characteristics. Because we study the property of DBCT for the same in both faces, we do not take this into account.

Now we will give the general formula for estimating the probability of the best boomerang cluster under certain active patterns.

**Probability in \(E_0/E_1\).** Suppose \(E_0\) covers the first \(r_0\) rounds, \(E_1\) consists of the last \(r_1\) rounds. For \(\forall \Delta, \Delta_1, \nabla_1, \nabla \neq 0\), the probability are \(\Pr_{E_0}(\Delta \rightleftharpoons \Delta_1) = \hat{p}\) and \(\Pr_{E_1}(\nabla_1 \rightleftharpoons \nabla) = \hat{q}\) on average, i.e.,

\[
\hat{p} = 2^{-s-c_0} \cdot \frac{1}{|\Delta_1|},
\]

\[
\hat{q} = 2^{-s-c_1} \cdot \frac{1}{|\nabla_1|},
\]

where \(c_0\) and \(c_1\) are the number of cells which need to be zero from uniformity and \(s\) is the cell size. For the sake of simplicity, we substitute \(2^{-s}\) for \(\frac{1}{2^{s-1}}\).

**Probability in \(E_m\).** Suppose \(E_m\) is composed of the middle \(r_m\) rounds. For \(\forall \Delta_1, \nabla_1 \neq 0\) the probability is \(\Pr_{E_m}(\Delta_1 \rightleftharpoons \nabla_1) = \hat{r}\) on average and

\[
\hat{r} = 2^{-s-c_m},
\]

where \(c_m\) is the condition consumed in \(E_m\). Usually, the characteristic of \(E_m\) is complex and \(c_m\) could not be determined easily. We make simplifications using Property 1 and then directly evaluate the probability by counting the number of conditions consumed. In the following, we discuss the computation of \(c_m\) in different situations.
• $\odot \rightarrow \square$. As with $E_0$ and $E_1$, the difference of $E_m$ is not constrained to propagate with probability 1. When there is one cell needed to be zero (White) from uniformity (Green) in the upper or lower path, the probability has to be multiplied by $2^{-s}$. Equivalently, it consumes 1 condition.

• UDDT2 and LDDT2: UDDT2 is independent of the lower path. When we process an S-box where UDDT2 applies, the probability has to be multiplied by $\sum_{\beta \in \mathbb{F}_2^s} \mathbb{P}_{\text{DDT}}(\alpha, \beta) 2^{-s}$ for $\forall \alpha \in \mathbb{F}_2^n$. Equally, there is 1 cell condition consumption for one UDDT2. Similarly, LDDT2 is independent of the upper path. When there is a LDDT2, it consumes 1 condition.

• UBCT·EBCT$_m$·LBCT: While EBCT may exist or not, UBCT and LBCT must appear in pairs (otherwise, it will degenerate into BCT). Due to Property 1 and Definition 8, for UBCT·EBCT$_m$·LBCT, $m \geq 0$ the effect is almost an S-box switch. Thus to satisfy an S-box switch, the probability is $2^{-s}$ on average. It is the same as the truncated differential processing technique and equivalent to 1 condition consumption for one pair UBCT and LBCT.

• BCT: Similar to UBCT, when we add one BCT, building a BCT table from $\alpha$ to $\beta$, the probability has to be multiplied by $\mathbb{P}_{\text{BCT}}(\alpha, \beta) > \mathbb{P}_{\text{DDT}}(\alpha, \beta) > 2^{-s}$. Thus, it consumes 1 condition.

Thus, the condition consumed in $E_m$ is the sum of the number of cells which need to be zero from uniformity, the number of UDDT2 and LDDT2, the number of UBCT·EBCT$_m$·LBCT and the number of BCT.

**Probability in $E$.** The probability of a boomerang distinguisher of $E$ is:

$$\mathbb{P}_E(\Delta \leftrightarrow \nabla) = \sum_{\Delta_1, \nabla_1 \neq 0} \hat{p}^2 \cdot \hat{q}^2 \cdot \hat{r} = 2^{-2s} c_0 - 2s c_1 - s c_m - s c'_0 - s c'_1,$$

where $c_0'$ is the number of UDDT2 in the upper boundary round and $c_1'$ is the number of LDDT2 in the lower boundary round.

**Remark 2.** In the above formula, all tables consume the same number of conditions. However, given an exact S-box, the consumption of condition for different tables may differ from 1 and a proper coefficient can be used to have a more accurate estimation.

### 5.2 MILP Mode to Search for Boomerangs with Good Cluster Probabilities

In this subsection, we give our MILP model for searching boomerangs, which takes the number of conditions consumed as the objective function. This model can be used alone to obtain good truncated boomerangs. Particularly, good boomerang clusters can be found if we instantiate the input and output differences for the obtained truncated ones.

**Notions.** We consider $E$, a classical SPN cipher with the round function composed of cell-level operations. Let $E$ be a cipher with $N_r$ rounds and $n$ cells state.

1. We use two bit variables to encode whether the difference of a cell will be free or controlled and whether its difference value will be known or unknown. A free difference can take any (nonzero) value uniformly while a controlled difference can not. Notably, a White cell is controlled, a Green cell is free and a Yellow cell is indeterminate. More specifically,
Modeling of the Attribute Propagation through SubBytes(S-RULE). The SubBytes operation does not change the activeness of a cell, but would change its difference from free to controlled, \( i.e., (0, 0) \rightarrow S(0, 0), (0, 1) \rightarrow S(0, 1)/(1, 0), (1, 0) \rightarrow S(1, 0), (1, 1) \rightarrow S(1, 1) \).

For the modeling point set, we use the convex hull computation method [SHW+14] to generate the set of inequalities.

Modeling of the Attribute Propagation through XOR Operation with the Condition Consuming(XOR-RULE).

- a White cell XOR-ed with a cell of any attribute results in the cell of the same attribute \( \mathbb{1} \), \( i.e., \mathbb{1} + \mathbb{1} \rightarrow \mathbb{1} + \mathbb{1} \).
- a Green cell XOR-ed with a cell of any attribute results in a Green cell, \( i.e., \mathbb{1} + \mathbb{1} \rightarrow \mathbb{1} \).
- a couple of Yellow cells results in a White cell with 1-cell condition consuming or a Green cell, \( i.e., \mathbb{1} + \mathbb{1} \rightarrow \mathbb{1} \).

<table>
<thead>
<tr>
<th>S-RULE</th>
<th>LDDT</th>
<th>UDDT</th>
<th>BCT</th>
<th>UBC</th>
<th>EBCT</th>
<th>LBCT</th>
<th>UDDT2</th>
<th>LDDT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0) \oplus (0, 1) \rightarrow (0, 0))</td>
<td>(0, 0) \oplus (0, 1) \rightarrow (0, 1))</td>
<td>(0, 0) \oplus (0, 1) \rightarrow (1, 0))</td>
<td>(0, 0) \oplus (0, 1) \rightarrow (1, 1))</td>
<td>(0, 0) \oplus (0, 1) \rightarrow (1, 1))</td>
<td>(0, 0) \oplus (0, 1) \rightarrow (1, 1))</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0, 1) \oplus (0, 1) \rightarrow (1, 1))</td>
<td>(0, 1) \oplus (0, 1) \rightarrow (0, 1))</td>
<td>(0, 1) \oplus (0, 1) \rightarrow (1, 0))</td>
<td>(0, 1) \oplus (0, 1) \rightarrow (1, 1))</td>
<td>(0, 1) \oplus (0, 1) \rightarrow (1, 1))</td>
<td>(0, 1) \oplus (0, 1) \rightarrow (1, 1))</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A free difference can take any (nonzero) value uniformly while a controlled difference can not. \(0, 1) \oplus (0, 1) = (1, 1)\) is (1, 1) and not (0, 1) because in boomerang attacks, the difference value after XOR is related to at least two input operands and is closer to the random case, so it is free. So we use \(0, 1) \oplus (0, 1) = (0, 0)\) with one cell condition, or \(0, 1) \oplus (0, 1) = (1, 1)\) to denote the XOR case.
Modeling of the Tables. To synthesize the upper characteristic and the lower characteristic, we use different 8 variables to model the different tables for one S-box. According to the definitions of tables, the modeling point set is given in Table 7:

<table>
<thead>
<tr>
<th>( (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) )</th>
<th>Table</th>
<th>( (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) )</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (0, 0, 0, 0, 1, 0, 1) )</td>
<td>LDDT = 1</td>
<td>( (0, 1, 0, 1, 0, 1, 0) )</td>
<td>EBCT = 1</td>
</tr>
<tr>
<td>( (0, 1, 0, 1, 0, 0, 0) )</td>
<td>UDDT = 1</td>
<td>( (0, 1, 1, 0, 1, 0, 1) )</td>
<td>LDDT = 1</td>
</tr>
<tr>
<td>( (0, 1, 1, 0, 1, 0, 1) )</td>
<td>BCT = 1</td>
<td>( (0, 1, 0, 1, 1, 1, 1) )</td>
<td>UDDT2 = 1</td>
</tr>
<tr>
<td>( (0, 1, 0, 1, 1, 0, 1) )</td>
<td>UBCT = 1</td>
<td>( (1, 1, 1, 0, 1, 0, 1) )</td>
<td>LDDT2 = 1</td>
</tr>
</tbody>
</table>

where \( \text{LDDT} + \text{LDDT} + \text{BCT} + \text{UBCT} + \text{EBCT} + \text{LBCT} + \text{UDDT2} + \text{LDCT} \leq 1 \), \( x_0, x_1 \) are the input variables for S-box in upper path, \( x_2, x_3 \) are the output variables for S-box in upper path, \( x_4, x_5 \) are the input variables for S-box in lower path and \( x_6, x_7 \) are the output variables for S-box in lower path. For example, the upper path of \( \text{UBCT} \) is \( (0, 1) \) and the lower path of \( \text{UBCT} \) is \( (1, 0) \), so we use \( (0, 1, 0, 1, 0, 0, 1) \) to indicate \( \text{UBCT} = 1 \).

Modeling of the Upper Boundary and Lower Boundary. The target cipher \( E \) is segmented into three parts automatically, such that the overall probability is maximized. We use two sets of variables \( \{ \text{tag}^u[r_x], 0 \leq r_x \leq N_r - 1 \} \) and \( \{ \text{tag}^l[r_x], 0 \leq r_x \leq N_r - 1 \} \) to identify the upper boundary and the lower boundary of the middle part. Each \( \text{tag}^u[r_x] \) or \( \text{tag}^l[r_x] \) needs to satisfy the condition in Table 8. In the upper path, when \( \text{tag}^u[i] = 1 \), \( \text{tag}^u[i + 1] \) must be 1. When \( \text{tag}^u[i] = 0 \), the table type of each cell needs to be considered. If there is a cell with the table type in Table 8, \( \text{tag}^u[i + 1] = 1 \). Otherwise, \( \text{tag}^u[i + 1] = 0 \). The lower path is a similar case. For instance, in Example 2, because the first three rounds have only \( \text{UDDT2} \), \( \text{tag}^u[0] = \text{tag}^u[1] = \text{tag}^u[2] = 0 \). And sin the fourth round has one UBCT, \( \text{tag}^u[3] = 1 \) and \( \text{tag}^u[i] = 1, i > 3 \). Consequently, the upper boundary of \( E_m \) is marked by the layer of S-boxes where \( \text{tag}^u[r_x] \) turns to 1 in the forward direction and the lower boundary is the layer of S-boxes where \( \text{tag}^l[r_x] \) turns to 1 in the backward direction. Thus S-boxes with \( \text{tag}^u[r_x] = \text{tag}^l[r_x] = 1 \) all belong to \( E_m \).

<table>
<thead>
<tr>
<th>( \text{tag}^u[r_x - 1]/\text{tag}^u[r_x + 1] + \text{Sum}[r_x] )</th>
<th>( \text{tag}^u[r_x]/\text{tag}^l[r_x] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>1</td>
</tr>
</tbody>
</table>

Remark 3. For the upper characteristic, the S-RULE, the upper boundary \( \text{tag}^u \), and XOR-RULE are forward propagation from \( i \)-th round to \( i + 1 \)-th round. For the lower characteristic, in turn, the S-RULE, the lower boundary \( \text{tag}^l \), and XOR-RULE are reverse propagation from \( i + 1 \)-th round to \( i \)-th round.

Objective Function. According to equation 3, the objective to minimize is the number of conditions consuming for \( E \):

\[
\text{obj} = 2c_0 + 2c_1 + c_0' + c_1' + c_m.
\]

We use the boundary tags \( \text{tag}^u \) and \( \text{tag}^l \) to automatically segment \( E \) into \( E_0, E_m \) and \( E_1 \), and use the variable \( c^u \) and \( c^l \) to identify the condition consumed in XOR operation. Then
As a result, our model is more likely to offer a good boomerang cluster when the input with truncated input and output differences together with the corresponding approximate which embedded yoyo within a boomerang.

The basic idea of modelling clusters’ probability, which transforms calculating the probability to simply recording the condition consumed, can be generalized to other attacks for word-oriented block ciphers, such as the boomeyong attack [RSP21], the mixture differential cryptanalysis [Gra18], and the retracing boomerang attack [DKRS20], which embedded yoyo within a boomerang.

### 5.3 Discussion

Similar to [DDV20], our model has the advantage of handling dependencies in the middle rounds automatically without specifying $E_m$ in advance. Besides, our model has two remarkable features as follows.

1. It incorporates Property 1 and Proposition 1 of DBCT so as to evaluate the probability of $E_m$ more accurately. Specifically, $UBCT \cdot EBCT \cdot LBCT$ which involves $t + 2$ active S-boxes actually consumes only about one condition, i.e., contributes a probability about $2^{-t}$. Therefore, our model reflects the probability of $E_m$ more accurately than just counting the number of active S-boxes of $E_m$ as has been done in previous works [DDV20, HBS21].

2. The clustering effect in both $E_0$ and $E_1$ are also well considered. We use variables $tag^u/\tag^l$ to mark the boundaries of $E_m$ so that the technique borrowed from the truncated differential analysis can be applied to take into account the clustering effect in $E_0$ and $E_1$. As a result, our model is more likely to offer a good boomerang cluster when the input and output differences are instantiated, which will be exemplified in the next subsection.

The basic idea of modelling clusters’ probability, which transforms calculating the probability to simply recording the condition consumed, can be generalized to other attacks for word-oriented block ciphers, such as the boomeyong attack [RSP21], the mixture differential cryptanalysis [Gra18], and the retracing boomerang attack [DKRS20], which embedded yoyo within a boomerang.

### 5.4 Boomerang Clusters by Applying the New Modeling

For a specific block cipher, the first step is to use our model to get a good boomerang cluster with truncated input and output differences together with the corresponding approximate probability. The second step is to instantiate the input and output differences and obtain the exact probability by experiments or computations if possible.

We apply the new model to CRAFT and obtain boomerang distinguishers of 6-14 rounds, including new 9-round and a new 10-round boomerang distinguishers with higher probability than the ones presented in [HBS21]. Fig. 12 depicts the 9-round and the 10-round boomerang distinguishers, where the 10-round boomerang distinguisher is obtained by appending one round to the 9-round boomerang distinguisher. They have the same 6-round $E_m$, which is divided automatically by the MILP model. The input and output differences in the 9-round distinguisher are chosen as follows:

\[
\Delta = 0x000a00aa00000000a, \nabla = 0x0000000000000000.
\]

The estimated probability by our method is:

\[
P_{E}(\Delta \xrightarrow{9r} \nabla) = \sum_{\Delta_1, \nabla_1} P_{E_0}(\Delta \xrightarrow{2r} \Delta_1) \cdot P_{E_m}(\Delta_1 \xrightarrow{6r} \nabla_1) \cdot P_{E_1}(\nabla_1 \xrightarrow{1r} \nabla) = 2^{-14.65}.
\]
Figure 12: A Boomerang Distinguisher for 9/10 rounds of CRAFT

The experimental probability is about $2^{-12.95}$ which is higher than the probability $2^{-14.50}$ of the 9-round boomerang distinguisher in [HBS21].

The input and output differences of the 10-round boomerang distinguisher are as follows:

$\Delta = 0\times000a0a0aa0000000a, \nabla = 0\times0000a000000000a000.$

And the estimated probability by our method is:

$$P_E(\Delta_{10r} \iff \nabla) = \sum_{\Delta_1, \nabla_1} P_E(\Delta_{2r} \iff \Delta_1) \cdot P_E(\Delta_1 \iff \nabla_1) \cdot P_E(\nabla_1 \iff \nabla) = 2^{-19.60}.$$  

The experimental probability is approximately $2^{-16.40}$ which is higher than the probability $2^{-18.17}$ of the 10-round boomerang distinguisher in [HBS21]. Our source code is provided in https://drive.google.com/file/d/1DIExHZpL0rbv9h1Ma0JrCXC0b3QpQMqR/view?usp=sharing.

6 Conclusion

In this paper, we observe an exciting property of DBCT that the ladder switch and S-box switch constitute most cases for two continuous S-box and all cases for certain S-boxes in boomerang attacks. The meaning of this observation is at least twofold. From the point of view of cryptanalysis, when there is strong dependency between the two differential trails (this is the case for many lightweight ciphers, such as CRAFT), DBCT helps to capture dependency easily, and when the S-box is hard, the treatment of dependency can be
simplified further, while this is not unveiled in previous works. For hard S-boxes with a complex linear layer, the property of the extension of DBCT also shows that the interactions between two S-box layers matter and should be treated carefully to avoid proposing flawed boomerang distinguishers. From the point of view of designers, for a cipher using a lightweight linear layer, one needs to pay more attention to the DBCT uniformity when choosing the S-box.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments. This research is supported by the National Natural Science Foundation of China (Grants No.62202460, 6202036, 62132008, 62172410, 62032014), the National Key Research and Development Program of China (No.2018YFA0704704 and No.2022YFB2701900), the Youth Innovation Promotion Association of Chinese Academy of Sciences and the Fundamental Research Funds for the Central Universities.

References


New Properties of Double Boomerang Connectivity Table


A Proofs

A.1 Proof of Proposition 2

Proof. If $S$ is hard, there is only the S-box switch in two continuous S-boxes for $\alpha_1, \beta_3 \neq 0$, i.e.,

$$
\text{DBCT}(\alpha_1, \beta_3) = \sum_{\alpha_2, \beta_2} \text{UBCT}(\alpha_1, \alpha_2, \beta_2) \cdot \text{LBCT}(\alpha_2, \beta_2, \beta_3)
$$

$$
= \sum_{\beta_1, \alpha_2, \beta_3} \text{EBCT}(\alpha_1, \beta_1, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3)
$$

$$
= \sum_{\beta_1, \alpha_2 = \beta_2, \alpha_3} \text{EBCT}(\alpha_1, \beta_1, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3)
$$

$$
= \sum_{\alpha_2 = \beta_2} \text{EBCT}(\alpha_1, \alpha_1, \alpha_2, \beta_2) \cdot \text{EBCT}(\beta_2, \beta_2, \beta_3, \beta_3)
$$

$$
= \sum_{\alpha_2} \text{DDT}(\alpha_1, \alpha_2) \cdot \text{DDT}(\alpha_2, \beta_3).
$$

In other words, the product is non-zero only with $\alpha_2 = \beta_2$. Thus for three continuous S-boxes, a necessary condition for the product to be nonzero is $\alpha_2 = \beta_2$, which further leads to $\alpha_3 = \beta_3$. Consequently,

$$
\text{3-BCT}(\alpha, \beta) = \sum_{\alpha_2, \beta_2, \alpha_3, \beta_3} \text{UBCT}(\alpha, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3) \cdot \text{LBCT}(\alpha_3, \beta_3, \beta)
$$

$$
= \sum_{\alpha_3, \beta_3} \left( \sum_{\alpha_2, \beta_2} \text{UBCT}(\alpha, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3) \right) \cdot \text{LBCT}(\alpha_3, \beta_3, \beta)
$$

$$
= \sum_{\alpha_3, \beta_3} \left( \sum_{\beta', \alpha_2, \beta_2} \text{EBCT}(\alpha, \beta', \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3) \right) \cdot \text{LBCT}(\alpha_3, \beta_3, \beta)
$$

$$
= \sum_{\alpha_3, \beta_3} \left( \sum_{\beta', \alpha_2 = \beta_2} \text{EBCT}(\alpha, \beta', \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_3, \beta_3) \right) \cdot \text{LBCT}(\alpha_3, \beta_3, \beta).
$$
Due to $\alpha_2 = \beta_2$, we get $\alpha = \beta'$ and $\alpha_3 = \beta_3$. Thus

$$3\text{-BCT}(\alpha, \beta) = \sum_{\alpha_2 = \beta_2} \left( \sum_{\alpha_2 = \beta_2} \text{EBCT}(\alpha_2, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_2, \beta_2, \alpha_2, \beta_2) \cdot \text{EBCT}(\alpha_3, \beta_3, \beta) \right) \cdot \text{LBCT}(\alpha_3, \beta_3, \beta)$$

$$= \sum_{\alpha_2, \alpha_3} \text{EBCT}(\alpha, \alpha, \alpha_2) \cdot \text{EBCT}(\alpha_2, \alpha_2, \alpha, \alpha_3) \cdot \text{EBCT}(\alpha_3, \alpha_3, \beta, \beta)$$

$$= \sum_{\alpha_2, \alpha_3} \text{DDT}(\alpha, \alpha_2) \cdot \text{DDT}(\alpha_2, \alpha_3) \cdot \text{DDT}(\alpha_3, \beta).$$

Similarly, it can go straight to the case with more continuous S-boxes, i.e.,

$$t\text{-BCT}(\alpha, \beta) = \sum_{\alpha_2, \ldots, \alpha_t} \text{DDT}(\alpha, \alpha_2) \cdot \text{DDT}(\alpha_2, \alpha_3) \cdot \ldots \cdot \text{DDT}(\alpha_t, \beta).$$

\[\square\]

### A.2 Proof of objective function

**Proof.** Let $\text{obj} = 1 + 2 + 3$. Due to the facts $tag^u[B^u] = 0, tag^u[B^u+1] = 1$ and $tag^l[B^l] = 0, tag^l[B^l-1] = 1$,

1. $\sum_{1 \leq i \leq n-1} (2 - tag^u[i]) \cdot c^u[i][j] + (2 - tag^l[i]) \cdot c^l[i][j]$

   $$= \sum_{1 \leq i \leq n-1} 2 \cdot c^u[i][j] + \sum_{r \leq j \leq n} 2 \cdot c^l[i][j] + \sum_{r+1 \leq j \leq n-1} c^u[i][j] + c^l[i][j]$$

   $$= 2c_0 + 2c_1 + X.$$

2. $\sum_{0 \leq i \leq n-1} (tag^u[i+1] - tag^u[i]) \cdot \text{UDDT2}[i][j] + (tag^l[i-1] - tag^l[i]) \cdot \text{LDDT2}[i][j]$

   $$= \sum_{0 \leq j \leq n-1} \text{UDDT2}[B^u][j] + \text{LDDT2}[B^l][j]$$

   $$= c'_0 + c'_1.$$

3. $\sum_{1 \leq i \leq n-1} tag^u[i] \cdot \text{UDDT2}[i][j] + tag^l[i] \cdot \text{LDDT2}[i][j] + \text{UBCT}[i][j] + \text{BCT}[i][j]$

   $$= \sum_{r_0 + 1 \leq j \leq r_0 + r_m} \text{UDDT2}[i][j] + \text{LDDT2}[i][j] + \text{UBCT}[i][j] + \text{BCT}[i][j]$$

   $$= c_m = X.$$

Thus $\text{obj} = 2c_0 + 2c_1 + c'_0 + c'_1 + c_m$. \[\square\]

## B Specification of CRAFT, TweAES, and Deoxys-BC

### B.1 Specification of CRAFT

CRAFT is a lightweight tweakable block cipher which introduced by Beierle et al. [BLMR19] at FSE 2019. CRAFT supports 64-bit message, 128-bit key and 64-bit tweak, and its round function is composed of involutory operations. The round function is shown in Fig. 13 and its operations are listed as follows:
• **MixColumns** (MC): The MC layer is the multiplication of internal state by the following binary matrix:

\[ MC = \begin{pmatrix}
1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}. \]

• **AddRoundConstants** (ARC): The state XOR-ed constant in cells 4 and 5.

• **AddTweaKey** (ATK): A 64-bit round tweakey is XOR-ed with state.

• **PermuteNibbles** (PN): The PN is an involutory permutation over nibbles of state:

\[ P = [15, 12, 13, 14, 10, 9, 8, 11, 6, 5, 4, 7, 1, 2, 3, 0]. \]

• **Sbox** (SB): CRAFT uses a 4-bit involutory S-box, the detail is given in Table 9.

<table>
<thead>
<tr>
<th>Table 9: The S-box used in CRAFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
</tr>
<tr>
<td>( S(x) )</td>
</tr>
</tbody>
</table>

Figure 13: A round function of CRAFT

### B.2 Specification of TweAES

The tweakable block cipher **TweAES** is one of the underlying primitives of Authenticated Encryption with Associated Data (AEAD) scheme ESTATE [CDJ+20], which is a second-round candidate of the NIST Lightweight Cryptography Standardization project. It is tweaked from AES-128 [DR02] and takes in as input a 4-bit tweak, a 128-bit key and a 128-bit block. Its round function has five operations, which are identical to that of AES except **AddTweak**. Next, we briefly describe the round function of **TweAES**.

- **SubBytes**: **TweAES** uses the same 8-bit S-box as AES.
- **ShiftRows**: The bytes in the \( i \)-th row are cyclically shifted by \( i \) places to the left.
- **MixColumns**: Multiply each column with an invertible MDS matrix.
- **AddKey**: XORRed 128-bit round key.
• **AddTweak**: The 8-bit tweak, which is expanded from 4-bit tweak, is added to the least significant bit of each byte in top two rows of the state at an interval of 2 rounds.

In more detail, \texttt{MixColumns} mixes every column by multiplication of a $4 \times 4$ MDS matrix

$$MC = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

over the finite field $\mathbb{F}_{2^8}$, where the irreducible polynomial is $x^8 + x^4 + x^3 + x + 1$.

### B.3 Specification of Deoxys-BC

\texttt{Deoxys-BC} is an AES-based tweakable block cipher [JNPS16], based on the tweakkey framework [JNP14]. The \texttt{Deoxys} authenticated encryption scheme makes use of two versions of the cipher as its internal primitive: \texttt{Deoxys-BC-256} and \texttt{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 tweak key state. For \texttt{Deoxys-BC-256} the tweak key size is 256 bits.

\texttt{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 $4 \times 4$ 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. \texttt{Deoxys-BC-256} has 14 rounds.

Similarly to the AES, one round of \texttt{Deoxys-BC} has the following four transformations applied to the internal state in the order specified below:

• **AddRoundTweakey** – XOR the 128-bit round subtweakey to the internal state.

• **SubBytes** – Apply the 8-bit AES S-box 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 $4 \times 4$ constant MDS matrix of AES.

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

We denote the concatenation of the key $K$ and the tweak $T$ as $KT$, i.e. $KT = K||T$. The tweak key state is then divided into 128-bit words. More precisely, in \texttt{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 $W_1$. Finally, we denote by $STK_i$ the 128-bit subtweakey that is added to the state at round $i$ during the **AddRoundTweakey** operation. For \texttt{Deoxys-BC-256}, a subtweakey is defined as $STK_i = TK_1^i \oplus TK_2^i \oplus RC_i$. The 128-bit words $TK_1^i, TK_2^i$ are outputs produced by a special tweakkey schedule algorithm, initialised with $TK_1^0 = W_1$ and $TK_2^0 = W_2$ for \texttt{Deoxys-BC-256}. The tweakkey schedule algorithm is defined as $TK_1^{i+1} = h(TK_1^i), TK_2^{i+1} = h(LFSR_2(TK_2^i))$, 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 tweak key word numbered by the usual AES byte ordering.
C Supplementary Materials for Boomerang Attacks

C.1 A Note on the Boomerang Attack with Swapped Ciphertexts

Recall that the probability of the 7-round boomerang distinguisher of CRAFT involves four DBCT. As the DBCT is equivalent to the S-box switch in most cases, i.e., a quartet is formed by two pairs of the same value, we check how the probability changes when new pairs of ciphertexts are generated by swapping certain cells of obtained pairs of ciphertexts, as the attacker does in the yoyo attack.

We reuse the 7-round boomerang distinguisher of CRAFT and test three kind of S-boxes, namely the S-box of CRAFT, PRESENT and TWNIE. Note the latter two are hard S-boxes. We then consider three pairs of boomerang attacks as follows.

Case A: Standard boomerang distinguisher with exact input difference and output difference (\(\Delta_{in}, \Delta_{out}\)) allowing the highest probability.

Case B: With the same difference as in Case A, but only check if the difference of the returned pair follows the truncated pattern of \(\Delta_{in}\).

Case C: Truncated boomerang distinguisher where the input and output difference are random for active cells.

Case A’, B’, C’: Respective variants of Case A, B and C where the output cells are swapped if the cells of \(\Delta_{out}\) at the same position are active.

Note that Case B is a variant of the standard boomerang attack which allows a higher probability at the cost of a lower signal to noise ratio. Case B has been used in attacks against AES and Deoxys-BC [Sas18]. And Case C’ is actually the yoyo attack.

Table 10: The experimental probability of the 7-round boomerang distinguisher in different cases

<table>
<thead>
<tr>
<th>S-box</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>A’</th>
<th>B’</th>
<th>C’</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRAFT</td>
<td>(2^{-10.11})</td>
<td>(2^{-9.74})</td>
<td>(2^{-12.80})</td>
<td>(2^{-8.70})</td>
<td>(2^{-8.36})</td>
<td>(2^{-9.16})</td>
</tr>
<tr>
<td>PRESENT</td>
<td>(2^{-15.14})</td>
<td>(2^{-14.74})</td>
<td>(2^{-14.04})</td>
<td>(2^{-11.40})</td>
<td>(2^{-10.01})</td>
<td>(2^{-10.19})</td>
</tr>
<tr>
<td>TWNIE</td>
<td>(2^{-15.53})</td>
<td>(2^{-14.50})</td>
<td>(2^{-14.54})</td>
<td>(2^{-11.64})</td>
<td>(2^{-10.60})</td>
<td>(2^{-10.68})</td>
</tr>
<tr>
<td>Random case</td>
<td>(2^{-6.94})</td>
<td>(2^{-6.60})</td>
<td>(2^{-6.60})</td>
<td>(2^{-6.64})</td>
<td>(2^{-6.60})</td>
<td>(2^{-6.60})</td>
</tr>
</tbody>
</table>

We conduct an experiment and the probabilities are summarized in Table 10. From Table 10 we have two observations.

- For each pair of cases like (A, A’), the probability is increased with swapped ciphertexts and the increase in probability is more significant for PRESENT’s S-box and TWNIE’s S-box. This is reasonable as these two kind of S-boxes are hard and thus only allow the S-box switch for the for DBCT.

- The probabilities in Case B and C (or B’ and C’) are very close for S-boxes PRESENT’s S-box and TWNIE’s S-box which have both good BCT and DBCT uniformity. That is, there is no special input difference \(\Delta_{in}\) leading to much higher probability than others and truncated ones are good enough. This reminds us that searching for truncated boomerang distinguishers with good probability might be a good idea that is worth trying. We try this idea in Section 5.

C.2 Tables and Distinguishers
Table 11: Setting of the first six rounds of the boomerang distinguisher of TweAES

<table>
<thead>
<tr>
<th>Round</th>
<th>State difference before SubBytes</th>
<th>Tweakey difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\Delta_s = 0x11001100000000000$</td>
<td>$\Delta TK_1 = 0x11001100000000000$</td>
</tr>
<tr>
<td>7</td>
<td>$\Delta s = 0x00000000000000000$</td>
<td>$\Delta TK_7 = 0x00110011000000000$</td>
</tr>
</tbody>
</table>

Table 12: DDT of CRAFT’s 4-bit S-box

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 13: DBCT of CRAFT’s 4-bit S-box

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>1</td>
<td>256</td>
<td>40</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>24</td>
<td>8</td>
<td>8</td>
<td>32</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>32</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>-</td>
<td>32</td>
<td>32</td>
<td>-</td>
<td>32</td>
<td>32</td>
<td>-</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>256</td>
<td>32</td>
<td>32</td>
<td>40</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>256</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>40</td>
<td>8</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>256</td>
<td>8</td>
<td>32</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>40</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>256</td>
<td>8</td>
<td>32</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>40</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>256</td>
<td>16</td>
<td>-</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>48</td>
<td>16</td>
<td>16</td>
<td>64</td>
<td>16</td>
<td>16</td>
<td>48</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>-</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>256</td>
<td>24</td>
<td>32</td>
<td>8</td>
<td>32</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>40</td>
<td>8</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>a</td>
<td>256</td>
<td>8</td>
<td>-</td>
<td>16</td>
<td>8</td>
<td>80</td>
<td>16</td>
<td>64</td>
<td>-</td>
<td>8</td>
<td>128</td>
<td>16</td>
<td>8</td>
<td>80</td>
<td>16</td>
</tr>
<tr>
<td>b</td>
<td>256</td>
<td>8</td>
<td>32</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>40</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>c</td>
<td>256</td>
<td>32</td>
<td>32</td>
<td>16</td>
<td>24</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>8</td>
<td>40</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>d</td>
<td>256</td>
<td>8</td>
<td>-</td>
<td>16</td>
<td>8</td>
<td>32</td>
<td>16</td>
<td>48</td>
<td>16</td>
<td>8</td>
<td>80</td>
<td>16</td>
<td>8</td>
<td>48</td>
<td>16</td>
</tr>
<tr>
<td>e</td>
<td>256</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>40</td>
<td>16</td>
</tr>
<tr>
<td>f</td>
<td>256</td>
<td>8</td>
<td>-</td>
<td>16</td>
<td>8</td>
<td>48</td>
<td>16</td>
<td>48</td>
<td>16</td>
<td>8</td>
<td>64</td>
<td>16</td>
<td>8</td>
<td>64</td>
<td>16</td>
</tr>
</tbody>
</table>
Figure 14: Truncated boomerang attack on 8-round Deoxys-BC in the RTK1 model, which is taken from [BL22]
Figure 15: Truncated boomerang attack on 10-round Deoxys-BC in the RTK2 model, which is taken from [BL22]