IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2020, No. 4, pp. 104–129. DOI:10.46586/tosc.v2020.i4.104-129

Catching the Fastest Boomerangs

Application to SKINNY

Stéphanie Delaune*, Patrick Derbez and Mathieu Vavrille

Univ Rennes, Centre National de la Recherche Scientifique (CNRS), Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Rennes, France
{stephanie.delaune,patrick.derbez,mathieu.vavrille}@irisa.fr

Abstract. In this paper we describe a new tool to search for boomerang distinguishers. One limitation of the MILP model of Liu *et al.* is that it handles only one round for the middle part while Song *et al.* have shown that dependencies could affect much more rounds, for instance up to 6 rounds for SKINNY. Thus we describe a new approach to turn an MILP model to search for truncated characteristics into an MILP model to search for truncated boomerang characteristics automatically handling the middle rounds. We then show a new CP model to search for the best possible instantiations to identify good boomerang distinguishers. Finally we systematized the method initiated by Song *et al.* to precisely compute the probability of a boomerang. As a result, we found many new boomerang distinguishers up to 24 rounds in the TK3 model. In particular, we improved by a factor 2^{30} the probability of the best known distinguisher against 18-round SKINNY-128/256.

Keywords: Boomerang · MILP model · SKINNY

1 Introduction

Differential cryptanalysis is one of the most powerful cryptanalysis techniques. It was proposed by Biham and Shamir in [BS91] and has generated much attention since then. The aim of differential cryptanalysis is to study the propagation of differences through a cipher to highlight unexpected behaviors compared to a random permutation. Typically it concerns the existence of a differential characteristic with a high probability but we may also search for impossible transitions [BBS99].

Nowadays we know how to design ciphers resistant to differential cryptanalysis, ciphers for which we can give upper bounds on the probability of the best differential characteristics. To go further, Wagner proposed the *boomerang attack* in [Wag99]. The main idea introduced by Wagner is that combining two short differentials may lead to a better probability than one long differential. In boomerang attacks, a cipher E is regarded as the composition of two sub-ciphers E_0 and E_1 so that $E = E_1 \circ E_0$. Suppose there exist both a differential $\alpha \to \beta$ for E_0 and a differential $\gamma \to \delta$ for E_1 with probabilities p and q respectively. If we assume the two differentials are independent then we obtain a boomerang distinguisher of probability:

$$\mathbb{P}\left(E^{-1}(E(P) \oplus \delta) \oplus E^{-1}(E(P \oplus \alpha) \oplus \delta) = \alpha\right) = p^2q^2.$$

However, in practice the independence assumption does not hold, especially at the junction of both the lower and upper differentials. At SAC'07, Wang *et al.* [WKD07] first

*Stéphanie Delaune, Patrick Derbez and Mathieu Vavrille were supported by the French Agence Nationale de la Recherche through the Decrypt project under Contract ANR-18-CE39-0007.

Received: 2020-09-01 Accepted: 2020-11-01 Published: 2020-12-10

gave some evidences for non-returning boomerangs (i.e. $\mathbb{P} = 0$ instead of p^2q^2). In 2011, Murphy [Mur11] provided several examples for both AES and DES of boomerangs never coming back. Similar results were obtained by Kircanski in [Kir15]: a SAT solver is used to show that previous rectangle/boomerang attacks on XTEA [Lu09], SM3 [WKD07] and SHACAL-1 [DKK06] primitives were based on incompatible characteristics.

Recently, in [CHP+18], Cid *et al.* proposed a new tool named *boomerang connectivity table* (BCT) to overcome the dependency issues. The BCT is actually a precomputation of all boomerangs through one single Sbox. Its main advantage is to provide a unified view of the *switches* previously introduced to refine the computation of the probability [BK09, DKS14]. In [SQH19], Song *et al.* give a generalized framework for the BCT and propose a method to precisely evaluate the probability of a boomerang. They reevaluated the probability of several boomerang distinguishers from [LGS17] against both SKINNY and AES, showing their real probability was much higher than expected.

Searching boomerangs. One natural question when facing a new cryptanalysis technique is how to find the best distinguishers. For boomerang distinguishers, the classical approach is to first search for two short characteristics with high probability and to combine them. But we believe this approach should now be deprecated since the dependency in the middle rounds may hugely affect the probability of the distinguisher and thus it seems sub-optimal to search for both the lower and upper differentials independently.

In [CHP+17], Cid *et al.* used an MILP model to study the ladder switch for a boomerang attack on Deoxys. A more generic approach was proposed in [LS19], where Liu *et al.* describe an MILP model to directly search for the best boomerang distinguisher against the block cipher GIFT. The cipher is decomposed in three parts E_0, E_m and E_1 where E_m is restricted to one single round, the junction of both differentials which handles the BCTs. With this model they found a new boomerang distinguisher on 19-round GIFT, achieving a better probability than merging two optimal short trails.

Contribution. In this paper, we propose to go further than both [SQH19] and [LS19] by providing a new tool to search for boomerang distinguishers. One limitation of the MILP model of Liu *et al.* is that it handles only one round for the middle part while Song *et al.* have shown that dependencies could affect much more rounds, for instance up to 6 rounds for SKINNY. First, we propose a new approach to turn an MILP model to search for truncated characteristics into an MILP model to search for truncated boomerang characteristics. The main novelty is that this model handles the dependencies in the middle rounds automatically. Furthermore, there is no need to specify which rounds are the middle ones, this is also directly handled by the model. Second, we propose a new Constraint Programming (CP) model to search for the best instantiation of a truncated boomerang characteristic. This model even goes further by clustering instantiations to improve the probabilities. Finally, we systematized the method from [SQH19] to precisely compute the probability of a boomerang.

We applied our new set of tools to the block cipher SKINNY [BJK+16] and found many new distinguishers on all versions of the ciphers. Our results are given in Table 1. All previous results from [SQH19] are improved, in particular we found a new boomerang distinguisher on 18-round SKINNY-128/256 (i.e. on the TK2 model) with probability $2^{-47.37}$ while the previous best distinguisher had probability $2^{-77.83}$. We experimentally verified some of the distinguishers to confirm the probabilities.

All our source codes and results (including characteristics) are publicly available at:

https://gitlab.inria.fr/pderbez/boomerangskinny

Organization of the paper. In Section 2 we recall previous definitions regarding DDT and BCT and introduced new interesting tables as the *Extended Boomerang Connectivity Table*. In Section 3 we detail how to precisely compute the probability of a boomerang on

Table 1: Results for different versions and number of rounds on SKINNY. Probabilities marked with asterisks have been validated experimentally. The four previous results from [SQH19] are also given in parenthesis.

version	block size	nb of rounds	proba
SK	64	11	$2^{-59.23}$
	128	13	$2^{-112.53}$
		14	$2^{-128.52}$
TK1	64	14	$2^{-40.34*}$
		15	$2^{-53.16}$
	128	14	$2^{-42.27}$
		15	$2^{-69.14}$
		16	$2^{-87.15}$
		17	$2^{-107.84}$

version	block size	nb of rounds	proba
TK2	64	17	$2^{-27.65*}$ $(2^{-29.78})$
		18	$2^{-38.20*}$
		19	$2^{-54.36}$
	128	18	$2^{-47.37}$ $(2^{-77.83})$
		19	$2^{-61.83}$
		20	$2^{-85.77}$
TK3	64	22	$2^{-39.44*}$ $(2^{-42.98})$
		23	$2^{-57.93}$
	128	22	$2^{-47.34}$ $(2^{-48.30})$
		23	$2^{-61.80}$
		24	$2^{-86.09}$

the middle rounds. In Section 4 we describe our new MILP model to search for truncated boomerang characteristics and give results for SKINNY. Finally in Section 5 we describe our CP model to instantiate truncated boomerang characteristics and detail the new boomerang distinguisher on 18-round SKINNY-128/256 we found.

2 Definitions

In this section we introduce the tables we will need to compute the probability of a boomerang characteristic. Figure 1 shows a representation of the differences used for each table. We also exhibit several properties on those tables.

2.1 Tables

We first recall the definitions of the DDT and the BCT [CHP+18] for an n-bit (invertible) Sbox S.

Definition 1. Given $\gamma, \theta, \delta \in \mathbb{F}_2^n$ three differences, the DDT and the BCT are defined as

$$\mathrm{DDT}(\gamma, \theta) = \#\{x \in \mathbb{F}_2^n \mid S(x) \oplus S(x \oplus \gamma) = \theta\}$$
$$\mathrm{BCT}(\gamma, \delta) = \#\{x \in \mathbb{F}_2^n \mid S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma\}$$

Recently in [WP19] two extra tables have been introduced under the name BDT and BDT'. We recall here their definitions but change their names to UBCT (for Upper BCT) and LBCT (for Lower BCT). We believe those names emphasize better the fact that the Upper (resp. Lower) BCT focuses on the upper (resp. lower) characteristic.

Definition 2. [WP19] Given $\gamma, \theta, \delta \in \mathbb{F}_2^n$ three differences, the UBCT and the LBCT are defined as

$$\mathrm{UBCT}(\gamma, \theta, \delta) = \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\}$$
$$\mathrm{LBCT}(\gamma, \lambda, \delta) = \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \lambda) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\}$$

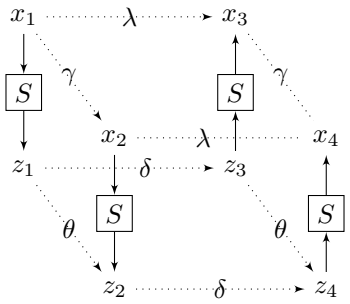

Figure 1: Differences used in the tables for a single layer of Sbox

These two tables are symmetric in the sense that one focuses on the upper characteristic, and the other focuses on the lower characteristic. All the properties found on one have their counterparts for the other table.

Finally, we define a last table called Extended BCT (EBCT) and originally proposed by Boukerrou *et al.* in [BHL+20]. This table counts the number of values such that the boomerang will return on a single Sbox, while fixing all the differences.

Definition 3. Given $\gamma, \theta, \lambda, \delta \in \mathbb{F}_2^n$ four differences, the EBCT is defined as

$$\mathrm{EBCT}(\gamma,\theta,\lambda,\delta) = \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S(x) \oplus S(x \oplus \lambda) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\}$$

Note that the EBCT contains 2^{4n} entries but only approximately $2^{5n-3n} = 2^{2n}$ should be non zero. Those entries can be computed in 2^{3n} simple operations. The idea is to exhaust all the possible values for x, γ, and λ and each time to compute the corresponding value of both θ and δ before checking whether the last equation holds. Actually all the tables described in this Section have approximately 2^{2n} non-zero entries and can be computed in 2^{3n} operations. In [Dun18], Dunkelman shows an algorithm to construct the BCT in $O(2^{2n})$ operations and it is an open problem to know whether the UBCT/LBCT and the EBCT could be constructed with the same complexity.

Definition 4. Let $\gamma, \theta, \lambda, \delta \in \mathbb{F}_2^n$. We define for each table the transition probability as

$$\begin{array}{rclcrcl} \mathbb{P}_{\mathrm{DDT}}(\gamma,\theta) & = & \mathrm{DDT}(\gamma,\theta)/2^n & & \mathbb{P}_{\mathrm{UBCT}}(\gamma,\theta,\delta) & = & \mathrm{UBCT}(\gamma,\theta,\delta)/2^n \\ \mathbb{P}_{\mathrm{BCT}}(\gamma,\delta) & = & \mathrm{BCT}(\gamma,\delta)/2^n & & \mathbb{P}_{\mathrm{LBCT}}(\gamma,\lambda,\delta) & = & \mathrm{LBCT}(\gamma,\lambda,\delta)/2^n \end{array}$$

$$\mathbb{P}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta) = \mathrm{EBCT}(\gamma,\theta,\lambda,\delta)/2^n$$

2.2 Table Properties

There are many properties linking each table. The BCT and the DDT have already been studied and properties between them can be found in [CHP+18, SQH19]. Here we present properties on the EBCT, as well as the UBCT and LBCT.

Property 1.

1. Properties on the EBCT

(a)
$$\begin{array}{rcl} \forall \alpha,\gamma,\theta \in \mathbb{F}_2^n,\ \mathbb{1}_{\alpha=0} \cdot \mathrm{DDT}(\gamma,\theta) & = & \mathrm{EBCT}(\alpha,0,\gamma,\theta) \\ & = & \mathrm{EBCT}(0,\alpha,\gamma,\theta) \\ & = & \mathrm{EBCT}(\gamma,\theta,\alpha,0) \\ & = & \mathrm{EBCT}(\gamma,\theta,0,\alpha) \end{array}$$

(b) $\forall \alpha, \beta, \delta \in \mathbb{F}_2^n,\ \mathbb{1}_{\alpha=\beta}\mathrm{DDT}(\alpha, \delta) = \mathrm{EBCT}(\alpha, \delta, \beta, \delta)$ (upper sbox switch)

(c) $\forall \alpha, \beta, \delta \in \mathbb{F}_2^n,\ \mathbb{1}_{\beta=\delta}\mathrm{DDT}(\alpha, \beta) = \mathrm{EBCT}(\alpha, \beta, \alpha, \delta)$ (lower sbox switch)

(d) $\forall \gamma, \delta \in \mathbb{F}_2^n,\ \mathrm{BCT}(\gamma, \delta) = \sum_{\theta,\lambda \in \mathbb{F}_2^n} \mathrm{EBCT}(\gamma, \theta, \lambda, \delta)$

2. Properties on the UBCT

(a) $\forall \gamma, \theta, \delta \in \mathbb{F}_2^n,\ \mathrm{UBCT}(\gamma, \theta, \delta) = \sum_{\lambda \in \mathbb{F}_2^n} \mathrm{EBCT}(\gamma, \theta, \lambda, \delta)$

(b) $\forall \alpha, \beta \in \mathbb{F}_2^n,\ \mathrm{DDT}(\alpha, \beta) = \mathrm{UBCT}(\alpha, \beta, \beta)$ (sbox switch)

(c) $\forall \alpha, \delta \in \mathbb{F}_2^n,\ \mathrm{UBCT}(\alpha, 0, \delta) = \mathrm{UBCT}(0, \alpha, \delta) = 2^n \cdot \mathbb{1}_{\alpha=0}$ (ladder switch)

(d) $\forall \gamma, \delta \in \mathbb{F}_2^n,\ \mathrm{BCT}(\gamma, \delta) = \sum_{\theta \in \mathbb{F}_2^n} \mathrm{UBCT}(\gamma, \theta, \delta)$

3. Properties on the LBCT

(a) $\forall \gamma, \lambda, \delta \in \mathbb{F}_2^n,\ \mathrm{LBCT}(\gamma, \lambda, \delta) = \sum_{\theta \in \mathbb{F}_2^n} \mathrm{EBCT}(\gamma, \theta, \lambda, \delta)$

(b) $\forall \alpha, \beta \in \mathbb{F}_2^n,\ \mathrm{DDT}(\alpha, \beta) = \mathrm{LBCT}(\alpha, \alpha, \beta)$ (sbox switch)

(c) $\forall \gamma, \alpha \in \mathbb{F}_2^n,\ \mathrm{LBCT}(\gamma, 0, \alpha) = \mathrm{LBCT}(\gamma, \alpha, 0) = 2^n \cdot \mathbb{1}_{\alpha=0}$ (ladder switch)

(d) $\forall \gamma, \delta \in \mathbb{F}_2^n,\ \mathrm{BCT}(\gamma, \delta) = \sum_{\lambda \in \mathbb{F}_2^n} \mathrm{UBCT}(\gamma, \lambda, \delta)$

Proof. All the properties directly follow from the definitions, and the property that $\forall \alpha \in \mathbb{F}_2^n, \mathrm{DDT}(0, \alpha) = \mathrm{DDT}(\alpha, 0) = 2^n \cdot \mathbb{1}_{\alpha=0}$ (injectivity of the Sbox). A detailed proof of those properties is given in Appendix C. □

The following property was already stated in [SQH19] in different terms. It explains the sandwich construction, when stating that the middle part of the cipher bears all the dependencies between the two characteristics.

Property 2. Let $\alpha, \beta \in \mathbb{F}_2^n$, and let $\delta \sim \mathcal{U}(\mathbb{F}_2^n)$ be a random variable following the uniform distribution over $\mathbb{F}_2^n$. Then

$$\mathbb{E}_\delta\left(\mathbb{P}_{\mathrm{UBCT}}(\alpha, \beta, \delta)\right) = \mathbb{P}_{\mathrm{DDT}}(\alpha, \beta)^2 \qquad \mathbb{E}_\delta\left(\mathbb{P}_{\mathrm{LBCT}}(\delta, \alpha, \beta)\right) = \mathbb{P}_{\mathrm{DDT}}(\alpha, \beta)^2$$

Proof. We prove below the property for the UBCT, the one for the LBCT can be done in a similar way.

$$
\begin{aligned}
\mathbb{E}_\delta\left(\mathbb{P}_{\mathrm{UBCT}}(\alpha,\beta,\delta)\right) &= \sum_{\delta\in\mathbb{F}_2^n}\frac{1}{2^n}\mathbb{P}_{\mathrm{UBCT}}(\alpha,\beta,\delta)\\
&= 2^{-2n}\cdot\sum_{\delta\in\mathbb{F}_2^n}\mathrm{UBCT}(\alpha,\beta,\delta)\\
&= 2^{-2n}\cdot\sum_{\delta\in\mathbb{F}_2^n}\#\left\{x\in\mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x)\oplus S(x\oplus\alpha)=\beta \\ S^{-1}(S(x)\oplus\delta)\oplus S^{-1}(S(x\oplus\alpha)\oplus\delta)=\alpha \end{array}\right\}\\
&= 2^{-2n}\cdot\sum_{\delta\in\mathbb{F}_2^n}\#\left\{x\in\mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x)\oplus S(x\oplus\alpha)=\beta \\ S^{-1}(S(x)\oplus\delta)\oplus S^{-1}(S(x)\oplus\beta\oplus\delta)=\alpha \end{array}\right\}\\
&= 2^{-2n}\cdot\sum_{\delta\in\mathbb{F}_2^n}\#\left\{(x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x)\oplus S(x\oplus\alpha)=\beta \\ S^{-1}(y)\oplus S^{-1}(y\oplus\beta)=\alpha \\ y=S(x)\oplus\delta \end{array}\right\}\\
&= 2^{-2n}\cdot\#\bigcup_{\delta\in\mathbb{F}_2^n}\left\{(x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x)\oplus S(x\oplus\alpha)=\beta \\ S^{-1}(y)\oplus S^{-1}(y\oplus\beta)=\alpha \\ y=S(x)\oplus\delta \end{array}\right\}\\
&= 2^{-2n}\cdot\#\left\{(x,y)\in\mathbb{F}_2^n\times\mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x)\oplus S(x\oplus\alpha)=\beta \\ S^{-1}(y)\oplus S^{-1}(y\oplus\beta)=\alpha \end{array}\right\}\\
&= 2^{-2n}\cdot\#\left(\begin{array}{l} \{x\in\mathbb{F}_2^n \mid S(x)\oplus S(x\oplus\alpha)=\beta\} \\ \times\left\{y\in\mathbb{F}_2^n \mid S^{-1}(y)\oplus S^{-1}(y\oplus\beta)=\alpha\right\} \end{array}\right)\\
&= 2^{-2n}\cdot\mathrm{DDT}(\alpha,\beta)^2\\
&= \mathbb{P}_{\mathrm{DDT}}(\alpha,\beta)^2
\end{aligned}
$$

□

This property helps us to understand the sandwich construction. The middle part of the cipher is where the dependencies between the upper and the lower characteristics happen. When the middle is big enough, the differences after the middle rounds in the upper characteristic will follow a uniform distribution, and thus the property can be applied to compute the probability for both E_0 and E_1 as the square of probability of the differential characteristics.

3 Probability of Boomerang Characteristics

In this section we detail the computation of the probability of a boomerang characteristic. We illustrate this on the block cipher `SKINNY`.

3.1 SKINNY

`SKINNY` is a family of very lightweight tweakable block ciphers designed by Beierle *et al.* and presented at CRYPTO'16 [BJK+16]. The round function of `SKINNY` follows a classical SPN structure and a non-MDS binary matrix is used for the **MixColumns** operation. The matrix has a low weight (only half of the coefficient are non-zero) and the tweakey is xored to the two first rows only. It enciphers blocks of length $n = 64$ or $n = 128$ bits seen as a 4×4 matrix of cells. We use positions to refer to them: the position 0 is the one at the top-left corner, and we then number them from left to right, and from top to bottom. `SKINNY` has three main tweakey size versions: the tweakey size can be equal to t=64 or 128 bits, t=128 or 256 bits, and t=192 or 384 bits, and we denote $z = n/t$ the tweak size

to the block size ratio. The round function of SKINNY is depicted on Figure 2, and the number of rounds is directly derived from the z value: between 32 rounds for the 64/64 version up to 56 for the 128/384 version. We refer the interested reader to [BJK+16] for more details.

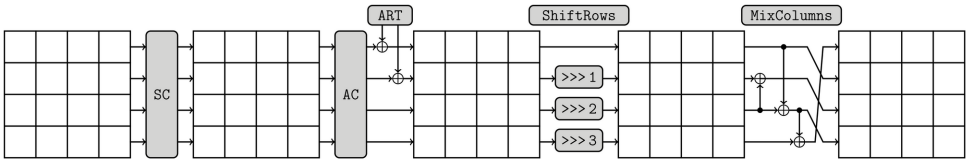

Figure 2: Round function of SKINNY [BJK+16]

3.2 Computing Probability

Given a boomerang characteristic, i.e. two differential characteristics for the upper and the lower characteristic, we now want to compute the probability for the boomerang to return. To compute the probability, one simply has to multiply the probability of transition for each Sbox separately. To compute the probability of transition for one particular Sbox, the tables defined in Section 2 are used, depending on the configuration of the differences on the characteristic. We show on an example how to choose the tables depending on the situation.

Example 1. Figure 3 shows a toy example of boomerang characteristic with two differential characteristics on 3 rounds of SKINNY-64 (the upper characteristic is above the lower characteristic). We suppose that we are not adding differences in the key (SK model), so we omit the key from the figure as it will not modify the differences. Lime (for the upper characteristic) and pink (for the lower characteristic) colored cells are non zero differences, and grey cells are unspecified differences. All the differences are given in hexadecimal. We denote the input and output differences by $\Delta_{ex} = [0, \mathrm{d}, \mathrm{d}, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0]$ and $\nabla_{ex} = [0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ respectively.

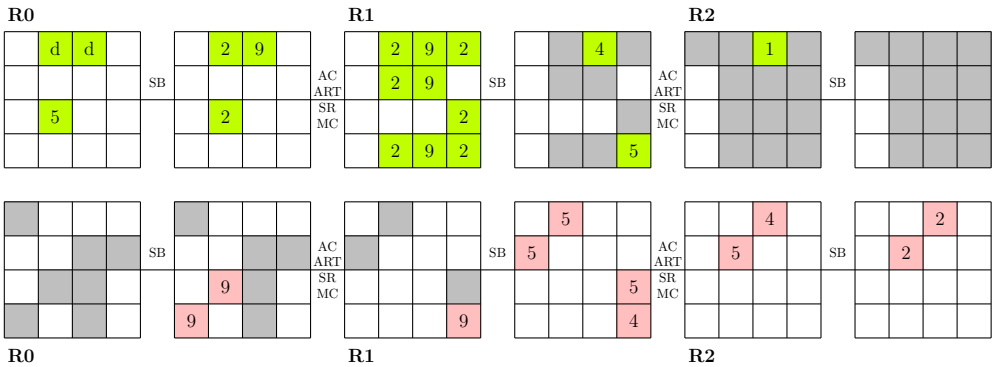

Figure 3: Toy example of boomerang characteristic on three rounds of SKINNY-64 in the SK model

The probability can be computed by multiplying the probability of transition for each Sbox independently.[1]

[1] In a round, the Sboxes are applied in parallel and thus are independent. In different rounds there should be interactions at the byte level, but these interactions are very hard to take into account. A common approximation made is to suppose all the rounds to be independent.

- Round 0: There are only three Sboxes to consider, the transitions on the other ones being valid with probability 1:
  - The Sboxes in Positions 1 and 2 (reading from left to right, and from top to bottom), having the transitions d $\to$ 2 and d $\to$ 9, are passed using the probability $\mathbb{P}_{\text{DDT}}$, because of Property 1 (1a) (the lower characteristic has a zero difference).
  - The Sbox in Position 9 has a difference in the upper characteristic (input and output), and a difference in the lower characteristic, but the input difference in the lower characteristic is not specified. This is the use case for the UBCT table.

  The transition probability for Round 0 is then
  $$\mathbb{P}_{\text{DDT}}(\text{d}, 2) \cdot \mathbb{P}_{\text{DDT}}(\text{d}, 9) \cdot \mathbb{P}_{\text{UBCT}}(5, 2, 9)$$
- Round 1: There are only four Sboxes to consider:
  - Same as in Round 0, the Sbox in Position 2 is passed with probability $\mathbb{P}_{\text{DDT}}(9, 4)$
  - The Sboxes in Positions 1 and 11 do specify the differences in the input of the upper characteristic and the output of the lower characteristic, but not the other. This is the use case for the BCT table.
  - The Sbox in Position 15 specifies all the differences, this is the use case for the EBCT table.

  The transition probability for Round 1 is
  $$\mathbb{P}_{\text{BCT}}(2, 5) \cdot \mathbb{P}_{\text{DDT}}(9, 4) \cdot \mathbb{P}_{\text{BCT}}(2, 5) \cdot \mathbb{P}_{\text{EBCT}}(2, 5, 9, 4)$$
- Round 2: There are only two Sboxes to consider:
  - The Sbox in Position 2 specifies the differences in the lower characteristic, and in the input of the upper characteristic, it is the use case for the LBCT table.
  - The Sbox in Position 5 specifies the differences in the lower characteristic, and the difference in the upper characteristic is unspecified. By Property 2, the probability is $\mathbb{P}_{\text{DDT}}(5, 2)^2$

  The transition probability for Round 2 is
  $$\mathbb{P}_{\text{LBCT}}(1, 4, 2) \cdot \mathbb{P}_{\text{DDT}}(5, 2)^2$$

The probability of the boomerang characteristic is then

$$\mathbb{P}\left(\Delta_{ex} \overset{E}{\rightleftarrows} \nabla_{ex}\right) = \begin{array}{l} \mathbb{P}_{\text{DDT}}(\text{d}, 2) \cdot \mathbb{P}_{\text{DDT}}(\text{d}, 9) \cdot \mathbb{P}_{\text{UBCT}}(5, 2, 9) \cdot \mathbb{P}_{\text{DDT}}(9, 4) \\ \cdot \mathbb{P}_{\text{BCT}}(2, 5)^2 \cdot \mathbb{P}_{\text{EBCT}}(2, 5, 9, 4) \cdot \mathbb{P}_{\text{LBCT}}(1, 4, 2) \cdot \mathbb{P}_{\text{DDT}}(5, 2)^2 \end{array}$$

To sum up when to use each table, we can give the following rules covering every case:

- $\mathbb{P}_{\text{DDT}}$: difference specified in one characteristic (input and output), and zero in the other characteristic
- $\mathbb{P}_{\text{DDT}}(\cdot, \cdot)^2$: difference specified in one characteristic (input and output), and unspecified difference in the other characteristic
- $\mathbb{P}_{\text{BCT}}$: input difference specified in the upper characteristic, output difference specified in the lower characteristic, and other differences unspecified.

- $\mathbb{P}_{\text{UBCT}}$: input and output difference specified in the upper characteristic, output difference specified in the lower characteristic (and input unspecified)
- $\mathbb{P}_{\text{LBCT}}$: input and output difference specified in the lower characteristic, input difference specified in the upper characteristic (and output unspecified)
- $\mathbb{P}_{\text{EBCT}}$: all differences specified

3.3 Clusters

Recall that for a boomerang distinguisher, the only important differences are the input of the upper characteristic (and upper key), and the output of the lower characteristic (and lower key). The values of the characteristics in all the rounds do not need to be specified at all, and it is simply a lower approximation of the distinguisher probability.

In this section we propose a high-level procedure as a simple method to generate a formula that will give a more precise probability of the distinguisher, without focusing on a single characteristic. It follows the work of Song *et al.* from [SQH19] in a more systematic manner. The main novelties are related to the accuracy of the formula and its computation. More precisely, thanks to the tables introduced Section 2, the only approximation in our formula comes from the assumption that Sboxes are independent. Of course this leads to a more complex formula which may be too costly to compute directly. To overcome this issue, we describe in Section 3.3.2 an algorithm to reorder the terms in the formula to simplify its computation.

3.3.1 Generation of the formula

The procedure takes as input a fixed number of rounds, the input difference of the upper characteristic (and the upper key), and the output difference of the lower characteristic. The procedure is symmetric in the upper and lower characteristic, but the lower characteristic applies the inverse operations (as in the decryption process).

1. Propagates the zeros from the initial differences:[2]
   (a) For the upper characteristic, start from the input difference, and propagates the zeros to the next rounds.
   (b) For the lower characteristic, start from the output difference, and propagates the zeros backward to the previous rounds.
2. Mark the differences: for each Sbox in the upper (resp. lower) characteristic with non-zero difference and such that the lower (resp. upper) difference is non-zero as well, mark the input (resp. output) difference of the upper (resp. lower) characteristic. We do not mark the input difference of the upper characteristic nor the output difference of the lower characteristic because the values are fixed.
3. Mark all the non-zero differences that propagate to a marked difference (recursively). The propagation goes from right to left for the upper characteristic, and from left to right for the lower characteristic.
4. Create a variable for every marked difference in the output (resp. input) of a Sbox for the upper (resp. lower) characteristic.[3]

[2] Equivalent to propagating the differences without allowing cancellations, i.e. with probability 1.

[3] The input (resp. output) differences for the upper (resp. lower) characteristic can be computed from the output (resp. input) differences of the Sboxes after applying the linear part of the cipher. Thus we do not introduce variables for these differences.

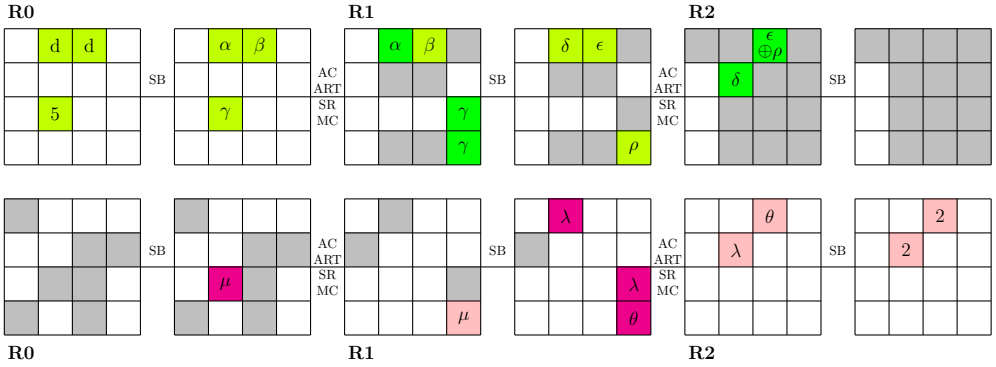

Figure 4: Toy example of boomerang **cluster** on 3 rounds of `SKINNY-64` in the SK model

5. Generate the formula by multiplying the probabilities of transitions for each Sbox using the previous rules, and sum the formula for each variable for each value in $\mathbb{F}_2^n$.

Let us take the same example as before, but this time we will not consider a fixed characteristic but rather the distinguisher.

Example 2. Figure 4 shows a representation of the distinguisher after applying the procedure to generate the formula. Let us follow step by step the procedure to show how to generate the figure.

1. The first step is to propagates the zeros, from the input of the upper characteristic and the output of the lower characteristic. This gives us the blank cells and the gray cells (currently no cells are colored, they are all either white or gray).

2. The second step will mark the cells colored in magenta and green. We do not mark R0 for the upper characteristic nor R2 for the lower characteristic because the values are fixed.

3. The third step will mark all the remaining colored cells in lime and pink. We start with the upper characteristic. To know the values of the two cells at Positions 2 and 5 in R2, we need to know the values at Positions 1, 2, and 15 before SB. Note that the other cells that are involved are already known to be zero. Then, recursively, to know the byte at Position 2, we have to know the one at Position 2 in the previous state. Note that the other ones (those at Positions 1 and 15) are already marked. A similar reasoning allows one to do the propagation regarding the cells at Positions 1, 11, and 15 in R1. Regarding the lower characteristic, we start in R0 with the cell at Position 9. To know its value, we need to know the values of the cells at Positions 7 and 15 in R1 (before the SB transformation). The one at Position 7 is zero, thus we only have to mark in pink the one at Position 15. Similarly, we do the propagation from R1 to R2.

4. The fourth step will create the variables, and apply the linear function to get the input (resp. output) of the Sboxes for the upper (resp. lower) characteristic.

5. The fifth step is simply writing a sum over all the variables, and multiplying each transition probability. The final formula is

$$\mathbb{P}\left(\Delta_{ex} \xrightleftarrows{E} \nabla_{ex}\right) = \sum_{\alpha,\ldots,\mu\in\mathbb{F}_2^n} \begin{array}{l} \mathbb{P}_{\mathrm{DDT}}(\mathrm{d},\alpha) \times \mathbb{P}_{\mathrm{DDT}}(\mathrm{d},\beta) \times \mathbb{P}_{\mathrm{UBCT}}(5,\gamma,\mu)\times \\ \mathbb{P}_{\mathrm{UBCT}}(\alpha,\delta,\lambda) \times \mathbb{P}_{\mathrm{DDT}}(\beta,\epsilon) \times \mathbb{P}_{\mathrm{BCT}}(\gamma,\lambda)\times \\ \mathbb{P}_{\mathrm{EBCT}}(\gamma,\rho,\mu,\theta) \times \mathbb{P}_{\mathrm{LBCT}}(\epsilon\oplus\rho,\theta,2) \times \mathbb{P}_{\mathrm{LBCT}}(\delta,\lambda,2) \end{array} .$$

3.3.2 Automated tool

We implemented in Java the algorithm to automatically compute the probability of a boomerang. This tool generates the formula and computes it (if feasible). It is designed specifically for all versions of `SKINNY`, but is modular and could easily be modified to generate and compute formulas for other ciphers.

The main issue with the accurate formula is the complexity of computing it. For instance, in the formula above, computing $\mathbb{P}\left(\Delta_{ex} \xrightarrow{E} \nabla_{ex}\right)$ requires to sum over 9 variables. While this is reasonable if those variables are nibbles it is not for bytes. To reduce the complexity of the computation we propose two optimizations.

Do not enumerate zeros. Do not enumerate inconsistent values for the variables. For example, there are only few values α such that $\mathbb{P}_{\mathrm{DDT}}(1,\alpha) \neq 0$, and only these values need to be considered.

Reordering the formula. It is possible to split it into two sub-formulas that can be computed recursively. This is very important because the number of variables involved in the sum is really the limiting factor. Let n be the number of variables involved in the sum. The goal is to rewrite the formula as a product of sums so that each of them can be computed independently. Of course the sums should involve as few variables as possible to be easy to compute. Hence the goal is to find the product of sums minimizing the maximum number of variables in each sum. This can be solved using dynamic programming. Let X be the set of all variables involved in the formula and T be a table indexed by all subsets of X. The idea is to store in $T[A]$ the depth of formula assuming variables from $X - A$ are fixed. If $|A| = 1$ then $T[A] = 1$. Assuming T is fulfilled for subsets of size k, let us see how fulfill T for subsets of size $k+1$. Let A be a subset of size $k+1$ and rewrite the formula as

$$\sum_{X} f(X) = \sum_{X-A} \sum_{A} f(X).$$

Now there are two cases. If there exists two non-empty subsets A_1 and A_2 such that $A_1 \cap A_2 = \emptyset$ and $f(X) = f_1(X-A, A_1) \times f_2(X-A, A_2)$, then the formula can be rewritten as

$$\sum_{X-A} \left(\sum_{A_1} f_1(X-A, A_1)\right) \times \left(\sum_{A_2} f_2(X-A, A_2)\right).$$

Thus $T[A] = \max(T[A_1], T[A_2])$. Note that here we do not have to try each possible decomposition. The idea is to start from a variable x of A and to construct the smallest set A_1 containing x such that $f(X) = f_1(X-A, A_1) \times f_2(X-A, A-A_1)$ and take $A_2 = A - A_1$. If such decomposition is impossible (*i.e.* if $A_1 = A$) then we need to find the variable x of A minimizing the complexity of computing the formula as

$$\sum_{X-A} \sum_{x} \sum_{A-x} f(X).$$

Thus in that case $T[A] = 1 + \min_{x \in A}(T[A-x])$. The time complexity of this procedure is upper bounded by $\sum_{k=1}^{n} \binom{n}{k} k = n2^{n-1}$ look-ups in a table of size 2^n. If n is too large to perform this algorithm one can use some heuristics to reorder the formula in the most efficient way possible. But note that for a byte-oriented block ciphers we can hardly compute a formula with a depth greater than 6 and thus n should most often belongs to the practical range of the algorithm. For instance the formula for $\mathbb{P}\left(\Delta_{ex} \xrightarrow{E} \nabla_{ex}\right)$ can

be rewritten with depth 5 as follows:

$$\sum_{\beta,\gamma} \mathbb{P}_{\mathrm{DDT}}(\mathrm{d},\beta) \left(\begin{array}{l} \sum\limits_{\alpha,\delta,\lambda} \mathbb{P}_{\mathrm{DDT}}(\mathrm{d},\alpha) \times \mathbb{P}_{\mathrm{UBCT}}(\alpha,\delta,\lambda) \times \mathbb{P}_{\mathrm{BCT}}(\gamma,\lambda) \times \mathbb{P}_{\mathrm{LBCT}}(\delta,\lambda,2) \\ \times \sum\limits_{\rho,\theta} \left(\begin{array}{l} \sum\limits_{\mu} \mathbb{P}_{\mathrm{UBCT}}(5,\gamma,\mu) \times \mathbb{P}_{\mathrm{EBCT}}(\gamma,\rho,\mu,\theta) \\ \times \left(\sum\limits_{\epsilon} \mathbb{P}_{\mathrm{DDT}}(\beta,\epsilon) \times \mathbb{P}_{\mathrm{LBCT}}(\epsilon \oplus \rho,\theta,2) \right) \end{array} \right) \end{array} \right)$$

Remarks. It may happen that the formula cannot be computed in a reasonable time and in that case we have no other choice than to make approximations as for instance assuming some variables as uniform. Note also the formula is valid only if the characteristics are the same in both sides of the boomerang. Actually when considering the distinguisher (hence only the input and output of the boomerang), the two sides of the boomerang could take completely different differential characteristics, as long as the output difference is the same, the boomerang returned. Supposing that the values on each side of the boomerang could be different is possible, but would suppose to generate tables with up to eight entries (duplicate each entry, and there are already 4 entries in the EBCT), and would also double the number of variables in the formula. These two reasons explains why we did not consider this case in this formula, but we still considered different comebacks in some parts of the distinguisher on particular cases in Section 5.3.

4 MILP Model to Search for Truncated Boomerangs

As for the search of differential characteristics, we chose to split the search of boomerang characteristics into two steps. In the first one we search for *good* truncated boomerang characteristics, and in the second one we search for the best possible instantiations of them. This section is dedicated to the MILP model we created to find truncated boomerang characteristics.

Let E be a classical SPN cipher of R round with an n-cell internal state and such that the round function is composed of a **SubCell** operation, a key addition and a linear layer which multiplies the internal state by a matrix M (at the cell level). Assuming an MILP model to search for truncated (related-key) differential characteristics on this cipher, we show how to turn it into an MILP model to search for truncated boomerang characteristics.

The first part of the model consists of writing twice the MILP model for truncated differential, once for the upper characteristic and once for the lower one. Such models are somehow easy to write and are already available for several block ciphers [ZDY19, BJK+16]. We assume for each cell of each internal state of the upper (resp. lower) characteristic a binary variable isActiveUp (resp. isActiveLo) indicating whether the cell is active or not. In the following we explain which constraints to add to search for *good* truncated boomerang characteristics. Note that we do not define E_0, E_m and E_1, because the upper and lower characteristics will be searched on all the rounds. To represent the fact that some differences will take any value uniformly, we introduce *free* variables (non free variables will be called *controlled* variables). Controlled variables are the differences that will be set to a fixed value in the characteristic.

4.1 Controlled/Free Variables

We introduce two sets of binary variables for each characteristic: isFreeXup and isFreeSBup (resp. isFreeXlo and isFreeSBlo) to indicate whether a difference will be free before and after the Sbox respectively. For the upper characteristic if a difference is free before an Sbox (i.e. isFreeXup = 1), then it is free after the Sbox (isFreeSBup = 1). For the lower

characteristic, if a difference is free after an Sbox, then it is free before the Sbox (because the propagation is done in the opposite direction). This leads to the following constraints:

$$\forall 0 \leq r < R, 0 \leq i < n, \begin{array}{rcl} \mathsf{isFreeSBup}[r][i] & \geq & \mathsf{isFreeXup}[r][i] \\ \mathsf{isFreeXlo}[r][i] & \geq & \mathsf{isFreeSBlo}[r][i] \end{array}$$

Those variables are also related to both isActiveUp and isActiveLo because a difference can be set to 0 only if the difference is controlled. Thus we have the following constraints:

$$\forall 0 \leq r < R, 0 \leq i < n, \begin{array}{rcl} \mathsf{isActiveUp}[r][i] & \geq & \mathsf{isFreeSBup}[r][i] \\ \mathsf{isActiveLo}[r][i] & \geq & \mathsf{isFreeXlo}[r][i] \end{array}$$

Another important constraint is the one stating that a free difference propagates with probability 1 (i.e. no cancellations occur). For the upper characteristic we define $\mathsf{depsU}(i)$ as the set of all the indexes j such that the coefficient $m_{i,j}$ of the matrix M (of the linear layer) is non-zero. For the lower one, we define depsL in a similar way but from the matrix M^{-1}. Then the constraint of propagation of free variables are simply:

$$\forall 0 < r < R, 0 \leq i < n, \begin{array}{rcl} \mathsf{isFreeXup}[r][i] & = & \bigvee_{j \in \mathsf{depsU}(i)} \mathsf{isFreeSBup}[r-1][j] \\ \mathsf{isFreeSBlo}[r-1][i] & = & \bigvee_{j \in \mathsf{depsL}(i)} \mathsf{isFreeXlo}[r][j] \end{array}$$

In order to apply the tables defined Section 2, we need an extra constraint to ensure that the probability of each Sbox can be computed. More precisely, we require that at most 2 variables can be free for each Sbox (considering upper and lower characteristic, before and after the Sbox). This leads to the following constraints:

$$\forall 0 \leq r < R, 0 \leq i < n, \begin{array}{rcl} \mathsf{isFreeSBup}[r][i] + \mathsf{isFreeSBlo}[r][i] & \leq & 1 \\ \mathsf{isFreeXup}[r][i] + \mathsf{isFreeXlo}[r][i] & \leq & 1 \end{array}$$

With all these constraints, the solutions generated will lead to truncated boomerang characteristics. We emphasize with our new set of constraints there is no *middle rounds* defined for our truncated boomerang characteristics. In particular, the BCTs are not necessarily all on the same round but may be spread over several rounds. Thus our modelization is more generic than the previous ones, in particular than the modelization proposed by Liu *et al.* in [LS19].

4.2 Objective Function

The MILP model describes all truncated boomerang characteristics but there are too many of them and it is impossible to exhaust all of them. But we still need to find good truncated characteristics. When searching for truncated differential characteristics, active Sboxes are counted to give a lower bound on the probability of the best possible instantiation of the actual characteristic. The same approach is done here, but because of the multiple different tables, it is not as easy. The important data is the maximum probability of a (non-trivial) transition for each table. As the probabilities are multiplied in the final formula, we consider the exponents in base two of the maximum probabilities so it will be possible to sum all the values. We create one variable per Sbox per table, equal to one if the table is used. These variables can be easily fixed from the previous variables using the rules for deciding when to use each table. More precisely we introduce the binary variables isDDT, isDDT2, isBCT, isUBCT, isLBCT and isEBCT. Then we have the constraints

$$\forall 0 \leq r < R, 0 \leq i < n, \ \mathsf{isDDT}[r][i] + \ldots + \mathsf{isEBCT}[r][i] \leq 1,$$

Table 2: Step 1 objectives in SKINNY. The objectives in parenthesis are upper bounds, the computations were stopped after 1 day on 32 cores.

nb rounds	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
SK	2	6	11	16	23	30	40	51	58	61	66	75	82	88	92	96	-	-	-	-	-	-	-
TK1	0	0	0	2	4	6	10	13	19	27	33	41	47	54	61	70	-	-	-	-	-	-	-
TK2	0	0	0	0	0	0	0	2	4	6	10	13	19	25	31	36	43	51	(59)	(66)	(72)	-	-
TK3	0	0	0	0	0	0	0	0	0	0	0	2	4	6	10	13	19	26	33	40	(47)	(54)	(63)

stating that only one table can be used for an Sbox. Writing the constraints relating those variables to variables isActive and isFree can be a bit tricky. We propose to use the approach used by Abdelkhalek *et al.* in [AST+17]. The idea is to write all impossible cases and to reduce the corresponding set of inequalities using the Quine-McCluskey algorithm. Then, the objective function will be a weighted sum over all the Sboxes and over all the tables, weighted by the maximum probability exponent. For the probability of transition $\mathbb{P}_{DDT}(\cdot,\cdot)^2$, simply multiply by two the weight of the DDT. Let n_T be the $-\log_2$ of the highest probability of a transition in table T. The objective to minimize is then:

$$obj \;=\; \sum_{r,i} n_{DDT}\mathsf{isDDT}[r][i] + \ldots + n_{EBCT}\mathsf{isEBCT}[r][i].$$

4.3 Application to SKINNY

We applied this model on all versions of SKINNY. We used the original MILP model described in [BJK+16] for truncated differential characteristics. Because for SKINNY the maximum probability of a non-trivial transition for the DDT, UBCT, LBCT and EBCT is 2^{-2}, and the maximum probability for the BCT is $1 = 2^0$, we simplified the model to create only two binary variables isTable and isDDT2 for each Sbox. The final objective is the sum of all these variables. The variables isDDT2 is equal to 1 if, and only if the transition probability is $\mathbb{P}_{DDT}(\cdot,\cdot)^2$, and the variable isTable is equal to 1 if, and only if there is a non-zero difference in the upper or lower characteristic through this Sbox, and the table used is *not* the BCT. Indeed, due to the possibility of a probability 1 transition through the BCT, it is not counted in the objective.

Truncated boomerang characteristics were generated for all the versions of SKINNY using the MILP solver Gurobi.[4] For truncated boomerangs, there is no difference between the 64 and 128 bits versions as for each table the optimal probability is the same for both versions. Gurobi allows to either get one optimal solution, or get the N best solutions. Both solvings are important: finding the optimal objective allows to know the number of rounds when the boomerang characteristics have probability lower than 2^{-64} or 2^{-128}; finding the best solutions allows to search for different characteristics in the next step, hence increasing the chances to find a good distinguisher.

Table 2 shows all the optimal objectives for different number of rounds on different attack models. Recall that this objective has to be multiplied by two to get the best boomerang characteristic probability exponent, because of optimizations done in the objective function dedicated to SKINNY.

Remark. It is interesting to note that our model automatically switches to classical truncated differential characteristics after reaching a particular number of rounds. More precisely, when the number of rounds studied is too high, the best boomerang characteristic is the best differential characteristic (the orthogonal difference is 0 for all rounds). For instance we observe on Table 2 that after 13 rounds in the single key model we obtain the exact same bounds than for truncated differential characteristics. This shows how much our model is generic.

[4] https://www.gurobi.com/, we used the parameters Method = 2 and Cuts = 2 for the solving.

4.4 Optimizations

For SPNs, a table showing the minimum number of active Sboxes for a given number of rounds is often available. For example on SKINNY, the table is given by the authors [BJK+16] for up to 30 rounds on the different attacker models (SK, TK1, TK2, and TK3). We can use this table to add non-trivial redundant constraints to the model. Let $tab[r]$ be the minimum number of active Sboxes on a differential characteristic with r rounds. Then we add the constraints:

$$\forall \widetilde{r} \in [1, R], r \in [0, R - \widetilde{r}], \sum_{k=0}^{\widetilde{r}-1} \sum_{i=0}^{15} \mathsf{isActive}[r+k][i] \geq tab[\widetilde{r}]$$

We experimentally noticed that adding those constraints really speeds up the solving process as it is now easier for the solver to prove lower bounds.

There are many minor optimizations that are possible but really dependent on the solver used. For Gurobi, we found out that solving a sub-problem first and then solving the full model really helps in the resolution (helping to know in advance what variables to branch on in the solving). What has been done is to add one constraint forcing to use a BCT table in one cell on Round $R/2$. Solving this problem is really faster because BCTs force to have free variables in the upper and lower characteristics that will propagate further into the rounds. Then we simply remove the constraint and solve the full model.

4.5 Limitations

Even though the model is really simple and can be generalized easily on different ciphers, some issues has been found when testing on SKINNY. The main issue is the fact that it is possible to have truncated boomerangs that differ only on some $free$ variables. These truncated boomerangs are duplicates in the point of view of distinguishers, and thus instantiations will be almost the same. Moreover, when applying the procedure to compute the probability of the boomerang, they will have exactly the same probability because the input and output will be the same.

The second issue that is related to these almost duplicates is the number of solutions which grows exponentially. The tool was configured to find the N best solutions and thus would find non optimal solutions. But as there were too many of them, it was not able to go much further than optimal objective (in a reasonable time limit).

Overcoming the limitations. Some solutions have been found to speed up the computations. The goal here is to add constraints that are always verified in the best solutions. These constraints will be specific to each cipher. On SKINNY, we found out that in the best solutions for both the TK2 and TK3 models, there were always a given number of rounds with only zero differences (in the upper and lower characteristics).

We added those constraints to force some rounds to be zero. To leave some freedom, we did not force the maximum number of rounds to zero, therefore, the solver could choose to add other rounds to zero before or after the ones we fixed. Adding these constraints really speed up the solving, and reduced the number of solutions generated, allowing the model to search for non optimal solutions. Actually, on SKINNY, most of the best boomerang distinguishers given in Table 1 were obtained this way, however the results in Table 2 were obtained without adding constraints to ensure to have a proved optimal objective.

Finally, we found that the most difficult part of the model to solve is when there is no BCT and adding a constraint to ensure that at least one BCT appears in the characteristic improves the solving time by a huge factor (≥ 100). But in that case we would miss some characteristics, in particular the model would not handle truncated differential

characteristics. Still we believe this is a good trade-off between the solving time and the quality of solutions.

5 Instantiation of Truncated Boomerangs

After having generated truncated boomerangs, the next step is to instantiate them to actual values and find the best possible instantiation.

5.1 CP Model

We made a Constraint Programming (CP) model which uses the truncated boomerangs to know which nibbles/bytes have to be fixed to zero, greatly reducing the search space. The model is inspired from [DDH+20] in which Delaune *et al.* describe a CP model to instantiate truncated differential characteristics on `SKINNY`. Their CP model seems much faster than the MILP approach used by Abdelkhalek *et al.* in [AST+17]. To adapt it to boomerang distinguishers we simply had to modify the table constraints representing the transition probability for each Sbox.

The CP solver used is Choco [PFL16] version 4.10.2. It allows to simply define the constraints, and change the search strategy to one that is more adapted to the problem. Using a truncated solution, it is straightforward to know which table (DDT/BCT/...) is used for each Sbox. Our model is very simple as each transition (DDT/BCT/... but also the xor operation) is written with a constraint table, *i.e.* using a table containing all the possible (or impossible) transitions. For instance, describing that $x \oplus y = z$ for binary variables could be done with constraint:

$$(x, y, z) \in \mathrm{Tab}_{\oplus}, \text{ where } \mathrm{Tab}_{\oplus} = \{(0,0,0), (0,1,1), (1,0,1), (1,1,0)\}$$

To get an efficient model, we chose to create a new model for each truncated boomerang characteristic. For each Sbox as well as each xor, we create the corresponding constraint table. An extra variable is added for each probabilistic transition such that every entry of the table is associated to the exponent in base two of the probability. The objective is then the sum of the exponents for each active Sbox.

5.2 Finding Characteristics

To find good characteristics, the solver first finds all the instantiations that have optimal probability and that have different input of upper characteristic, output of lower characteristic, and key differences (meaning that the characteristics comes from different distinguishers). Actually, due to the methods for solving CP models, we first have to find the optimal probability and then fix it and enumerate all the instantiations with this probability. Of course it is not always possible to find this optimal probability in a reasonable time and we added a time limit to the solver.

Then, for each distinguisher found (associated to one truncated boomerang), we can enumerate all the instantiations having the same input and output and key differences, and sum the probabilities of these instantiations. In practice it is not possible to enumerate *all* the instantiations, so we have to give a bound on the number of solutions generated, or the probability up to which we consider the instantiation. Experimentally, we found out that searching for solutions with probability better than $2^{-10} \cdot opt$ was a good compromise between the running time and the final probability raised by the model.

This solving process allowed us to get good boomerang distinguishers for each truncated characteristic. A time limit of ten minutes has been put to limit the computation time, but it was only limiting for some instances of TK3 in 128 bits. Then, to get a better

approximation of the probability, we took the best distinguisher found and analysed it with a more precise cluster analysis.

After finding a good characteristic, a more precise cluster analysis is done on the best characteristic found to have a better approximation of the probability.

5.3 Precise Cluster Analysis

To get a better approximation of the boomerang probability, we then had to apply our tool from Section 3.3.2 to compute the probability of the best distinguisher found by the CP model. In practice, the tool can not compute the probability for more than 6 rounds because of the number of variables created. Hopefully, we were able to use the particular form of the solutions of boomerangs on SKINNY.

Table 3 and Figure 5 both depict the same boomerang characteristic on 18-round SKINNY-128-256 (in the attack model TK2 for which differences are introduced in the two parts of the tweakey). We show here how we were able to compute precisely the probability of this boomerang distinguisher.

We first note that some of the differences are fixed because of the cancellations. This happens for the differences 03 in round R0, 02 in round R2, and 04 in round R6 for the upper characteristic, and differences 90 in round R11 and 21 in round R15 for the lower characteristic.

Next, we observe that middle rounds are defined automatically from the structure of the distinguisher. We do not explain in details here how to compute the probability of the middle rounds R7 to R11 as it was already done in Section 3.3. The only detail is the approximation in round R11 where we supposed that the difference in Position 2 is uniform in the upper characteristic. This approximation has been made because without it the formula was too expensive to compute.

Due to the zero Rounds R3 to R6 for the upper characteristic and R12 to R15 for the lower one, Rounds R0, R1, R2 and Rounds R16 and R17 can be be analyzed as differential characteristics but the probability will be squared (the difference in the other characteristic is uniform, hence we can apply Property 2). Applying the standard formula to compute the probability of the differential characteristics, we obtain:

- **Upper:** $\mathbb{P}_{\mathrm{DDT}}(\mathtt{80},\mathtt{03})^2 \cdot \sum_{\alpha,\beta\in\mathbb{F}_2^n} \mathbb{P}_{\mathrm{DDT}}(\mathtt{01},\alpha)^6 \cdot \mathbb{P}_{\mathrm{DDT}}(\alpha,\beta)^2 \cdot \mathbb{P}_{\mathrm{DDT}}(\beta,\mathtt{02})^2$
- **Lower:** $\sum_{\lambda\in\mathbb{F}_2^n} \mathbb{P}_{\mathrm{DDT}}(\mathtt{21},\lambda)^2 \cdot \mathbb{P}_{\mathrm{DDT}}(\lambda,\mathtt{80})^6$

As explained previously, it is no required to have the same differences in the comeback of the boomerang. This means that for each difference where we define a variable, we now define two variables, one for each side of the boomerang. The positions at which we were able to define two variables for the back and forth of the boomerang are marked with a tilde. The probabilities then becomes:

- **Upper:** $\mathbb{P}_{\mathrm{DDT}}(\mathtt{80},\mathtt{03})^2 \cdot \left(\sum_{\widetilde{\alpha},\widetilde{\beta}\in\mathbb{F}_2^n} \mathbb{P}_{\mathrm{DDT}}(\mathtt{01},\widetilde{\alpha})^3 \cdot \mathbb{P}_{\mathrm{DDT}}(\widetilde{\alpha},\widetilde{\beta}) \cdot \mathbb{P}_{\mathrm{DDT}}(\widetilde{\beta},\mathtt{02})\right)^2$
- **Lower:** $\left(\sum_{\widetilde{\lambda}\in\mathbb{F}_2^n} \mathbb{P}_{\mathrm{DDT}}(\mathtt{21},\widetilde{\lambda}) \cdot \mathbb{P}_{\mathrm{DDT}}(\widetilde{\lambda},\mathtt{80})^3\right)^2$

For Rounds R7 to R11, we apply the precise boomerang probability computation presented in Section 3.3.1. The formula was generated by the automated tool, but as already mentioned we had to approximate the upper characteristic in Round R11 as uniform, otherwise 8 extra variables would appear in the formula making it too expensive

Table 3: Characteristic on 18 rounds of SKINNY-128-256. Values are represented in hexadecimal, greek letters are variables used in the sum, and dashes are unspecified differences. For each round, the two first lines describe the state before and after the **SubCells** operation and the third line describes the round key.

	Upper Trail	Lower Trail
R0	0,80,0,0, 01,0,0,0, 0,0,0,01, 0,0,01,0 0,03,0,0, α,0,0,0, 0,0,0, α, 0,0, α,0 0,03,0,0, 0,0,0,0	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0
R1	0,0,0,0, 0,0,0,0, 0,0,0,0, 0, α,0,0 0,0,0,0, 0,0,0,0, 0,0,0,0, 0, β,0,0 0,0,0,0, 0,0,0,0	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,53,0,0
R2	β,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0 02,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0 02,0,0,0, 0,0,0,0	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0
R3, R4, R5, R6	Only zero differences except R6 subkey $\Delta TK6 =$ 0,0,0,0, 04,0,0,0	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,-
R7	0,0,0,0, 0,0,0,0, 0,04,0,0, 0,0,0,0 0,0,0,0, 0,0,0,0, 0, γ,0,0, 0,0,0,0 0,0,0,0, 0,0,0,0	-,-,0,-, -,-,-,-, 0,-,-,-, 0,0,-,- -,-,0,-, -,-,-,-, 0,ρ,-,-, 0,0,-,- 0,7c,0,0, 0,0,0,0
R8	0,0,0,γ, 0,0,0,0,0, 0,0,0,γ, 0,0,0,γ 0,0,0,δ, 0,0,0,0,0, 0,0,0,ϵ, 0,0,0,- 0,0,0,0, 0,0,0c,0	0,0,-,0, -,-,0,ρ, -,0,-,0, 0,-,0,0 0,0,-,0, -,-,0,ω, -,0,-,0, 0,-,0,0 0,0,0,0, 0,0,0,0
R9	0,-,-,δ, 0,0,0,-, 0,ϵ,0,-, 0,-,0,- 0,-,-,ϕ, 0,0,0,-, 0,θ,0,-, 0,-,0,- 0,0,0,0, 0,0,0,0	-,0,0,0, 0,0,-,0, 0,-,0,0, ω,0,0,0 -,0,0,0, 0,0,-,0, 0,μ,0,0, μ,0,0,0 d8,0,0,0, 0,0,0,0
R10	-,-,-,-, 0,-,-,-, -,-,0,-, 0,-,-,$\phi \oplus \theta$ -,-,-,-, 0,-,-,-, -,-,0,-, 0,-,-,- 0,0,0,0, 0,1c,0,0	0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,μ 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,η 0,0,0,0, 0,0,0,0
R11	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0	0,0,η,0, 0,0,0,0,0, 0,0,0,0, 0,0,0,0 0,0,90,0, 0,0,0,0,0, 0,0,0,0, 0,0,0,0 0,0,90,0, 0,0,0,0
R12, R13, R14, R15	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,-	Only zero differences except R15 subkey $\Delta TK15 =$ 0,0,0,0, 0,0,21,0
R16	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,ff,0,0, 0,0,0,0	0,0,0,0, 0,0,0,0, 0,0,0,21, 0,0,0,0 0,0,0,0, 0,0,0,0, 0,0,0, λ, 0,0,0,0 0,0,0,0, 0,0,0,0
R17	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,-	0, λ,0,0, 0,0,0,0, 0, λ,0,0, 0, λ,0,0 0,80,0,0, 0,0,0,0, 0,80,0,0, 0,80,0,0

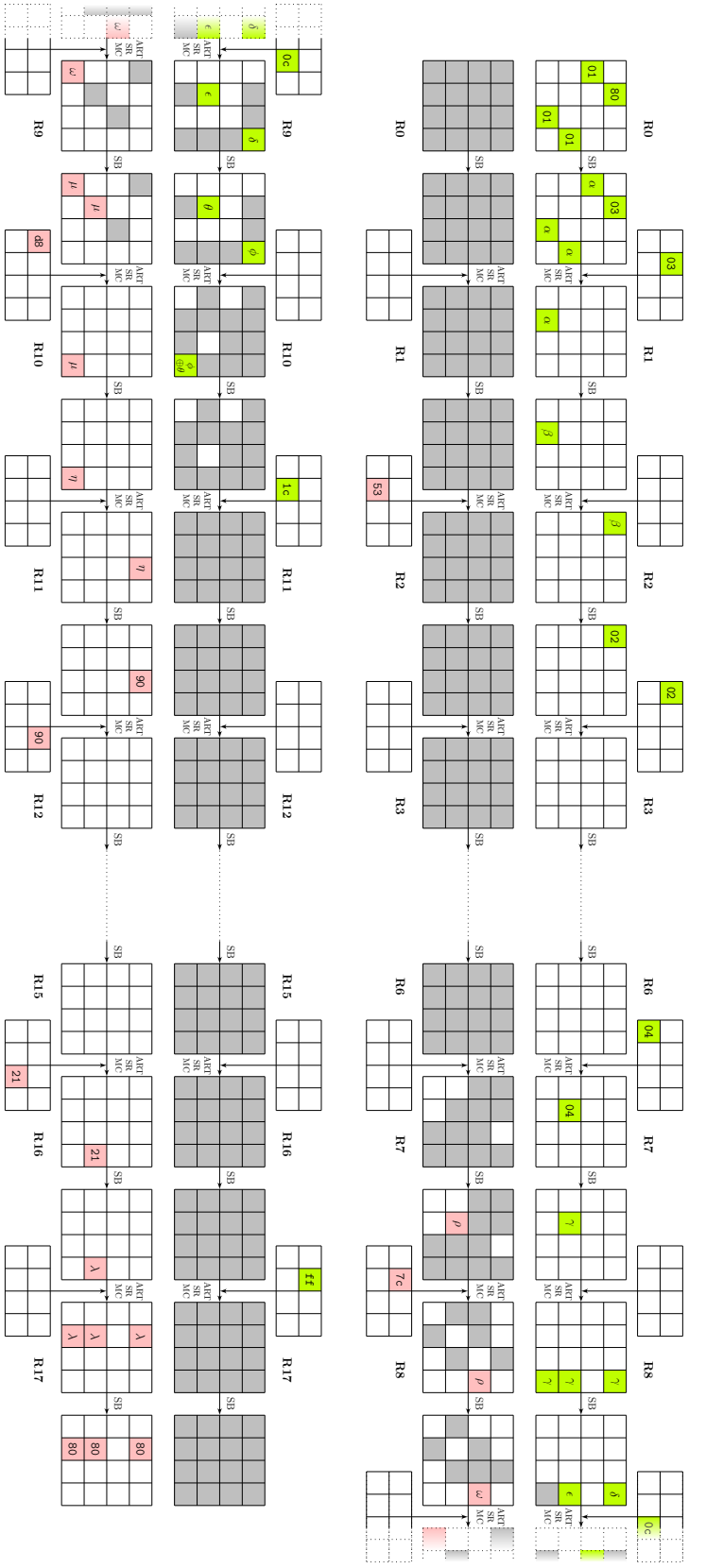

Figure 5: Representation of the distinguisher on 18 rounds of SKINNY-128-256

to compute. The formula (after reorganizing the sum to compute optimally sub-sums) is

$$\sum_{\gamma,\mu}\left(\begin{array}{l}\sum\limits_{\rho,\omega}\mathbb{P}_{\mathrm{UBCT}}(\mathtt{04},\gamma,\rho)\cdot\mathbb{P}_{\mathrm{DDT}}(\rho,\omega)\cdot\mathbb{P}_{\mathrm{DDT}}(\omega,\mu)\\ \times\sum\limits_{\phi}\left(\begin{array}{l}\sum\limits_{\delta}\mathbb{P}_{\mathrm{DDT}}(\gamma,\delta)\cdot\mathbb{P}_{\mathrm{DDT}}(\delta,\phi)\\ \times\sum\limits_{\theta}\left(\begin{array}{l}\sum\limits_{\epsilon}\mathbb{P}_{\mathrm{DDT}}(\gamma,\epsilon)\cdot\mathbb{P}_{\mathrm{UBCT}}(\epsilon,\theta,\mu)\\ \times\sum\limits_{\eta}\mathbb{P}_{\mathrm{LBCT}}(\phi\oplus\theta,\mu,\eta)\cdot\mathbb{P}_{\mathrm{DDT}}(\eta,\mathtt{90})^2\end{array}\right)\end{array}\right)\end{array}\right)$$

The computation of this formula can be fastened by computing the sub-sums independently. Also, the differences are *a priori* in $\mathbb{F}_2^8$, but most of the differences will lead to probability zero because of the impossibility to have some differences propagated through one Sbox. For example, instead of summing over all values $\eta \in \mathbb{F}_2^8$ (256 different values), it is possible to reduce to only the differences that can propagate to the difference $\mathtt{90}$ (meaning that $\mathbb{P}_{\mathrm{DDT}}(\eta, \mathtt{90}) \neq 0$), and there are only 14 such differences, which drastically reduce the number of differences considered. In the end, the time to compute this formula is around 3 minutes.

Remark. Beside values, both the upper and lower characteristics are the same than the ones studied in details in [SQH19]. The only difference is that the upper characteristic is shifted such that the active byte on R7 is on the second column instead of the third. This small change increases the probability of the boomerang by a huge factor of 2^{30}, showing that it is important to search a boomerang as a whole and not as two independent characteristics.

Other characteristics. The same approach was used for all the characteristics given in Table 1 but the higher the number of rounds is, the higher the number of approximations that have to be made in the formula. As a result, all the probabilities for the different versions and number of rounds in `SKINNY` were computed in less than 10 minutes each, except for one that took 30 minutes.

5.4 Experimental Results

To validate our results we experimentally verified the probability of four boomerang distinguishers. For each of them we first randomly pick a key and then, as long as a rightful quartet is not found, randomly pick a plaintext, construct the corresponding quartet and then check whether the boomerang came back. The number of trials required to generate a rightful quartet is then compared to the expected probability of the distinguisher.

Cipher	Runs	Trials (avg)	Probability (expected)
14-round `SKINNY`-64/64 (TK1)	10	$2^{37.46}$	$2^{-40.34}$
17-round `SKINNY`-64/128 (TK2)	1000	$2^{26.43}$	$2^{-27.65}$
18-round `SKINNY`-64/128 (TK2)	10	$2^{37.98}$	$2^{-38.20}$
22-round `SKINNY`-64/192 (TK3)	20	$2^{39.59}$	$2^{-39.44}$

6 Conclusion

In this paper we proposed a new set of tools to search for boomerang characteristics on SPN. In particular, we have shown how to turn an MILP model searching for truncated differential characteristics into an MILP model searching for truncated boomerang characteristics. The main advantage of our method is that the middle rounds of the boomerang do not have to be specified and are automatically handled by the model. We have also shown

that a CP model can be used to find the best possible instantiations of a given truncated boomerang characteristic. Finally we systematized the method developed by Song *et al.* in [SQH19] to precisely compute the probability of a boomerang.

As a result, we applied our new tools to the block cipher `SKINNY` and found many new boomerang distinguishers up to 24 rounds in the TK3 model. In particular, we improved by a factor 2^{30} the probability of the best known distinguisher against 18-round `SKINNY`-128/256.

References

[AST+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP modeling for (large) s-boxes to optimize probability of differential characteristics. *IACR Trans. Symmetric Cryptol.*, 2017(4):99–129, 2017.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999.

[BHL+20] Hamid Boukerrou, Paul Huynh, Virginie Lallemand, Bimal Mandal, and Marine Minier. On the feistel counterpart of the boomerang connectivity table introduction and analysis of the FBCT. *IACR Trans. Symmetric Cryptol.*, 2020(1):331–362, 2020.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

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

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

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

[CHP+18] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 683–714. Springer, 2018.

[DDH+20] Stéphanie Delaune, Patrick Derbez, Paul Huynh, Marine Minier, Victor Mollimard, and Charles Prud'homme. Skinny with scalpel - comparing tools for differential analysis. Cryptology ePrint Archive, Report 2020/1402, 2020. https://eprint.iacr.org/2020/1402.

[DKK06] Orr Dunkelman, Nathan Keller, and Jongsung Kim. Related-key rectangle attack on the full SHACAL-1. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers*, volume 4356 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2006.

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

[Dun18] Orr Dunkelman. Efficient construction of the boomerang connection table. *IACR Cryptol. ePrint Arch.*, 2018:631, 2018.

[Kir15] Aleksandar Kircanski. Analysis of boomerang differential trails via a sat-based constraint solver URSA. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers*, volume 9092 of *Lecture Notes in Computer Science*, pages 331–349. Springer, 2015.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song. Security analysis of SKINNY under related-tweakey settings (long paper). *IACR Trans. Symmetric Cryptol.*, 2017(3):37–72, 2017.

[LS19] Yunwen Liu and Yu Sasaki. Related-key boomerang attacks on GIFT with automated trail search including BCT effect. In Julian Jang-Jaccard and Fuchun Guo, editors, *Information Security and Privacy - 24th Australasian Conference, ACISP 2019, Christchurch, New Zealand, July 3-5, 2019, Proceedings*, volume 11547 of *Lecture Notes in Computer Science*, pages 555–572. Springer, 2019.

[Lu09] Jiqiang Lu. Related-key rectangle attack on 36 rounds of the XTEA block cipher. *Int. J. Inf. Sec.*, 8(1):1–11, 2009.

[Mur11] Sean Murphy. The return of the cryptographic boomerang. *IEEE Trans. Inf. Theory*, 57(4):2517–2521, 2011.

[PFL16] Charles Prud'homme, Jean-Guillaume Fages, and Xavier Lorca. *Choco Solver Documentation*. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S., 2016.

[SQH19] Ling Song, Xianrui Qin, and Lei Hu. Boomerang connectivity table revisited. application to SKINNY and AES. *IACR Trans. Symmetric Cryptol.*, 2019(1):118–141, 2019.

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

[WKD07] Gaoli Wang, Nathan Keller, and Orr Dunkelman. The delicate issues of addition with respect to XOR differences. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *Selected Areas in Cryptography, 14th International Workshop,*

SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers, volume 4876 of *Lecture Notes in Computer Science*, pages 212–231. Springer, 2007.

[WP19] Haoyang Wang and Thomas Peyrin. Boomerang switch in multiple rounds. application to AES variants and deoxys. *IACR Trans. Symmetric Cryptol.*, 2019(1):142–169, 2019.

[ZDY19] Baoyu Zhu, Xiaoyang Dong, and Hongbo Yu. Milp-based differential attack on round-reduced GIFT. In Mitsuru Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 372–390. Springer, 2019.

A Dependence sets of SKINNY

The constraints obtained from the MixColumn of SKINNY are:

$$
\begin{aligned}
depsU(0,j) =& \{(0,j),(2,(j+2)\%4),(3,(j+1)\%4)\}\\
depsU(1,j) =& \{(0,j)\}\\
depsU(2,j) =& \{(1,(j+3)\%4),(2,(j+2)\%4)\}\\
depsU(3,j) =& \{(0,j),(2,(j+2)\%4)\}\\
depsL(0,j) =& \{(1,j)\}\\
depsL(1,j) =& \{(1,(j+1)\%4),(2,(j+1)\%4),(3,(j+1)\%4)\}\\
depsL(2,j) =& \{(1,(j+2)\%4),(3,(j+2)\%4)\}\\
depsL(3,j) =& \{(0,(j+3)\%4),(3,(j+3)\%4)\}
\end{aligned}
$$

B Boomerang characteristic on TK3 24 rounds 128 bits

Table 4 shows the distinguisher found on 24 rounds of SKINNY 128 bits, under TK3.

To be able to compute a probability, some approximation have been made. Ideally we would not do any approximation and consider a middle round composed of the rounds R9 to R15, but too many variables would be introduced and the final formula could not be computed. Three types of approximations have been made:

- Fixing a value of a cell: the value of cell 14 of the lower characteristic in round R14 have been fixed to `0x01`, and cell 15 of round R10 in the lower characteristic have been fixed to `0x10`.

- Supposing the uniformity of some cells: for the upper characteristic cells 8 of R13, 14 of round R14 and 1 of round R15 have been supposed to take uniform values. For the lower characteristic cells 9 of round R9 and cells 3 and 10 of round R10 have been supposed to take uniform values.

- Forcing cancellations: To enforce cancellations (forcing zeros in some rounds), some cells have been fixed to the same value. This is the case for cells 3 and 9 of round R11 of the upper characteristic (θ), and cells 7 and 15 of round R13 of the lower characteristic (υ).

Table 4: Characteristic on 24 rounds of SKINNY-128-384. Values are represented in hexadecimal, greek letters are variables used in the sum, and dashes are unspecified differences.

	Upper Trail	Lower Trail
R0	0,0,0,0, 0,10,0,20, 10,0,0,0, 0,0,0,10 0,0,0,0, 0, α ,0,83, α ,0,0,0, 0,0,0, α 0,0,0,0, 0, 0 ,0,83	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0
R1	0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0, α ,0 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0, β ,0 0,0,0,0, 0,0,0,0	- ,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- - ,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 68,0,0,0, 0,0,0,0
R2	0, β ,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0 0,01,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0 0,01,0,0, 0,0,0,0	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0
R3, R4, R5 R6, R7, R8	Only zero differences except R8 subkey $\Delta TK8 =$ 01,0,0,0, 0,0,0,0	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,-
R9	0,0,0,0, 0,0,0,0, 0,04,0,0, 0,0,0,0 0,0,0,0, 0,0,0,0, 0, γ ,0,0, 0,0,0,0 0,0,0,0, 0,0,0,0	-,-,0,-, -,-,-,-,-,-,-,-, -,-,-,- -,-,0,-, -,-,-,-,-,-,-,-, -,-,-,- 0,0,0,0, 0,f6,0,0
R10	0,0,0, γ, 0,0, 0 ,0, 0,0,0, γ, 0,0,0, γ 0,0,0, δ, 0,0, 0 ,0, 0,0,0, ϵ, 0,0,0,- 0,0,0,0, 0,0,83,0	0,0,-,0, -,-,0,-, -,0,-,-, -,-,-, - 0,0,-,0, -,-,0,-, -,0,-,-, -,-,-,10 0,0,0,0, 0,0,0,0
R11	0, ϵ,-, δ, 0,0,0,-, 0, ϵ,0,83, 0, ϵ,0,- 0, η,-, θ, 0,0,0,-, 0, θ,0, κ , 0,-,0,- 0,0,0,0, 0,0,0,0	-,0,0, $-$, 0,0,-,0, 0,-,0,0, -,$-$,10,0 -,0,0, $\sigma \oplus e$3, 0,0,-,0, 0, σ,0,0, -, χ, ϕ ,0 0,0,0, e3, 0,0,0,0
R12	-, $\eta \oplus \kappa$,-,0, 0, - ,-, θ, -, κ,0, θ, 0,-,-,0 -, λ,-,0, 0, - ,-,-,-, -, μ,0,-, 0,-,-,0 0, 0,0,0, 0,c6,0,0	0, ϕ,0,-, χ,0,0, σ, 0,0,0,-, ψ,0,0,0 0, ρ,0,-, ρ,0,0, τ, 0,0,0, ρ, υ,0,0,0 0,0,0,0, 0,0,0,0
R13	-,-,-,-, -, λ,-,0, -,-,-,-, -,-,-, μ -,-,-,-, -,-,-,0, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0	0,0,0,0, 0, ρ ,0, υ , τ ,0,0,0, 0,0,0, υ 0,0,0,0, 0,01,0,c1, 01,0,0,0, 0,0,0,01 0,0,0,0, 0, 0 ,0,c1
R14	-,-,-, - , -,-,-,-, -,-,-,-, -,-,-,- -,-,-, - , -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,ec, 0,0,0,0	0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,01,0 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0, ν ,0 0,0,0,0, 0,0,0,0
R15	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,0,0, 0,0,0,0	0, ν ,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0 0,80,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0 0,80,0,0, 0,0,0,0
R16, R17, R18 R19, R20, R21	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,-	Only zero differences except R21 subkey $\Delta TK21 =$ 0,0,0,0, 80,0,0,0
R22	-,-, - ,-, -,-,-,-, -,-,-,-, -,-,-,- -,-, - ,-, -,-,-,-, -,-,-,-, -,-,-,- 0,0,d1,0, 0,0,0,0	0,0,0,0, 0,0,0,0, 0,80,0,0, 0,0,0,0 0,0,0,0, 0,0,0,0, 0, ω ,0,0, 0,0,0,0 0,0,0,0, 0,0,0,0
R23	-,-,-,-, -,-,-,-, -,-,-,-, -,-,-,- -,-,-,-, -,-,-,-, -,-,-,-, -,-,-,-	0,0,0, ω , 0,0,0,0, 0,0,0, ω , 0,0,0, ω 0,0,0,08, 0,0,0,0, 0,0,0,08, 0,0,0,08

C Proof of Property 1

In this section we detail the proof for some of the properties stated in Section 2.2.

1. Properties on the EBCT

   (a) Let $\alpha, \gamma, \theta \in \mathbb{F}_2^n$. We only prove $\mathbb{1}_{\alpha=0} \cdot \mathrm{DDT}(\gamma, \theta) = \mathrm{EBCT}(\alpha, 0, \gamma, \theta)$ as the other properties can be proved in a similar way. Any x satisfying the constraints of $\mathrm{EBCT}(\alpha, 0, \gamma, \theta)$ should satisfy $S(x) \oplus S(x \oplus \alpha) = 0$, i.e. $x \oplus \alpha = x$ (because the Sbox is bijective), which is impossible if $\alpha \neq 0$. Therefore we have that $\mathrm{EBCT}(\alpha, 0, \gamma, \theta) = 0$ if $\alpha \neq 0$. Now, if $\alpha = 0$, then the definition of the EBCT becomes

$$\begin{aligned}
\mathrm{EBCT}(0, 0, \gamma, \theta) &= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus 0) = 0 \\ S(x) \oplus S(x \oplus \gamma) = \theta \\ S^{-1}(S(x) \oplus \theta) \oplus S^{-1}(S(x \oplus 0) \oplus \theta) = 0 \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S^{-1}(S(x) \oplus \theta) = S^{-1}(S(x) \oplus \theta) \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;|S(x) \oplus S(x \oplus \gamma) = \theta \right\} \\
&= \mathrm{DDT}(\gamma, \theta)
\end{aligned}$$

   (b) Let $\alpha, \beta, \delta \in \mathbb{F}_2^n$. Any x satisfying the constraints of the $\mathrm{EBCT}(\alpha, \delta, \beta, \delta)$ should satisfy the two constraints $S(x) \oplus S(x \oplus \alpha) = \delta$ and $S(x) \oplus S(x \oplus \beta) = \delta$, which can be satisfied if, and only if, $\alpha = x \oplus S^{-1}(S(x) \oplus \delta) = \beta$. Therefore, we have that $\mathrm{EBCT}(\alpha, \delta, \beta, \delta) = 0$ if $\alpha \neq \beta$. Now if $\alpha = \beta$,

$$\begin{aligned}
\mathrm{EBCT}(\alpha, \delta, \beta, \delta) &= \mathrm{EBCT}(\alpha, \delta, \alpha, \delta) \\
&= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \alpha) = \delta \\ S(x) \oplus S(x \oplus \alpha) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \alpha) \oplus \delta) = \alpha \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \alpha) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \alpha) \oplus \delta) = \alpha \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \alpha) = \delta \\ S^{-1}(S(x \oplus \alpha)) \oplus S^{-1}(S(x \oplus \alpha) \oplus \delta) = \alpha \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \alpha) = \delta \\ S^{-1}(S(x \oplus \alpha) \oplus \delta) = x \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \alpha) = \delta \\ S(x \oplus \alpha) \oplus \delta = S(x) \end{array} \right\} \\
&= \#\left\{ x \in \mathbb{F}_2^n \;|S(x) \oplus S(x \oplus \alpha) = \delta \right\} \\
&= \mathrm{DDT}(\alpha, \delta)
\end{aligned}$$

2. Properties on the UBCT

   (a) Let $\gamma, \theta, \delta \in \mathbb{F}_2^n$. Let's note, for any $\lambda \in \mathbb{F}_2^n$,

$$\mathcal{X}_{\mathrm{EBCT}}(\gamma, \theta, \lambda, \delta) = \left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S(x) \oplus S(x \oplus \lambda) = \delta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\}$$

and

$$\mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta) = \left\{ x \in \mathbb{F}_2^n \;\middle|\; \begin{array}{l} S(x) \oplus S(x \oplus \gamma) = \theta \\ S^{-1}(S(x) \oplus \delta) \oplus S^{-1}(S(x \oplus \gamma) \oplus \delta) = \gamma \end{array} \right\}$$

For any $\lambda_1, \lambda_2 \in \mathbb{F}_2^n$ distinct, the sets $\mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda_1,\delta)$ and $\mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda_2,\delta)$ are disjoint (because of the constraint $S(x)\oplus S(x\oplus\lambda) = \delta$, equivalent to $\lambda = x \oplus S^{-1}(S(x) \oplus \delta)$). We now prove that $\mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta) = \bigcup_{\lambda\in\mathbb{F}_2^n} \mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta)$.

- First, remark that $\forall\lambda \in \mathbb{F}_2^n$, $\mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta) \subseteq \mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta)$, then $\bigcup_{\lambda\in\mathbb{F}_2^n} \mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta) \subseteq \mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta)$.
- Let $x \in \mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta)$, let $\lambda = x\oplus S^{-1}(S(x)\oplus\delta)$, then $x \in \mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta)$. This proves $\mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta) \subseteq \bigcup_{\lambda\in\mathbb{F}_2^n} \mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta)$.

Now we are able to finish the proof since the sets presented here are disjoint. Hence, we are able to sum their sizes.

$$\begin{aligned} \mathrm{UBCT}(\gamma,\theta,\delta) &= \#\mathcal{X}_{\mathrm{UBCT}}(\gamma,\theta,\delta) \\ &= \#\left(\bigcup_{\lambda\in\mathbb{F}_2^n} \mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta)\right) \\ &= \sum_{\lambda\in\mathbb{F}_2^n} \#\mathcal{X}_{\mathrm{EBCT}}(\gamma,\theta,\lambda,\delta) \\ &= \sum_{\lambda\in\mathbb{F}_2^n} \mathrm{EBCT}(\gamma,\theta,\lambda,\delta) \end{aligned}$$