Truncated Boomerang Attacks and Application to AES-based Ciphers

Augustin Bariant, Gaëtan Leurent

Inria, Paris, France

Abstract. The boomerang attack is a cryptanalysis technique that combines two short differentials instead of using a single long differential. It has been applied to many primitives, and results in the best known attacks against several AES-based ciphers (Kiasu-BC, Deoxys-BC). In this paper, we introduce a general framework for boomerang attacks with truncated differentials.

We show that the use of truncated differentials provides a significant improvement over the best boomerang attacks in the literature. In particular, we take into account structures on the plaintext and ciphertext sides, and include an analysis of the key recovery step. On 6-round AES, we obtain a competitive structural distinguisher with complexity 2^{87} and a key recovery attack with complexity 2^{61}.

The truncated boomerang attack is particularly effective against tweakable AES variants. We apply it to 8-round Kiasu-BC, resulting in the best known attack with complexity 2^{83} (rather than 2^{103}). We also show an interesting use of the 6-round distinguisher on the full TNT-AES, a tweakable block-cipher using 6-round AES as a building block. Finally, we apply this framework to Deoxys-BC, using a MILP model to find optimal trails automatically. We obtain the best attacks against round-reduced versions of all variants of Deoxys-BC.

Keywords: Truncated differential · Boomerang attack · AES · Kiasu · Deoxys · TNT-AES · MILP

1 Introduction

The AES [15] is the most widely used block cipher today, and we have a good understanding of its security. Its round function is strongly byte-aligned; this simplifies the analysis with the wide-trail strategy, and many cryptanalysis techniques rely on truncated trails to take advantage of this property. After 20 years of analysis, we have a high confidence in the design, and many recent tweakable proposals reuse the AES round function with different tweakey schedules (Kiasu-BC [26], Deoxys-BC [28], and TNT-AES [1]). However, an attacker can introduce a difference in the additional tweak of these constructions, so they must be analysed in the related-tweak model, and by extension in the related-key model. In these models, the boomerang attack is particularly effective because both high-probability differentials composing the boomerang reach more rounds.

In particular, the best known attacks against Kiasu-BC and Deoxys-BC are boomerang attacks.

In this work, we carefully and systematically analyse the interaction between truncated differentials and boomerang attacks. Our approach is similar to the analysis of impossible differential attacks in [11]: we aim at providing a unified formula taking into account many details of a broad class of attacks. We integrate and improve a set of techniques proposed in different variants of the boomerang attack, leading to the best boomerang attacks against several AES-based ciphers.

Our results. We present a generic framework to describe boomerang attacks based on truncated differentials (section 3). Instead of first building a boomerang distinguisher and then appending extra key recovery rounds, we consider the truncated boomerang attack as a whole, including the key recovery exploiting the first and last round transitions. The framework integrates and improves on previous analyses, including structures of plaintexts and ciphertexts [9], and truncated differentials as introduced by Wagner [39]. Our improvements come principally from the use of structures of ciphertexts, which were underused in recent works. We also consider boomerang trails with smaller probability than the random case, and mount attacks by gathering enough samples to detect the bias.

We first apply our framework to reduced AES (section 4). On 6-round AES, we obtain a distinguisher with complexity 2^{87}, and a key-recovery with complexity 2^{61}, improving the previous best boomerang attack with complexity 2^{71} [10].

We adapt the key-recovery attack to 8-round Kiasu-BC (section 5) by revisiting a previous boomerang attack with complexity 2^{103} [18]. Using structures of ciphertexts, we obtain the best attack against Kiasu-BC, with complexity 2^{83}.

We also apply a variant of the 6-round attack to the full TNT-AES [1], and obtain a marginal distinguisher with complexity slightly below 2^{128} (section 6). The attack is not competitive with the generic attack against TNT with complexity $\mathcal{O}(\sqrt{n} \cdot 2^{3n/4})$ [25], but it uses a lower memory (2^{32} instead of 2^{96}), and it can distinguish TNT with 6-round AES from TNT with a PRP. Moreover, this is the first property of 6-round AES that can be used to target a generic construction using 6-round AES as a building block (to the best of our knowledge). We also provide an attack on reduced TNT-AES, using a 5-round boomerang trail.

In section 7, we build a MILP model implementing our framework, to find good parameters for the full attack automatically. The model allows both fixed differences and truncated differences, and takes into account the complexity of the key recovery, instead of just optimizing a boomerang distinguisher. It confirms that our basic attack on AES is optimal within our framework.

Finally, we apply the MILP model to Deoxys-BC, and obtain improved attacks against most variants (section 8). Although the boomerang trails on AES and Deoxys-BC are quite different, the underlying analysis is the same.

Due to space constraints, some results are only available in the full version of this paper [6]. Our code is also available as additional data [7].

Distinguishers and key-recovery attacks. In this work, we report dinstiguishers and key-recovery attacks, with key-recovery typically having a lower complexity on the same number of rounds. Obviously, a key-recovery attack can be used as a distinguisher, but we focus on structural distinguishers that only use statistical properties of the block cipher, without guessing subkey material (denoted as "independent of the secret key" in [24]). Indeed, a series of recent works have proposed complex distinguishers on 5-round [24] and 6-round AES [2,5,33], and we obtain similar results with simpler techniques. This notion of distinguisher is not clearly defined, but our distiguishers can be used with secret S-Boxes, which is not the case for most key-recovery attacks.

2 Preliminaries

2.1 The AES Round Function

AES was designed in 1998 by Daemen and Rijmen (as Rijndael) and won the NIST standardization competition in 2000 [15]. Three instances of the cipher exist, for key sizes of 128, 196 and 256 bits, but we only consider AES-128 in this paper. Since we do not exploit the AES key schedule, we only describe the round function. AES-128 operates on a 128-bit state, represented as a 4×4-byte array, and iterates on 10 rounds a round function composed of the following operations:

- **SubBytes**: The AES S-Box is applied to each byte of the state.
- **ShiftRows**: The second row is shifted by 1 cell to the left, the third row by 2 cells, and the fourth row by 3 cells.
- **MixColumns**: Each column is multiplied by an MDS Matrix.
- **AddRoundKey**: Each byte is XORed with a byte of the round key.

There is one extra AddRoundKey operation before the first round, and the last round omits the MixColumns operation.

Due to the popularity of the AES, and its availability in hardware on several platforms, many constructions reuse its round function. In particular, Kiasu [27] and Deoxys [28] are two tweakable block ciphers that reuse the AES round function, with a modified tweakey schedule (combining the key and tweak) to compute the round (tweak)keys. Deoxys has been selected in the CAESAR portfolio. TNT-AES [1] is another tweakable block cipher using the AES round function, where the tweak is only XORed to the internal state twice.

Kiasu-BC tweakey schedule. Kiasu-BC has a 128-bit key and 64-bit tweak, with 10 rounds. The round tweakeys are computed as $k_i + t$ where k_i is the round key following the AES key schedule, and t is the tweak (encoded in the first two rows). In particular, Kiasu-BC with the zero tweak is the same as the AES.

Deoxys-BC tweakey schedule. Deoxys-BC has two variants: Deoxys-BC-256 has a 256-bit tweakey with 14 rounds, and Deoxys-BC-384 has a 384-bit tweakey with 16 rounds. The tweakey material is composed of a variable length key and tweak

Table 1. AES distinguisher and key recovery attacks with known and secret S-Boxes. CP: chosen plaintexts / ACC: chosen plaintexts and adaptively-chosen ciphertexts

	Rounds	Type	Data		Time	Ref
AES Distinguishers	5	Multiple-of-n	2^{32}	CP	$2^{36.6}$	[24]
	6	Yoyo	$2^{122.8}$	ACC	$2^{121.8}$	[33]
	6	Exchange attack	$2^{88.2}$	CP	$2^{88.2}$	[5]
	6	Exchange attack	2^{84}	ACC	2^{83}	[4]
	6	Truncated differential	$2^{89.4}$	CP	$2^{96.5}$	[2]
	6	Truncated boomerang	2^{87}	ACC	2^{87}	subsection 4.1
AES Key-recovery	6	Square	2^{32}	CP	2^{71}	[14]
	6	Partial-sum	2^{32}	CP	2^{48}	[21]
	6	Boomerang	2^{71}	ACC	2^{71}	[10]
	6	Mixture	2^{26}	CP	2^{80}	[3]
	6	Retracing boomerang	2^{55}	ACC	2^{80}	[19]
	6	Boomeyong	$2^{79.7}$	ACC	2^{78}	[32]
	6	Truncated boomerang	2^{59}	ACC	2^{61}	subsection 4.2
AES Secret S-Box KR	5	Square	2^{40}	CP	2^{40}	[37]
	5	Multiple-of-n	$2^{53.3}$	CP	$2^{52.6}$	[22]
	5	Retracing boomerang	$2^{25.8}$	ACC	$2^{25.8}$	[19]
	6	Square	2^{64}	CP	2^{90}	[37]
	6	Truncated boomerang	2^{94}	ACC	2^{94}	subsection 4.3

Table 2. Boomerang (B) and rectangle (R) attacks against variants of Deoxys-BC. Most attacks succeed with probability 1/2.

			Previous					New			
Model	Rnd		Data	Time	Mem	Ref		Data	Time	Mem	Ref
RTK1	8						B	2^{88}	2^{88}	2^{73}	*Full version* [6]
	9						B	2^{135}	2^{174}	2^{129}	*Full version* [6]
RTK2	8	B	2^{28}	2^{28}	2^{27}	[34][a]	B	2^{27}	2^{27}	2^{27}	*Full version* [6]
	9	B	2^{98}	2^{112}	2^{17}	[34]	B	$2^{55.2}$	$2^{55.2}$	$2^{55.2}$	*Full version* [6]
	10	B	$2^{98.4}$	$2^{109.1}$	2^{88}	[42]	B	$2^{94.2}$	$2^{95.2}$	$2^{94.2}$	section 8
	11	R	$2^{122.1}$	$2^{249.9}$	$2^{128.2}$	[42]	B	2^{129}	$2^{223.9}$	2^{129}	section 8
RTK3	10	B	2^{22}	2^{22}	2^{17}	[34]	B	$2^{19.4}$	$2^{19.4}$	2^{18}	*Full version* [6]
	11	B	2^{100}	2^{100}	2^{17}	[34]	B	$2^{32.7}$	$2^{32.7}$	$2^{32.7}$	*Full version* [6]
	12	B	2^{98}	2^{98}	2^{64}	[42]	B	$2^{67.4}$	$2^{67.4}$	2^{65}	*Full version* [6]
	13	R	$2^{125.2}$	$2^{186.7}$	2^{136}	[43]	B	$2^{126.7}$	$2^{170.2}$	$2^{126.7}$	section 8
	14	R	$2^{125.2}$	$2^{282.7}$	2^{136}	[43]	B	2^{129}	$2^{278.8}$	2^{129}	*Full version* [6]

[a] The probability of Sasaki's trail is 2^{-56} with structures, thus we believe that the complexity of the attack is actually 2^{30} in data and time and 2^{29} in memory.

Table 3. Attacks against Kiasu-BC and TNT-AES

	Rounds	Type	Data		Time	Ref
Kiasu-BC	7	Square (KR)	$2^{43.6}$	CP	$2^{48.5}$	[17]
	8	Meet-in-the-Middle (KR)	2^{116}	CP	2^{116}	[38]
	8	Imposs. Diff (KR)	2^{118}	CP	2^{118}	[18]
	8	Boomerang (KR)	2^{103}	ACC	$2^{103.1}$	[18]
	8	Truncated boomerang (KR)	2^{83}	ACC	2^{83}	section 5
TNT-AES	$*$-5-$*$	Boomerang (dist.)	2^{126}	ACC	2^{126}	[1]
	5-$*$-$*$	Impossible differential (KR)	$2^{113.6}$	CP	$2^{113.6}$	[25]
	$*$-$*$-$*$	Generic (dist.)	$2^{99.5}$	CP	$2^{99.5}$	[25]
	$*$-5-$*$	Truncated boomerang (dist.)	2^{76}	ACC	2^{76}	*Full version* [6]
	5-5-$*$	Truncated boomerang (KR)	2^{87}	ACC	2^{87}	*Full version* [6]
	$*$-6-$*$	Truncated boomerang (dist.)	$2^{127.8}$	ACC	$2^{127.8}$	section 6

summing to 256 or 384 bits; for simplicity, we assume that the key length is a multiple of 128. The tweakey material is divided in words of 128 bits (denoted TK^i). Eventually, the round tweakey of round j is defined as:

$$STK_j = \begin{cases} RC_j + TK^1_j + TK^2_j & \text{For Deoxys-BC-256} \\ RC_j + TK^1_j + TK^2_j + TK^3_j & \text{For Deoxys-BC-384} \end{cases}$$

TK^i_j is the tweakey state, initialized as $TK^i_0 = TK^i$ and updated with

$$TK^1_{j+1} = h(TK^1_j) \quad TK^2_{j+1} = h(\mathsf{LFSR}_2(TK^2_j)) \quad TK^3_{j+1} = h(\mathsf{LFSR}_3(TK^3_j))$$

where h is a byte permutation, and LFSR_2 and LFSR_3 are LFSRs that operate in parallel on each byte of the tweakey. This construction (the STK construction [27]) ensures that differences in subtweakey byte position may only cancel out up to $i-1$ times every 15 rounds if differences are introduced in i tweakey words.

Notations. We denote E a block cipher operating on a state of n bits. In a 4×4 matrix, the bytes are numbered in the AES order (column-major). When k_i is a sub(twea)key, we denote $k_i^{eq} = \mathsf{MixColumns}^{-1}(k_i)$.

2.2 Differentials and Truncated Differentials

We use $+$ to denote the XOR operation (the addition in $\mathbb{F}_{2^u}^v$). A differential is defined by an input difference $\varDelta_{\text{in}} \in \{0,1\}^n$ and an output difference $\varDelta_{\text{out}} \in \{0,1\}^n$. We use the notation $\varDelta_{\text{in}} \xrightarrow[E]{p} \varDelta_{\text{out}}$ when a differential exists with probability p, where the probability is defined over a random plaintext P:

$$p = \Pr[\varDelta_{\text{in}} \xrightarrow[E]{} \varDelta_{\text{out}}] = \Pr\left[E(P) + E(P + \varDelta_{\text{in}}) = \varDelta_{\text{out}}\right]$$

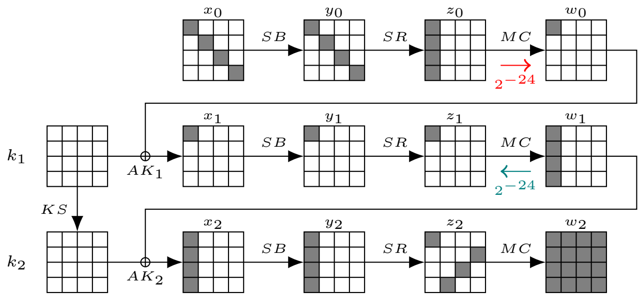


Fig. 1. Example of a truncated differential trail on 3-round AES.

Since E is a permutation, we have $\Pr[\Delta_{\text{in}} \xrightarrow[E]{} \Delta_{\text{out}}] = \Pr[\Delta_{\text{out}} \xrightarrow[E^{-1}]{} \Delta_{\text{in}}]$.

A truncated differential is defined by a set of input differences $\mathcal{D}_{\text{in}}$ and a set of output differences $\mathcal{D}_{\text{out}}$. We use the notation $\mathcal{D}_{\text{in}} \xrightarrow[E]{p} \mathcal{D}_{\text{out}}$ to denote the existence of a truncated differential with probability $\vec{p}$, defined as (with Avg denoting the average):

$$\vec{p} = \operatorname*{Avg}_{\Delta_{\text{in}} \in \mathcal{D}_{\text{in}}} \Pr\left[E(P) + E(P + \Delta_{\text{in}}) \in \mathcal{D}_{\text{out}}\right]$$

We also define the probability of the reverse truncated differential as

$$\overleftarrow{p} = \operatorname*{Avg}_{\Delta_{\text{out}} \in \mathcal{D}_{\text{out}}} \Pr\left[E^{-1}(P) + E^{-1}(P + \Delta_{\text{out}}) \in \mathcal{D}_{\text{in}}\right]$$

In general, the two probabilities are different, and related as follow:

$$\frac{\vec{p}}{|\mathcal{D}_{\text{out}}|} = \frac{\overleftarrow{p}}{|\mathcal{D}_{\text{in}}|} = \operatorname*{Avg}_{\Delta_{\text{in}} \in \mathcal{D}_{\text{in}}, \Delta_{\text{out}} \in \mathcal{D}_{\text{out}}} \Pr\left[E(P) + E(P + \Delta_{\text{in}}) = \Delta_{\text{out}}\right]$$

Figure 1 gives an example of a truncated differential on 3 rounds of AES, with respectively 4, 1, and 4 active S-Boxes in each round. $\mathcal{D}_{\text{in}}$ and $\mathcal{D}_{\text{out}}$ are vector spaces with $|\mathcal{D}_{\text{out}}| = |\mathcal{D}_{\text{in}}| = 2^{32}$. The probability of the truncated differential is $\vec{p} = 2^{-24}$ and the reverse probability is $\overleftarrow{p} = 2^{-24}$.

2.3 Boomerang Attacks

Boomerang attacks, introduced by Wagner in 1999 [39], use adaptive plaintext and ciphertext queries to generate quartets with specific differences at an intermediate state of the cipher. The attacker decomposes the full cipher E into two subciphers E_0 (the upper part) and E_1 (the lower part), with $E = E_1 \circ E_0$, with high probability differentials on E_0 and E_1 (of probabilities p and q), denoted respectively $\Delta_{\text{in}} \xrightarrow[E_0]{p} \Delta_{\text{out}}$ and $\nabla_{\text{in}} \xrightarrow[E_1]{q} \nabla_{\text{out}}$. The attack proceeds as follows:

1. Generate pairs of plaintext (P, P') such that $P + P' = \Delta_{\text{in}}$, and query the corresponding ciphertexts $(C, C') = (E(P), E(P'))$.
2. Shift the ciphertexts pairs into new pairs $(\overline{C}, \overline{C'}) = (C + \nabla_{\text{out}}, C' + \nabla_{\text{out}})$ and query their decryptions $(\overline{P}, \overline{P'}) = (E^{-1}(\overline{C}), E^{-1}(\overline{C'}))$.
3. Look for pairs with $\overline{P} + \overline{P'} = \Delta_{\text{in}}$.

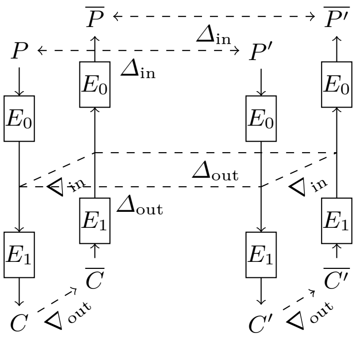


Fig. 2. Construction of a boomerang quartet.

Analysis. We have $E_0(P) = E_1^{-1}(C)$ because $E = E_1 \circ E_0$. In particular,

$$E_0(\overline{P}) + E_0(\overline{P}^{\prime}) = E_1^{-1}(\overline{C}) + E_1^{-1}(\overline{C'}) + E_0(P) + E_1^{-1}(C) + E_0(P') + E_1^{-1}(C')$$

Moreover, the differentials in E_0 and E_1 imply that:

$$\begin{aligned} \Pr[E_0(P) + E_0(P') = \Delta_{\text{out}}] &= p \\ \Pr[E_1^{-1}(C) + E_1^{-1}(\overline{C}) = \nabla_{\text{in}}] &= q \\ \Pr[E_1^{-1}(C') + E_1^{-1}(\overline{C'}) = \nabla_{\text{in}}] &= q \end{aligned}$$

When the three events are satisfied, we obtain $E_0(\overline{P}) + E_0(\overline{P'}) = \Delta_{\text{out}}$ and with an additional probability p, $\overline{P} + \overline{P}^{\prime} = \Delta_{\text{in}}$. Finally, assuming that all events are independent, we compute the boomerang trail probability p_b as a lower bound of the probability of the boomerang relation $\overline{P} + \overline{P'} = \Delta_{\text{in}}$:

$$\Pr\left[E^{-1}(E(P) + \nabla_{\text{out}}) + E^{-1}(E(P + \Delta_{\text{in}}) + \nabla_{\text{out}}) = \Delta_{\text{in}}\right] \geq p_b = p^2 \times q^2$$

Figure 2 shows the construction of a boomerang quartet. When $p^2 \times q^2 \gg 2^{-n}$, this gives a distinguisher for the cipher using $\mathcal{O}(p^{-2} \times q^{-2})$ quartets because the probability of detecting a quartet is 2^{-n} for a random permutation. In most cases, the distinguisher can be converted into a key recovery by exploiting key dependencies in the distinguisher.

2.4 Improvements of the Boomerang Attack

Analysis of the Connection Probability. The analysis above assumes that the four pairs involved in a boomerang quartet follow their corresponding differentials independently. In practice, we usually obtain a probability higher than p^2q^2, but it is also possible for the four events to be incompatible [30]. Several techniques have been proposed to improve this analysis.

Multiple Differentials. Since the differences $\varDelta_{\mathrm{out}}$ and ∇_{in} are not used by the attacker, boomerang quartets can be detected with any internal difference, as long as the same difference is obtained with both pairs. Following the analysis of [39,8], this increases the probability to

$$p_b = \hat{p}^2\hat{q}^2 \qquad \hat{p} = \sqrt{\sum_{\varDelta_{\mathrm{out}}} \Pr[\varDelta_{\mathrm{in}} \xrightarrow[E_0]{} \varDelta_{\mathrm{out}}]^2} \qquad \hat{q} = \sqrt{\sum_{\nabla_{\mathrm{in}}} \Pr[\nabla_{\mathrm{in}} \xrightarrow[E_1]{} \nabla_{\mathrm{out}}]^2}$$

The Sandwich Attack. Instead of splitting the cipher E into two parts $E = E_1 \circ E_0$, Dunkelman, Keller and Shamir [20] proposed to split it in three parts $E = E_1 \circ E_m \circ E_0$ with a small E_m in the middle. For the analysis, they evaluate the probability of the boomerang trail using the connection probability r of E_m:

$$\Pr\left[\overline{P} + \overline{P}' = \varDelta_{\mathrm{in}}\right] \geq p_b = p^2q^2r$$
$$r = \Pr\left[E_m^{-1}(E_m(X) + \nabla_{\mathrm{in}}) + E_m^{-1}(E_m(X + \varDelta_{\mathrm{out}}) + \nabla_{\mathrm{in}}) = \varDelta_{\mathrm{out}}\right]$$

The connection probability r can be evaluated experimentally, and some specific choices of E_m result in $r = 1$ (in particular, when E_m is the identity, we fall back to the standard analysis of boomerangs). The Boomerang Connectivity Table (BCT) was later introduced [13] to analyze the case where E_m is an S-Box layer. The case where E_m is composed of several rounds has been analyzed in further works [36,40,16]. Recent works also show that setting E_m to a single S-Box layer might lead to unaccurate connection probabilities [41].

Structures. Biham, Dunkelman and Shamir have introduced a variant of the boomerang attack using structures for the key recovery [9]. They start from a boomerang distinguisher with fixed differences $\varDelta_{\mathrm{in}}$ and ∇_{out}, and add extra rounds at the beginning and at the end. By propagating the differences $\varDelta_{\mathrm{in}}$ and ∇_{out}, they obtain a set of possible input differences $\mathcal{D}_{\mathrm{in}}$ and output differences $\mathcal{D}_{\mathrm{out}}$. In a typical SPN cipher, these sets are vector spaces.

The attacker builds a structure $P + \mathcal{D}_{\mathrm{in}} = \{P + \delta \,:\, \delta \in \mathcal{D}_{\mathrm{in}}\}$, and uses it as starting point for the attack. A structure of $|\mathcal{D}_{\mathrm{in}}|$ elements defines $|\mathcal{D}_{\mathrm{in}}|^2/2$ pairs, and $|\mathcal{D}_{\mathrm{in}}|/2$ of them lead to the fixed difference $\varDelta_{\mathrm{in}}$. Therefore, the use of structures covers additional rounds without increasing the data complexity.

Structures can also be used on the ciphertext side, by shifting each ciphertext with all differences in $\mathcal{D}_{\mathrm{out}}$. However, many later works do not use structures on the ciphertext side.

Retracing Boomerang. Dunkelman *et al.* [19] proposed an improvement where the ciphertexts are chosen so that the two returning pairs $(C, \overline{C})$ and $(C', \overline{C'})$ are dependant: if one passes the trail on E_1, the other passes it too. This increases the probability of the trail to $p_b = p^2q$. In the same spirit, Rahman *et al.* [32] recently proposed a boomerang attack embedding a yoyo distinguisher.

3 Truncated Boomerang Attacks

We consider boomerang attacks with truncated differentials, as introduced by Wagner in the original paper [39]. We obtain a key-recovery attack, improving on the use of structures of Biham *et al.* [9] by considering truncated differentials for the full cipher, instead of starting from a shorter boomerang distinguisher with fixed input/output differences and adding truncated trails for the key-recovery rounds. Some boomerang attacks of this type have been proposed on AES [10] and Kiasu-BC [18], but they only use structures on the plaintext side. Our framework unifies and improves on previous boomerang attacks, using truncated differentials for E_0 and E_1, and structures on both sides.

3.1 Truncated Boomerang Distinguisher

Let us consider two truncated differentials $\mathcal{D}^0_{\text{in}} \xrightarrow[E_0]{p} \mathcal{D}^0_{\text{out}}$ and $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{q} \mathcal{D}^1_{\text{out}}$ with probabilities $\vec{p}, \overleftarrow{p}$ and $\vec{q}, \overleftarrow{q}$ on E_0 and E_1. We assume that $\mathcal{D}^0_{\text{in}}$ is a vector subspace of $\{0,1\}^n$ and $0 \notin \mathcal{D}^1_{\text{out}}$. The truncated boomerang attack proceeds as follows:

1. Choose a random plaintext P_0, and query the encryption oracle over the structure $P_0 + \mathcal{D}^0_{\text{in}}$; for each $i \in \mathcal{D}^0_{\text{in}}$, we define $P_i = P_0 + i$ and $C_i = E(P_i)$.
2. For each ciphertext C_i, query the decryption oracle over the set $C_i + \mathcal{D}^1_{\text{out}}$: for each $j \in \mathcal{D}^1_{\text{out}}$, we define $\overline{C_i}^j = C_i + j$ and $\overline{P_i}^j = E^{-1}(\overline{C_i}^j)$.
3. Count the number of pairs $(\overline{P_i}^j, \overline{P_{i'}}^{j'})$ with $\overline{P_i}^j + \overline{P_{i'}}^{j'} \in \mathcal{D}^0_{\text{in}}$ (and $i \neq i'$). This can be done efficiently by projecting the plaintext values on the orthogonal complement of $\mathcal{D}^0_{\text{in}}$ in $\{0,1\}^n$, and looking for collisions.
4. If needed, repeat steps 1 to 3 with different plaintext structures.

Analysis. We consider a potential quartet $(P, P', \overline{P}, \overline{P'})$ corresponding to $(C, C', \overline{C}, \overline{C'})$, with $P + P' \in \mathcal{D}^0_{\text{in}}$ and $C + \overline{C}, C' + \overline{C'} \in \mathcal{D}^1_{\text{out}}$. We have:

$$\Pr[E_0(P) + E_0(P') \in \mathcal{D}^0_{\text{out}}] = \vec{p}$$
$$\Pr[E_1^{-1}(C) + E_1^{-1}(\overline{C}) \in \mathcal{D}^1_{\text{in}}] = \overleftarrow{q}$$
$$\Pr[E_1^{-1}(C') + E_1^{-1}(\overline{C'}) \in \mathcal{D}^1_{\text{in}}] = \overleftarrow{q}$$

Following the sandwich attack analysis (with $E_m = \text{id}$), we define the connection probability:

$$r = \Pr\left[E_0(\overline{P}) + E_0(\overline{P'}) \in \mathcal{D}^0_{\text{out}} \;\middle|\; \begin{array}{r} E_0(P) + E_0(P') \in \mathcal{D}^0_{\text{out}} \\ E_1^{-1}(C) + E_1^{-1}(\overline{C}) \in \mathcal{D}^1_{\text{in}} \\ E_1^{-1}(C') + E_1^{-1}(\overline{C'}) \in \mathcal{D}^1_{\text{in}} \end{array} \right]$$

If the four events hold, we have $\overline{P} + \overline{P'} \in \mathcal{D}^0_{\text{in}}$ with an additional probability $\overleftarrow{p}$. This analysis of the truncated boomerang distinguisher is the same as proposed by Wagner [39], but our attack is more general with structures on both sides.

In general, we have $E_1^{-1}(C)+E_1^{-1}(\overline{C})$ and $E_1^{-1}(C')+E_1^{-1}(\overline{C'})$ in $\mathcal{D}^1_{\text{in}}$, therefore they are equal with probability $|\mathcal{D}^1_{\text{in}}|^{-1}$, and this implies $E_0(\overline{P}) + E_0(\overline{P'}) \in \mathcal{D}^0_{\text{out}}$; hence $r \geq |\mathcal{D}^1_{\text{in}}|^{-1}$. Moreover, if $\mathcal{D}^1_{\text{in}}$ and $\mathcal{D}^0_{\text{out}}$ are vector subspaces, then $\varSigma = E_0(P)+E_0(P')+E_0(\overline{P})+E_0(\overline{P'}) \in \mathcal{D}^1_{\text{in}}$; in particular, $\varSigma \in \mathcal{D}^0_{\text{out}}$ with probability $|\mathcal{D}^0_{\text{out}} \cap \mathcal{D}^1_{\text{in}}|/|\mathcal{D}^1_{\text{in}}|$; this implies $E_0(\overline{P})+E_0(\overline{P'}) \in \mathcal{D}^0_{\text{out}}$ hence $r \geq |\mathcal{D}^0_{\text{out}} \cap \mathcal{D}^1_{\text{in}}|/|\mathcal{D}^1_{\text{in}}|$.

Assuming that all the events are independent, each quartet $(P_i, P_{i'}, \overline{P_i}^j, \overline{P_{i'}}^{j'})$, defined by a pair $(i,j), (i',j')$, follows the truncated boomerang trail with probability p_b, and randomly satisfies $\overline{P_i}^j + \overline{P_{i'}}^{j'} \in \mathcal{D}^0_{\text{in}}$ with probability $p_\$$:

$$p_b = \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}^2 \cdot r \qquad\qquad r \geq |\mathcal{D}^1_{\text{in}}|^{-1} \tag{1}$$

$$p_\$ = |\mathcal{D}^0_{\text{in}}|/2^n \tag{2}$$

We assume that the boomerang probability can be well approximated as $p_b + p_\$$. We distinguish the cipher E from a random permutation when the expected number of remaining quartets (quartets with $\overline{P_i}^j + \overline{P_{i'}}^{j'} \in \mathcal{D}^0_{\text{in}}$) is significantly higher for E than for a random permutation. We define the signal-to-noise ratio:

$$\sigma = p_b/p_\$ \tag{3}$$

When $\sigma \gg 1$, we obtain a distinguisher using $Q = \mathcal{O}(p_b^{-1})$ quartets. More precisely, with $Q = \mu \cdot p_b^{-1}$ (μ a small constant) we expect μ remaining quartets with the cipher E, versus $\mu \cdot \sigma^{-1} \ll 1$ for a random permutation. A distinguisher that detects the presence of at least one quartet has a success rate of $1 - e^{-\mu}$.

When σ is smaller, we need to collect a large number of quartets, and compare the expected number of remaining quartets q_b for E and $q_\$$ in the random case:

$$q_b = Q \times (p_\$ + p_b) = Q \times p_\$(1+\sigma) \qquad\qquad q_\$ = Q \times p_\$$$

We detect the bias with $Q = \mathcal{O}(p_\$^{-1}\sigma^{-2}) = \mathcal{O}(p_b^{-1}\sigma^{-1})$ samples, following [29, Theorem 2]. Using $Q = c \times p_b^{-1}\sigma^{-1}$ with a small constant c and setting a threshold at $Q \times p_\$(1+\sigma/2)$, the distinguisher has a success rate of $\varPhi(\sqrt{c}/2)$, with $\varPhi$ the cumulative distribution function of the standard normal distribution.

If Q is smaller than the number of quartets in a full structure ($|\mathcal{D}^0_{\text{in}}|^2|\mathcal{D}^1_{\text{out}}|^2/2$), we use a partial structure with only $\sqrt{2Q}$ elements. Otherwise, we need $N = 2Q \times |\mathcal{D}^0_{\text{in}}|^{-2}|\mathcal{D}^1_{\text{out}}|^{-2}$ structures of $S = |\mathcal{D}^0_{\text{in}}||\mathcal{D}^1_{\text{out}}|$ elements. Finally, we obtain a distinguisher with a constant probability of success with the following complexity in number of quartets, time, data, and memory:

$$Q = \mathcal{O}\left(\max(p_b^{-1}, \sigma^{-1} \cdot p_b^{-1})\right) \tag{4}$$

$$T = D = \max(\sqrt{2Q}, 2Q \times |\mathcal{D}^0_{\text{in}}|^{-1}|\mathcal{D}^1_{\text{out}}|^{-1}) \tag{5}$$

$$M = \min(D, |\mathcal{D}^0_{\text{in}}||\mathcal{D}^1_{\text{out}}|) \tag{6}$$

Application to 6-round AES. To explain the truncated boomerang distinguisher in practice, we give a truncated boomerang trail on 6-round AES in figure 3, using the 3-round trail of figure 1 twice. $\mathcal{D}^0_{\text{in}}$ and $\mathcal{D}^1_{\text{in}}$ are the sets of all

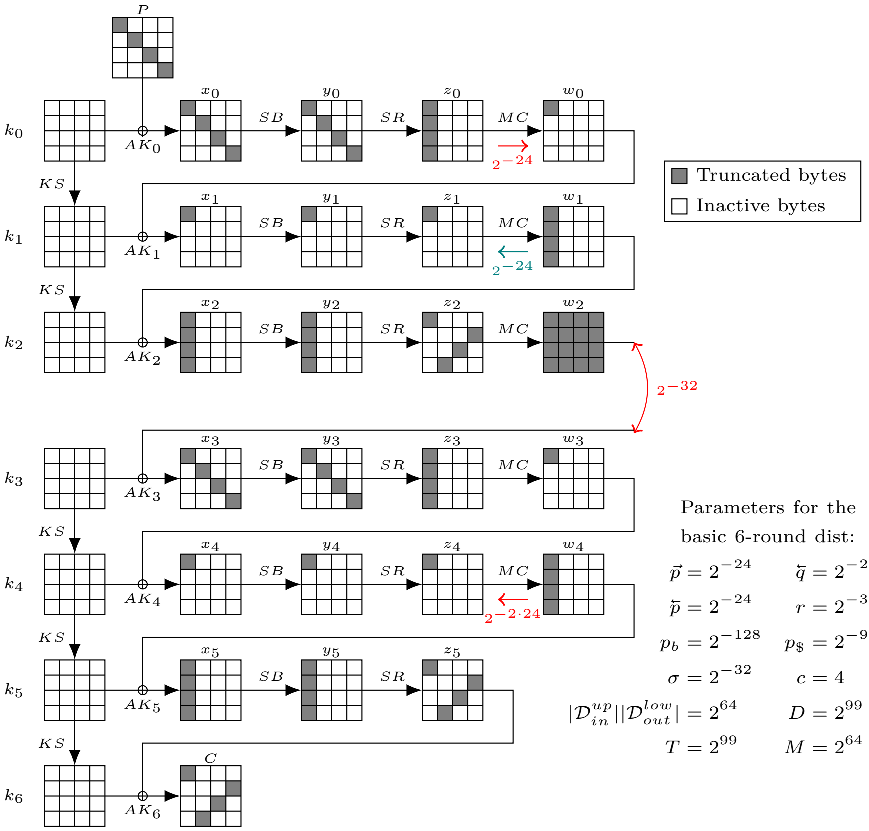


Fig. 3. A truncated boomerang trail on 6-round AES.

states that have zeros on all diagonals except the main one. $\mathcal{D}^0_{\text{out}}$ is the same as the output set of figure 1, and $\mathcal{D}^1_{\text{out}}$ is active on the main anti-diagonal (it differs because we omit the last MixColumns operation). We have

$$|\mathcal{D}^0_{\text{in}}| = |\mathcal{D}^0_{\text{out}}| = 2^{32} \qquad \vec{p} = 2^{-24} \qquad \overleftarrow{p} = 2^{-24}$$
$$|\mathcal{D}^1_{\text{in}}| = |\mathcal{D}^1_{\text{out}}| = 2^{32} \qquad \vec{q} = 2^{-24} \qquad \overleftarrow{q} = 2^{-24}$$

Since $\mathcal{D}^0_{\text{out}} \cap \mathcal{D}^1_{\text{in}} = \{0\}$, we have $r = |\mathcal{D}^1_{\text{in}}|^{-1}$, and the analysis above gives the following parameters:

$$p_b = \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-128} \qquad \sigma = 2^{-32}$$
$$p_\$ = |\mathcal{D}^0_{\text{in}}|/2^n = 2^{-96} \qquad Q = c \cdot 2^{160}$$

Using $c = 4$ and the formulas of Equations (4), (5), and (6), we obtain a distinguisher with complexity:

$$T = D = 2^{99} \qquad M = 2^{64}$$

The distinguisher is given in algorithm form in the full version [6]. It makes 2^{67} encryption queries and 2^{99} decryption queries, for a total data complexity of $D = 2^{67} + 2^{99} \approx 2^{99}$. In total, we have $Q = 2^{35} \times 2^{64} \times 2^{64}/2 = 2^{162}$ quartets $(P_i, P_{i'}, \overline{P_i}^{j}, \overline{P_{i'}}^{j'})$, so that the expected number of remaining quartets is:

$$q_{\$} = Q \times 2^{-96} = 2^{66} \qquad\qquad q_E = Q \times (2^{-96} + 2^{-128}) = 2^{66} + 2^{34}$$

The distinguisher returns the correct answer with probability $\Phi(\sqrt{c}/2) \approx 0.84$.

This distinguisher is interesting because it is very generic: it does not require knowledge of the S-Box or the MDS matrix, and it can be considered as "key-independent" in the sense of [24]. As seen in Table 1, the complexity is slightly higher than previous distinguishers with similar properties. However, the simplicity of this distinguisher makes it more likely to be applicable when 6-round AES is used as a building block in a more complex structure, as shown in section 6 against TNT-AES.

3.2 Truncated Boomerang Key-recovery Attack

We now consider key-recovery attacks. As opposed to typical differential or linear attacks, we do not add rounds on top of the distinguisher. Instead, we assume that the truncated boomerang covers the full targeted cipher, and we design a key-recovery attack with smaller complexity than the corresponding distinguisher.

When $\sigma \geq 1$, the truncated boomerang distinguisher is easy to turn into a key-recovery attack, but we cannot reduce the complexity. Indeed, the bottleneck of the distinguisher is to have enough data so that a boomerang quartet exists. When a quartet with $\overline{P} + \overline{P'} \in \mathcal{D}^0_{\text{in}}$ is found, it has a high probability of following the boomerang, and standard methods can be used to recover key candidates. Therefore, we focus on the case $\sigma \ll 1$, where the distinguisher requires multiple quartets following the boomerang.

Given a candidate quartet with $\overline{P} + \overline{P'} \in \mathcal{D}^0_{\text{in}}$, we can extract some key information assuming that it follows the boomerang. If this is the case, we have two pairs of known plaintexts (P, P') and $(\overline{P}, \overline{P'})$ following the truncated differential $\mathcal{D}^0_{\text{in}} \xrightarrow[E_0]{p} \mathcal{D}^0_{\text{out}}$, and two pairs of known ciphertexts $(C, \overline{C})$ and $(C', \overline{C'})$ following the truncated differential $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{q} \mathcal{D}^1_{\text{out}}$. Using standard techniques from differential cryptanalysis, we can usually extract partial information about the first and last subkeys. We denote by κ the number of key bits that can be extracted, and by ℓ the average number of κ-bit key candidates suggested by a quartet. Note that the key information suggested by a quartet might be incompatible between both pairs of plaintexts following the upper differential (or between both pairs of ciphertexts following the lower differential), in this case the quartet is discarded.

We follow the standard approach to identify the most likely candidates for the κ bits of key: we build a table of 2^{κ} counters corresponding to key candidates, and we increment the counter of each key suggested by each quartet. With enough data, the right key is expected to be among the top $2^{\kappa - a}$ counters (a denotes the advantage of the attack).

Analysis. Following the previous analysis, we expect $Q \times (p_{\$} + p_b)$ quartets with $\overline{P} + \overline{P'} \in \mathcal{D}^0_{\text{in}}$: $Q \times p_b$ quartets following the boomerang trail (right quartets), and $Q \times p_{\$}$ false positives. For a right quartet, the correct key is among the deduced key candidates, and for a wrong quartet, we expect that ℓ random key candidates are deduced. Assuming that all the quartets behave independently, the wrong counters follow the binomial distribution $\mathcal{B}(Q, (p_{\$} + p_b) \times \ell \times 2^{-\kappa})$ and the right counter follows the distribution $\mathcal{B}(Q, p_{\$} \times \ell \times 2^{-\kappa} + p_b)$. We denote the probabilities of suggesting a wrong key and the right key as:

$$p_w = (p_{\$} + p_b) \times \ell \times 2^{-\kappa} \approx p_{\$} \times \ell \times 2^{-\kappa} \tag{7}$$

$$p_0 = p_{\$} \times \ell \times 2^{-\kappa} + p_b \approx p_w + p_b \tag{8}$$

We obtain a higher signal-to-noise ratio $\tilde{\sigma}$ than previously:

$$\tilde{\sigma} = p_b / p_w = \sigma \times 2^{\kappa} / \ell \tag{9}$$

When $\tilde{\sigma} \gg 1$, only a handful of right quartets are necessary to have the right key ranked first, so that $Q = \mathcal{O}(p_b^{-1})$.

When $\tilde{\sigma} \ll 1$, the counters can be approximated by normal distributions, and we use the work of Selçuk [35, Theorem 3] to evaluate the number of samples needed to have the right key among the top $2^{\kappa - a}$ key candidates (depending on the success rate). For a fixed value of a, we need Q proportional to $p_b^{-1}\tilde{\sigma}^{-1}$, and the complexity increases linearly in a. Finally, the increased signal-to-noise ratio $\tilde{\sigma} \gg \sigma$ reduces the data complexity to:

$$Q = \mathcal{O}\left(\max(p_b^{-1}, \tilde{\sigma}^{-1} \times p_b^{-1})\right) \tag{10}$$

$$D = \max(\sqrt{2Q}, 2Q \times |\mathcal{D}^0_{\text{in}}|^{-1} |\mathcal{D}^1_{\text{out}}|^{-1}) \tag{11}$$

The time complexity is harder to evaluate; it can be bounded with T_E the cost of an oracle call (by convention, $T_E = 1$), and T_C the cost of deducing key candidates from a quartet:

$$T = D \times T_E + Q \times p_{\$} \times T_C \tag{12}$$

When $T_C \ll 2^n \times |\mathcal{D}^0_{\text{in}}|^{-2} |\mathcal{D}^1_{\text{out}}|^{-1}$, we have $Q \times p_{\$} \times T_C \ll D$ and the second term is negligible; the cost of the attack is thus dominated by the oracle queries. Otherwise, it is often possible to reduce the second term with more advanced filtering, but this requires a dedicated analysis for each attack.

After recovering $2^{\kappa - a}$ candidates for the κ-bit partial key, the full key can be recovered by exhaustive search of the remaining bits with complexity 2^{n-a}, or by launching a variant of the attack on a different set of key bits.

Success Probability. When $\tilde{\sigma} \ll 1$, the average values of right and wrong counters are high enough to approximate them with normal distributions. In that case, the success rate can be evaluated using the formula given by [35], under additional assumptions about the independence of key counters, and order statistics:

$$P_S = \Phi\left(\frac{\sqrt{\mu\tilde{\sigma}} - \Phi^{-1}(1 - 2^{-a})}{\sqrt{\tilde{\sigma} + 1}}\right) \tag{13}$$

with $\mu = Q \times p_b$ the expected number of right quartets.

When $\tilde{\sigma}$ is high, the binomial distributions of right and wrong counters have their average values respectively $Q \times p_b \approx 1$ and $Q \times p_w \ll 1$. As discussed in [35, section 3.2.1], the normal approximation is inaccurate in this case; we obtain a more accurate estimate of the success probability using a Poisson approximation.

Extracting Key Candidates. When the truncated differentials are described by truncated trails (with a set of intermediate differences at each step), the parameters ℓ and κ can often be deduced directly from the trail. We assume that E_0 starts with the addition of a subkey K_0, followed by an S-Box layer SB, and we denote the set of differences after the S-Box layer by $\mathcal{D}^0_{\text{mid}}$:

$$E_0 = \tilde{E_0} \circ \mathsf{SB} \circ \mathsf{AK}_{K_0} \qquad \mathcal{D}^0_{\text{in}} \xrightarrow[\mathsf{SB}]{p_0} \mathcal{D}^0_{\text{mid}} \qquad \mathcal{D}^0_{\text{mid}} \xrightarrow[\tilde{E_0}]{p_1} \mathcal{D}^0_{\text{out}}$$

We also assume that $\mathcal{D}^0_{\text{in}}$ is a vector subspace aligned with the S-Box layer (each S-Box is either inactive, or active with all possible differences). $\mathcal{D}^0_{\text{mid}}$ is a subset of $\mathcal{D}^0_{\text{in}}$; typically it is constructed so that some parts of the state have fixed differences after the linear layer. For instance, in the AES trail of Figure 1, $\mathcal{D}^0_{\text{mid}}$ corresponds to differences δ such that $\mathsf{MixColumns}(\mathsf{ShiftRows}(\delta))$ is active only on the first cell, with $|\mathcal{D}^0_{\text{mid}}| = 2^8$ and $\vec{p}_0 = 2^{-24}$. In general, we have:

$$\vec{p}_0 = |\mathcal{D}^0_{\text{mid}}| / |\mathcal{D}^0_{\text{in}}| \qquad \vec{p} = \vec{p}_0 \times \vec{p}_1 \tag{14}$$

We consider a pair (P, P'), and assume that it follows the truncated trail, *i.e.* $\mathsf{SB}(P + K_0) + \mathsf{SB}(P' + K_0) \in \mathcal{D}^0_{\text{mid}}$. This constrains the partial subkey $K_{0|\mathcal{D}^0_{\text{in}}}$ corresponding to the active S-Boxes in SB. More precisely, for each difference δ in $\mathcal{D}^0_{\text{mid}}$, we expect on average 0.5 unordered pairs $\{X, X'\}$ such that $X + X' = P + P'$ and $\mathsf{SB}(X) + \mathsf{SB}(X') = \delta$ (restricted to the active bytes $\mathcal{D}^0_{\text{mid}}$). This pair can be recovered efficiently after pre-computing the DDT of the S-Box, and we deduce two possible keys $X + P$ and $X + P'$. Therefore, we have the following parameters when extracting key candidates from a pair (P, P'):

$$\ell^0 = |\mathcal{D}^0_{\text{mid}}| \qquad \kappa^0 = \log_2(|\mathcal{D}^0_{\text{in}}|) \qquad T^0_C = \ell^0 = |\mathcal{D}^0_{\text{mid}}|$$

Starting from a candidate quartet, we have two different pairs (P, P') and $(\overline{P}, \overline{P}')$ assumed to follow the upper differential. Therefore, we expect only $|\mathcal{D}^0_{\text{mid}}|^2 / |\mathcal{D}^0_{\text{in}}|$ key candidates compatible with both pairs. We apply the same reasoning to the lower trail (using ciphertext pairs), and deduce the parameters ℓ and κ for a quartet in the general case:

$$\ell = |\mathcal{D}^0_{\text{mid}}|^2 \cdot |\mathcal{D}^1_{\text{mid}}|^2 \cdot |\mathcal{D}^0_{\text{in}}|^{-1} \cdot |\mathcal{D}^1_{\text{out}}|^{-1} \qquad \kappa = \log_2(|\mathcal{D}^0_{\text{in}}| \cdot |\mathcal{D}^1_{\text{out}}|) \tag{15}$$

Using the probability $\vec{p}_0$ for the first round and $\overleftarrow{q}_0$ for the last round, we have

$$\ell \cdot 2^{-\kappa} = \vec{p}_0^{\,2} \cdot \overleftarrow{q}_0^{\,2} \tag{16}$$

For the lower trail, we only have to process a fraction $|\mathcal{D}^0_{\text{mid}}|^2 / |\mathcal{D}^0_{\text{in}}|$ of the candidate quartets (with a key compatible with both pairs). In particular, when $|\mathcal{D}^0_{\text{mid}}|^2 \ll |\mathcal{D}^0_{\text{in}}|$, the time complexity is dominated by the first extraction step: $T_C = |\mathcal{D}^0_{\text{mid}}|$.

Application to 6-round AES. This attack can directly be applied to AES, using the same 3-round trails as in the previous section (see Figure 3):

$$\begin{array}{llll}
|\mathcal{D}^0_{\text{in}}| = |\mathcal{D}^0_{\text{out}}| = 2^{32} & |\mathcal{D}^0_{\text{mid}}| = 2^{8} & \vec{p} = 2^{-24} & \overleftarrow{p} = 2^{-24} \\
|\mathcal{D}^1_{\text{in}}| = |\mathcal{D}^1_{\text{out}}| = 2^{32} & |\mathcal{D}^1_{\text{mid}}| = 2^{8} & \vec{q} = 2^{-24} & \overleftarrow{q} = 2^{-24}
\end{array}$$

Using the parameters of the key extraction, our analysis results in

$$\begin{array}{ll}
\ell = |\mathcal{D}^0_{\text{mid}}|^2 \cdot |\mathcal{D}^1_{\text{mid}}|^2 \cdot |\mathcal{D}^0_{\text{in}}|^{-1} \cdot |\mathcal{D}^1_{\text{out}}|^{-1} = 2^{-32} & \kappa = \log_2(|\mathcal{D}^0_{\text{in}}| \cdot |\mathcal{D}^1_{\text{out}}|) = 64 \\
p_b = \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-128} & \\
p_w = |\mathcal{D}^0_{\text{in}}| \times 2^{-n} \times \ell \times 2^{-\kappa} = 2^{-192} & \tilde{\sigma} = 2^{64}
\end{array}$$

Since $\tilde{\sigma} \gg 1$, we only need a few right quartets; with $\mu = 4$ we obtain

$$Q = \mu \times p_b^{-1} = 2^{130} \qquad\qquad D = 2^{67}$$

Time complexity. With these parameters, the attack complexity is dominated by the oracle queries. We use 8 structures of 2^{64} elements; in each structure we detect $2^{64} \times 2^{63} \times p_{\$} = 2^{31}$ pairs with $\overline{P} + \overline{P}' \in \mathcal{D}^0_{\text{in}}$, resulting in $8 \times 2^{31} = 2^{34}$ candidate quartets in total. Each quartet suggests on average 2^{-32} candidates for 64 bits of key (for most of the quartets, there is no key compatible with both sides of the quartet). Finally, we expect 2^2 suggestions of wrong keys (each key is suggested 2^{-62} times on average), and $\mu = 4$ suggestions for the correct key. With high probability, the key with the most suggestions is the correct one.

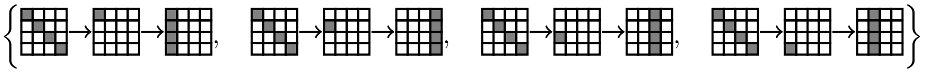
We implemented the attack on a reduced AES with 4-bit S-Boxes, and it behaves as expected [7].

4 Optimized Boomerang Attacks on 6-round AES

As shown by Biryukov [10], boomerang attacks on AES can be optimized using multiple trails. We now present improved versions of our attacks using this technique, including a 6-round key-recovery attack with complexity 2^{61}. The improvement compared to the attack of Biryukov with complexity 2^{71} is due to the use of structures on the ciphertext side.

4.1 Optimized distinguisher

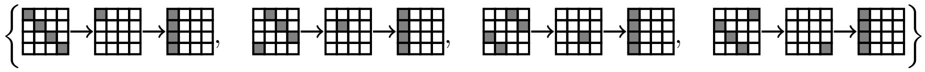
Instead of only considering the trail of Figure 1 with fixed positions for all the active bytes, we consider a collection of four different trails for upper part:



The collection can be considered as a truncated differential $\mathcal{D}^0_{\text{in}} \xrightarrow[E_0]{p} \mathcal{D}^0_{\text{out}}$ with

$$\vec{p} = 2^{-22} \qquad \overleftarrow{p} = 2^{-24} \qquad |\mathcal{D}^0_{\text{in}}| = 2^{32} \qquad |\mathcal{D}^0_{\text{out}}| = 2^{34}$$

Similarly, we consider four trails for the lower part:

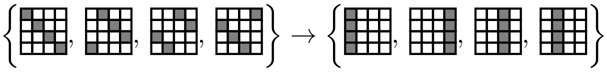


Again, this can be considered as a truncated differential $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{q} \mathcal{D}^1_{\text{out}}$ with

$$\vec{q} = 2^{-24} \qquad \overleftarrow{q} = 2^{-22} \qquad |\mathcal{D}^1_{\text{in}}| = 2^{34} \qquad |\mathcal{D}^1_{\text{out}}| = 2^{32}$$

The analysis of the previous sections can be applied as-is with these trails. We obtain a better attack because we increase $\vec{p}$ and $\overleftarrow{q}$ by a factor 4, even though $|\mathcal{D}^1_{\text{in}}|$ increases by a factor 4; we obtain $p_b = 2^{-124}$ instead of 2^{-128}. The distinguisher is exactly the same because $\mathcal{D}^0_{\text{in}}$ and $\mathcal{D}^1_{\text{out}}$ are the same, but this improved analysis shows that the complexity of the distinguisher can be reduced to $T = D = 2^{91}$ (with $c = 4$, $\sigma = 2^{-28}$ and $Q = 2^{154}$).

Larger set $\boldsymbol{\mathcal{D}^1_{\text{out}}}$. We further improve the distinguisher using a collection of 16 trails with the following input and output sets for the lower trail:



This collection can be considered as a truncated differential $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{q} \mathcal{D}^1_{\text{out}}$ with

$$\vec{q} = 2^{-22} \qquad \overleftarrow{q} = 2^{-22} \qquad |\mathcal{D}^1_{\text{in}}| = 2^{34} \qquad |\mathcal{D}^1_{\text{out}}| = 2^{34}$$

This does not affect the probability p_b, but generates larger structures; the complexity is reduced to $T = D = 2^{89}$ with $Q = 2^{154}$.

Different Set $\boldsymbol{\overline{\mathcal{D}}^0_{\text{in}}}$ for Returning Pairs. Following Biryukov [10], we use a higher probability differential for the returning pair $(\overline{P}, \overline{P'})$, different from for the initial pair (P, P'), and with a larger set $\overline{\mathcal{D}}^0_{\text{in}}$. We consider the same collection of 16 trails as above, corresponding to a truncated differential $\overline{\mathcal{D}}^0_{\text{in}} \xrightarrow[E_0]{\bar{p}} \mathcal{D}^0_{\text{out}}$ with

$$\vec{\overline{p}} = 2^{-22} \qquad \overleftarrow{\overline{p}} = 2^{-22} \qquad |\overline{\mathcal{D}}^0_{\text{in}}| = 2^{34} \qquad |\mathcal{D}^0_{\text{out}}| = 2^{34}$$

This corresponds to keeping quartets with a single active diagonal in $\overline{P} + \overline{P'}$, but not necessarily the main one. We adapt our analysis to account for the two distinct upper differentials and we obtain

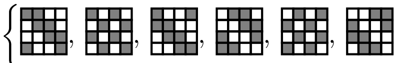
$$p_b = \vec{p} \cdot \overleftarrow{\overline{p}} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-122} \qquad \sigma = 2^{-28}$$
$$p_\$ = |\overline{\mathcal{D}}^0_{\text{in}}|/2^n = 2^{-94} \qquad Q = 2^{152}$$

Finally, we obtain a distinguisher with complexity $T = D = 2^{87}$ (with $c = 4$).

4.2 Optimized Key-recovery Attack

For a key-recovery attack, we can use the trails above to obtain an attack with complexity $T = D = 2^{62.5}$. We keep the set $\mathcal{D}^1_{\mathrm{out}}$ active only in the first column ($|\mathcal{D}^1_{\mathrm{out}}| = 2^{32}$) in order to extract information on the same key for each quartet. More details are given in the full version [6].

In order to further reduce the complexity of this attack, we use truncated boomerang characteristics with lower signal-to-noise ratios, taking advantage of the additional filter provided by the key extraction. Following [10], we modify the truncated trail on the returning side $(\overline{P}, \overline{P}')$ to allow any combination of two active diagonals in input, leading to the following parameters:

$\overline{\mathcal{D}}^0_{\mathrm{in}} = \Big\{$  $\Big\}$

$$
\begin{array}{llllll}
\vec{p} = 2^{-22} & \overleftarrow{p} = 2^{-24} & & |\mathcal{D}^0_{\mathrm{in}}| = 2^{32} & |\mathcal{D}^0_{\mathrm{out}}| = 2^{34} & |\mathcal{D}^0_{\mathrm{mid}}| = 2^{10} \\
\vec{\overline{p}} = 2^{-46} & \overleftarrow{\overline{p}} = 6 \times 2^{-16} = 2^{-13.4} & & |\overline{\mathcal{D}}^0_{\mathrm{in}}| = 6 \times 2^{64} & |\mathcal{D}^0_{\mathrm{out}}| = 2^{34} & \\
\vec{q} = 2^{-24} & \overleftarrow{q} = 2^{-22} & & |\mathcal{D}^1_{\mathrm{in}}| = 2^{34} & |\mathcal{D}^1_{\mathrm{out}}| = 2^{32} & |\mathcal{D}^1_{\mathrm{mid}}| = 2^{10}
\end{array}
$$

When extracting the key, we recover information about the main diagonal of k_0 from (P, P'), and information about the first anti-diagonal of k_6 from $(C, \overline{C})$ and $(C', \overline{C}')$ (note that $(\overline{P}, \overline{P}')$ is not necessarily active in the main diagonal). Moreover, the key suggested by $(C, \overline{C})$ and $(C', \overline{C}')$ must lead to the same active byte in z_4, so that

$$\ell^0 = 2^{10} \qquad \kappa^0 = 32 \qquad \ell^1 = 2^{-14} \qquad \kappa^1 = 32 \qquad \ell = 2^{-4} \qquad \kappa = 64$$

Using the previous analysis, we obtain

$$
\begin{aligned}
p_b &= \vec{p} \cdot \overleftarrow{\overline{p}} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\mathrm{in}}|^{-1} = 2^{-113.4} \\
p_w &= |\overline{\mathcal{D}}^0_{\mathrm{in}}| 2^{-n} \times \ell \times 2^{-\kappa} = 2^{-129.4} \qquad\qquad \tilde{\sigma} = 2^{16}
\end{aligned}
$$

Since $\tilde{\sigma} \gg 1$, a few right quartets are sufficient for the success of this attack; we use $\mu = 8$, this corresponds to $Q = 2^{116.4}$ and we use a partial structure of $D = 2^{58.7}$ elements.

Success probability. We assume that the attacker keeps key candidates with counter values of at least 5. With $\tilde{\sigma} \gg 1$, we approximate the wrong key counters by Poisson distributions with $\lambda = Q \times p_w = 2^{-13}$, each of which equal 5 or more with probability $1 - e^{-\lambda}(1 + \lambda + \lambda^2/2 + \lambda^3/6 + \lambda^4/24) \approx 2^{-71.9}$; we don't expect to keep any wrong keys. On the other hand, the counter for the right key follows a Poisson distribution with $\lambda = \mu = 8$. It reaches a value of 5 or more with probability ≈ 0.9.

Time complexity. After recovering a candidate for 64 bits of key (32 bits of k_0 and 32 bits of k_6), we repeat the attack with $\mathcal{D}^0_{\text{in}}$ in a different diagonal and use the partial knowledge of k_6 to increase the probability $\overline{q}$. This step has a negligible complexity.

The time complexity is balanced between oracle queries and extracting key candidates. Indeed, we filter $2^{58.7} \times 2^{57.7} \times |\overline{\mathcal{D}}^0_{\text{in}}| \times 2^{-n} = 2^{55}$ candidates with $\overline{P} + \overline{P}' \in \overline{\mathcal{D}}^0_{\text{in}}$ using 6 hash tables indexed by each combination of two active columns. The complexity T_C to generate key candidates for a given quartet is essentially 4×2^{10} accesses to a small table; we approximate it as $T_C \approx 2^{5.4} T_E$ (since one encryption has 6×16 S-Boxes). Finally, the time complexity is

$$T = 2^{58.7} T_E + 2^{55} T_C \approx 2^{60.8} T_E$$

4.3 Key-recovery with Secret S-Boxes

The techniques described in subsection 3.2 assume that the S-Box and MDS matrix are known to the attacker in order to extract key information. However, it is also possible to extract key information with an unknown S-Box under some conditions. Following [23], we assume that all S-Boxes in a column are identical, and that the MDS matrix has two identical coefficients in each row.

As a concrete example, we consider the AES MixColumns matrix

$$\mathsf{MC} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

We consider a pair $(C, \overline{C})$ following the truncated trail of Figure 3. According to the trail, the difference before the last round (w_4) is in a set of 2^8 differences; in particular, the difference in cell 1 is equal to the difference in cell 2:

$$w_4 + \overline{w}_4 \in \left\{ \begin{bmatrix} 2\delta & 0 & 0 & 0 \\ \delta & 0 & 0 & 0 \\ \delta & 0 & 0 & 0 \\ 3\delta & 0 & 0 & 0 \end{bmatrix} : \delta \in \{0,1\}^8 \right\} = \left\{ \mathsf{MC} \cdot \begin{bmatrix} \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} : \delta \in \{0,1\}^8 \right\}$$

Moreover, we assume that the differences in cells 13 and 10 of the ciphertext are equal (they are moved to cell 1 and 2 by ShiftRows)

$$C + \overline{C} = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta \\ 0 & 0 & \beta & 0 \\ 0 & \gamma & 0 & 0 \end{bmatrix}$$

In this case, S-Boxes 1 and 2 in the last round follow the same transition $\delta \to \beta$. With high probability, this implies that the unordered pairs of input/output are equal; in particular $\{C[13] + k_6[13], \overline{C}[13] + k_6[13]\} = \{C[10] + k_6[10], \overline{C}[10] + k_6[10]\}$. This suggests two key candidates:

$$k_6[13] + k_6[10] \in \left\{ C[13] + C[10], C[13] + \overline{C}[10] \right\}$$

In order to use this property in a truncated boomerang attack, we use the multiple upper trails of subsection 4.1, and a single lower trail with a restricted

$\mathcal{D}^1_{\text{out}}$ of size 2^{24} to ensure that $C+\overline{C}$ and $C'+\overline{C}'$ have the required properties for all quartets considered:

$$\mathcal{D}^1_{\text{out}} = \left\{ \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & 0 & 0 & b \\ 0 & 0 & b & 0 \\ 0 & c & 0 & 0 \end{bmatrix} \ : \ a, b, c \in \{0,1\}^8 \right\}$$

The corresponding parameters are:

$$\begin{array}{llll}
\vec{p} = 2^{-22} & \overleftarrow{p} = 2^{-24} & |\mathcal{D}^0_{\text{in}}| = 2^{32} & |\mathcal{D}^0_{\text{out}}| = 2^{34} \\
\vec{\overline{p}} = 2^{-22} & \overleftarrow{\overline{p}} = 2^{-22} & |\overline{\mathcal{D}}^0_{\text{in}}| = 2^{34} & |\mathcal{D}^0_{\text{out}}| = 2^{34} \\
\vec{q} = 2^{-32} & \overleftarrow{q} = 2^{-24} & |\mathcal{D}^1_{\text{in}}| = 2^{32} & |\mathcal{D}^1_{\text{out}}| = 2^{24}
\end{array}$$

For each quartet, the pair $(C, \overline{C})$ suggests two values for $k_6[13] + k_6[10]$, and $C', \overline{C'}$ also suggests two values. Therefore a quartet suggests on average $\ell = 2^{-6}$ values for $\kappa = 8$ bits of key. Using the analysis of subsection 3.2, we obtain:

$$\begin{array}{ll}
p_b = \vec{p} \cdot \overleftarrow{\overline{p}} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-124} & \tilde{\sigma} = 2^{-16} \\
p_w = |\overline{\mathcal{D}}^0_{\text{in}}| \times 2^{-n} \times \ell \cdot 2^{-\kappa} = 2^{-108} & Q = \mathcal{O}(2^{140})
\end{array}$$

To obtain a high probability of success we use $Q = 2^{145}$, *i.e.* $D = 2^{90}$. Since $\tilde{\sigma} \ll 1$, the counter distribution of the right key can be approximated to the normal distribution $\mathcal{N}(2^{37} + 2^{21}, 2^{37})$ while wrong key counters distributions can be approximated to $\mathcal{N}(2^{37}, 2^{37})$. We expect the correct key to be ranked first with very high probability ($P_S > 0.99$ using the formula from [35]).

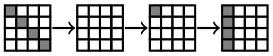
The time complexity is dominated by the oracle queries: for each structure of 2^{56} plaintexts/ciphertexts, we filter $2^{56} \times 2^{55} \times |\overline{\mathcal{D}}^0_{\text{in}}| \times 2^{-n} = 2^{17}$ candidate quartets with $\overline{P} + \overline{P}' \in \overline{\mathcal{D}}^0_{\text{in}}$, and the time to extract the key candidates is negligible. We can repeat the attack to recover up to 16 key bytes in different positions, with a complexity of $D = T = 2^{94}$ (but only 12 recovered bytes are linearly independent).

5 Application to 8-round Kiasu-BC

Kiasu-BC [26] is an instance of the TWEAKEY framework [27], reusing the AES round function in a tweakable block cipher. The 6-round boomerang attack on the AES can be extended to 8-round Kiasu-BC by taking advantage of the tweak input to cancel state differences in order to have one inactive round in the upper and lower trails. Indeed, the best known attack on Kiasu-BC is an 8-round attack with complexity 2^{103} in data and time [18] following this idea. Following our framework, we improve this attack with a better use of structures.

Truncated Boomerang. Since we use a tweak difference Δ_{tw}, we slightly generalize our truncated differential framework to allow a set of tweak differences

$\mathcal{D}_{\text{tw}}$. We start from a 4-round truncated trail $(\mathcal{D}_{\text{in}}, \mathcal{D}_{\text{tw}}) \xrightarrow[E]{p} \mathcal{D}_{\text{out}}$ with probability 2^{-32}, similar to the 3-round trail of previous sections:



$\mathcal{D}_{\text{tw}}$ is the set of differences active in the first cell of the tweak; following the tweakey schedule of Kiasu-BC, this results in a tweakey difference in $\mathcal{D}_{\text{tw}}$ at each round. We obtain an 8-round boomerang with two 4-round differentials:

$$\begin{array}{lllll} \vec{p} = 2^{-32} & \overleftarrow{p} = 2^{-32} & |\mathcal{D}^0_{\text{in}}| = 2^{32} & |\mathcal{D}^0_{\text{out}}| = 2^{32} & |\mathcal{D}^0_{\text{tw}}| = 2^{8} \\ \vec{q} = 2^{-32} & \overleftarrow{q} = 2^{-32} & |\mathcal{D}^1_{\text{in}}| = 2^{32} & |\mathcal{D}^1_{\text{out}}| = 2^{32} & |\mathcal{D}^1_{\text{tw}}| = 2^{8} \end{array}$$

Following the analysis of the AES attack in subsection 3.2, we deduce on average $\ell = 2^{-32}$ candidates of $\kappa = 64$ key bits per quartet. Therefore, we obtain

$$\begin{aligned} p_b &= \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-160} \\ p_w &= |\mathcal{D}^0_{\text{in}}|/2^n \times \ell \times 2^{-\kappa} = 2^{-192} \qquad\qquad \tilde{\sigma} = 2^{32} \end{aligned}$$

Since $\tilde{\sigma} \gg 1$, we only need a few right quartets. Taking $\mu = 4$, we obtain an attack with $Q = 2^{162}$ quartets. We take advantage of the tweak to build larger structures (iterating over the tweak and data inputs), of size $|\mathcal{D}^0_{\text{in}}| \cdot |\mathcal{D}^0_{\text{tw}}| \cdot |\mathcal{D}^1_{\text{out}}| \cdot |\mathcal{D}^1_{\text{tw}}| = 2^{80}$. Thus we only need 8 structures, with data complexity $D = 2^{83}$. In each structure of 2^{80} elements, we expect 2^{63} quartets with $\overline{P} + \overline{P}' \in \mathcal{D}^0_{\text{in}}$, therefore the time complexity for the key recovery is negligible, and $T = D = 2^{83}$.

Success Probability. There are 2^{66} quartets with $\overline{P} + \overline{P}' \in \mathcal{D}^0_{\text{in}}$, suggesting on average 2^{-32} key candidates each; hence a total of 2^{34} candidates for 64 bits of key. We keep key candidates whose counter reaches 2 or more. Modeling counters for wrong keys with a Poisson distribution with $\lambda = 2^{-30}$, the probability for a specific wrong key counter to be at least 2 is $1 - e^{-\lambda}(1 + \lambda) \approx 2^{-61}$; therefore we expect to keep 8 wrong keys. On the other hand, the counter for the right key follows a Poisson distribution with $\lambda = 4$. It reaches a value of 2 or more with probability ≈ 0.9.

As in the AES attacks, we recover the full key by repeating the attack with $\mathcal{D}^0_{\text{in}}$ in a different diagonal. Taking advantage of the recovered values of the last round key, this adds a negligible complexity.

6 Application to TNT-AES

TNT-AES is a tweakable block cipher reusing the AES round function published at Eurocrypt 2020 [1]. It is part of the Tweak-aNd-Tweak framework, building a tweakable block cipher $\tilde{E}$ from a block cipher E:

$$\tilde{E}_{K_0,K_1,K_2} : P, T \mapsto C = E_{K_2}\Big(T + E_{K_1}\big(T + E_{K_0}(P)\big)\Big)$$

In order to improve its efficiency, TNT-AES uses a 6-round AES as building block E. The designers of TNT proved its security up to $2^{2n/3}$ queries, and conjectured a higher security bound. Later work [25] proved the bound to be at least $\Omega(2^{3n/4})$ queries, and exhibited a distinguisher with $\mathcal{O}(\sqrt{n} \cdot 2^{3n/4})$ queries.

Truncated Boomerang. Our attack focuses on the middle cipher E_{K_1}, between both tweak additions. In order to skip the initial and final ciphers E_{K_0} and E_{K_2}, we introduce differences in the tweak, instead of introducing them in the plaintext and ciphertext. We fix a plaintext P, and consider four tweaks $T, T', \overline{T}, \overline{T}'$ to create quartets as follows:

1. Query $C = \tilde{E}(P, T)$ and $C' = \tilde{E}(P, T')$
2. Query $\overline{P} = \tilde{E}^{-1}(C, \overline{T})$ and $\overline{P}' = \tilde{E}^{-1}(C', \overline{T}')$
3. Detect when $\overline{P} = \overline{P}'$

We denote the inputs and outputs of E_{K_1} as X and Y, with $Y = E_{K_1}(X)$:

$$
\begin{array}{llll}
X = E_{K_0}(P) + T & X' = E_{K_0}(P) + T' & \overline{X} = E_{K_0}(\overline{P}) + \overline{T} & \overline{X}' = E_{K_0}(\overline{P}') + \overline{T}' \\
Y = E_{K_2}^{-1}(C) + T & Y' = E_{K_2}^{-1}(C') + T' & \overline{Y} = E_{K_2}^{-1}(C) + \overline{T} & \overline{Y}' = E_{K_2}^{-1}(C') + \overline{T}'
\end{array}
$$

When $\overline{P} = \overline{P}'$, we have a boomerang quartet for E_{K1} with differences

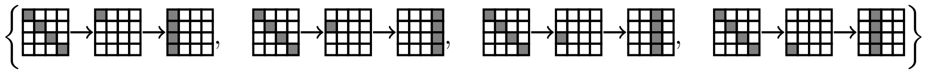
$$
\begin{array}{rr}
X + X' = T + T' = \mathit{\Delta}_{\text{in}} & \overline{X} + \overline{X}' = \overline{T} + \overline{T}' = \mathit{\Delta}'_{\text{in}} \\
Y + \overline{Y} = T + \overline{T} = \nabla_{\text{out}} & Y' + \overline{Y}' = T' + \overline{T}' = \nabla'_{\text{out}}
\end{array}
$$

When using a truncated boomerang trail (with a fixed P and a set of tweaks), there are two important limitations compared to the previous attacks:

- We only detect when the difference $\overline{X} + \overline{X}'$ matches exactly $\overline{T} + \overline{T}'$, instead of detecting a set of differences $\mathcal{D}^0_{\text{in}}$. The boomerang trail probability decreases.
- We necessarily have $\mathit{\Delta}_{\text{in}} + \mathit{\Delta}'_{\text{in}} = \nabla_{\text{out}} + \nabla'_{\text{out}}$. For the 6-round AES truncated boomerang trail of figure 3, this implies $\mathit{\Delta}_{\text{in}} = \mathit{\Delta}'_{\text{in}}$ and $\nabla_{\text{out}} = \nabla'_{\text{out}}$. Therefore, we cannot take advantage of structures on the ciphertext side.

Nonetheless, truncated boomerangs can be used with structures of tweaks on the plaintext side, and the analysis of the middle rounds as truncated differentials significantly reduces the complexity compared to the analysis of [1].

Upper Differential. We use the same collection of 4 upper trails as in our optimized attack on AES:



We have the following parameters

$$
\vec{p} = 2^{-22} \qquad \overleftarrow{p} = 2^{-24} \qquad |\mathcal{D}^0_{\text{in}}| = 2^{32} \qquad |\mathcal{D}^0_{\text{out}}| = 2^{34}
$$

For the return trail, we must hit a fixed $\overline{T} + \overline{T'} = \mathit{\Delta}^0_{\text{in}}$:

$$
\vec{\overline{p}} = 2^{-22} \qquad \overleftarrow{\overline{p}} = 2^{-56} \qquad |\overline{\mathcal{D}}^0_{\text{in}}| = 2^{0} \qquad |\mathcal{D}^0_{\text{out}}| = 2^{34}
$$

Lower Differential. Since we cannot use structures on the ciphertext side, we use a fixed value Δ^1_{out} to maximize the probability of the trail. We observe that in an AES column, the transition $\delta \to (*, 0, 0, 0)$ through a layer of inverse S-Boxes followed by inverse MixColumns happens with probability $2272/2^{32} \approx 2^{-20.85}$ with $\delta = (L(\beta/2), L(\beta), L(\beta), L(\beta/3))$, with L the linear transform inside the AES S-Box (see the full version [6] for more details). We choose $\Delta^1_{\text{out}} = \mathsf{MixColumns}(\mathsf{ShiftRows}(\delta))$:

$$\vec{q} = 2^{-52.85} \qquad \overleftarrow{q} = 2^{-20.85} \qquad |\mathcal{D}^1_{\text{in}}| = 2^{32} \qquad |\mathcal{D}^1_{\text{out}}| = 2^0$$

Boomerang Trail Probability. We obtain:

$$p_b = \vec{p} \cdot \overleftarrow{\overline{p}} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-151.7} \qquad p_\$ = |\overline{\mathcal{D}}^0_{\text{in}}|/2^n = 2^{-128}$$

As shown in the full version, we obtain a slightly better probability p_b by carefully analyzing the boomerang, and correlation between the sides:

$$p_b = 2^{-151.4}$$

It is not possible to recover actual key material with this attack because X is unkown. However, we can use $E_{K_0}(P) + K_1$ as an equivalent subkey if all queries are made with the same P. Using the pair (X, X') we extract $\ell = 2^{10}$ candidates for $\kappa = 32$ key bits. Unfortunately, we cannot use the pairs $(Y, \overline{Y})$ for filtering on the ciphertext side since the unkown value Y is different in each quartet. Similarly, the pair $(\overline{X}, \overline{X}')$ is unusable for key extraction. Therefore,

$$p_b = 2^{-151.4} \qquad p_w = p_\$ \times \ell \times 2^{-\kappa} = 2^{-150} \qquad \tilde{\sigma} = 2^{-1.4}$$

With $\tilde{\sigma} < 1$, we need $Q = c \cdot \tilde{\sigma}^{-1} \cdot p_b^{-1}$ with a small constant c; we take $c = 64$, $Q = 2^{158.8}$. Since we have structures of size 2^{32}, this corresponds to $D = 2^{127.8}$.

Distinguisher. With $2^{127.8}$ queries we obtain a distinguisher between TNT-AES (using 6-round AES as the building block) and a PRP (or TNT using a PRP). This obviously does not threaten the security of TNT-AES, but we believe that it is an interesting use case showing that a 6-round boomerang distinguisher can be extended to a larger scheme, even if the attack is marginal.

In order to minimize the number of queries, we use the 255 possible values of $\Delta^1_{\text{out}} = (L(\beta/2), L(\beta), L(\beta), L(\beta/3))$ with $\beta \in \mathbb{F}_{256} \setminus \{0\}$, so that each encryption query is amortized: we obtain $2^{158.8}$ quartets with $2^{127.8}/255$ encryption queries and $2^{127.8}$ decryption queries. After collecting the quartets, we expect that the counter corresponding to the right key follows the distribution $\mathcal{N}(2^{8.8} + 2^{7.4}, 2^{8.8})$ while counters for the wrong keys follow the distribution $\mathcal{N}(2^{8.8}, 2^{8.8})$ (the distance between the expected values is 8 times the standard deviation).

We obtain a distinguisher by observing whether the maximum counter is higher than a threshold $t = 2^{8.8} + 7 \times 2^{4.4}$. The probability that all counters for wrong keys are lower than t is $\Phi(7)^{2^{32}} \approx 0.995$, therefore the probability of false positive is 0.005. The probability that the counter for the right key is higher than t is $\Phi(1) = 0.84$ so the probability of false negative is 0.16.

Finally, we can increase the success rate by running three attacks in parallel using three input sets $\mathcal{D}^0_{\text{in}}$, $\mathcal{D}'^0_{\text{in}}$, $\mathcal{D}''^0_{\text{in}}$ on three different diagonals. Using superstructures of 2^{96} values, we run all three attacks with the same queries, and generate counters for three sets of 2^{32} equivalent keys. Using a threshold of $t = 2^{8.8} + 7.1 \times 2^{4.4}$, we keep the probability of false positive below 1%, while the probability that at least one of the three counters corresponding to right keys is higher than the threshold increases to 99%.

TNT with 5-round AES. A reduced version of TNT-AES with 5 AES rounds can be attacked more efficiently, using a probability-one truncated differential for one half of the cipher. In the full version [6] we give a distinguisher with 2^{76} queries, and a key-recovery with complexity 2^{87}.

7 Modeling the framework using MILP

MILP modelling encodes a cryptographic problem as a Mixed Integer Linear Programming problem, and uses an available solver to find optimal solutions. This method was applied to the search of boomerang distinguisher on Deoxys by Cid *et al.* [12]. Their MILP model encodes the activity of each state byte with a binary variable that equals 1 if its corresponding byte is active and 0 if not, and that is constrained depending on the activity pattern of Deoxys operations. In order to build a boomerang trail, their model includes two separate differential trails with two overlapping rounds in the middle (in order to account for the ladder switch and the BCT analysis). The objective function to minimize is roughly the number of active S-Boxes, *i.e.* the sum of all variables representing the activity of S-Box input (or output) bytes.

After generating the optimal boomerang template, they instantiate active bytes with concrete differences that maximize S-Box transition probabilities. An important contribution of their work is an analysis of the degrees of freedom of the tweakey differences. Their MILP model counts the number of linear relations between the tweakey differences and ensures that at least one degree of freedom remains in the final trail, otherwise it is unlikely to find concrete differences for the tweakey.

In 2019, Zhao et al. [42,43] improved this MILP model by adding two extra rounds at the end of the lower trail, containing truncated differences.

Our MILP model. Previous works [12,34,43] showed that the best attack is not always obtained with the best distinguisher. Therefore we follow the same high-level approach as in [31]: our main objective is to cover the full boomerang attack with the MILP model. We give an overview of the model in the full version [6], and for more details the full code is available [7]. Our model is based on [12]; the main improvement is to allow 4 possible states for each byte of the trail, instead of only 2 (active or inactive):

- inactive, with a zero difference, denoted as □;
- active with a *fixed* non-zero difference, denoted as 🟦;
- active with an *unknown* (truncated) difference, denoted as ■.
- active with an *equal* (but unknown) difference for both pairs, denoted as ▣.

From these possible states, we can deduce the parameters of the attacks, including the structure size, which allows us to ask the MILP solver to minimize the formula for the data complexity of the attack given in subsection 3.2.

Bytes with *equal* differences encode relations between the two different pairs that follow the same trail, rather than properties of a trail by itself. This allows the MILP model to capture trails like the 6-round AES boomerang of Figure 3. The model does not encode linear relations between active bytes (*e.g.* the set of differences for w_2 of Figure 3 is active on all bytes but has size 2^{32}), but using this type of constraint is sufficient in many cases because it is propagated through the linear layer.

Our model generates boomerang trails without instanciating the differences. In order to derive concrete attacks from the trail, we instantiate the trail with differences that optimize the different S-Box transitions.

Limitations of the MILP model. Our model handles only fixed differences in the tweakey. More importantly, although our model takes into account the ladder switch to compute the boomerang connection probability, it does not accurately compute the connection probability. Some boomerang trails given in a previous version of this paper were even found incompatible by Song et al. [41]. More generally, some boomerang trails returned by the MILP are not instantiable. When that happens in practice, we modify the boomerang trail squeleton or we generate a different trail using the MILP solver.

To instantiate boomerang trails generated by the MILP, we use the tables DDT, BCT [13], UBCT, LBCT and EBCT, introduced in [36,40,16]. In order to ensure that the trails are not incompatible, we verified experimentally the probability of the middle rounds [7]. For each trail, we indicate the rounds that have been checked, with the theoretical probability p_{th} for the middle rounds, and the experimental value observed p_{exp}. In some attacks, the experimental probability slightly differs from the theoretical one; we deduce an adjusted trail probability $\tilde{p}_b$ that is used to calculate the complexity of the attack.

7.1 Results on AES-128 and Kiasu-BC

We use the MILP model to search for attacks on AES-128 and Kiasu-BC and compare them with the results of the previous sections.

On AES-128, the model returns a trail corresponding to Figure 3 with a full *equal* state in w_2. This confirms the optimality of our truncated trail within our framework. However, the model does not handle multiple trails, so that it cannot suggest the improved attack of section 4.

On Kiasu-BC, the model cannot find the attack of section 5, because it does not handle truncated differences in the tweak. With fixed difference in the tweak, the best trail found by the solver is unfortunately not instantiable, because of incompatibilities in the middle rounds. The solver nevertheless ensures that no attack with complexity less than 2^{80} exist in that framework.

8 Application to Deoxys-BC

Deoxys-BC [28] is a tweakable block cipher of the TWEAKEY framework [27] based the AES round function, on which the best known attacks are based on boomerangs [12,34,42,43]. Due to the large choice of tweakey differences, finding the best truncated boomerang trails manually is a tedious work. Instead, we use our MILP model of section 7.

In the single tweakey model, the analysis is exactly that of the AES, and the best known boomerang attack is given in section 4.

In the related tweakey model, the attacker can insert differences in some of the tweakey words TK^i. Depending on the tweak size and differences used, this can be either a single-key attack with chosen tweaks, or a related-key attack. We denote as RTKr a model with differences in r 128-bit states, corresponding to:

- RTK1: single-key attacks on any variant with at least 128 bits of tweak.
- RTK2: single-key attacks on Deoxys-BC-384 with 256 bits of tweak, or related-key attacks on Deoxys-BC-256.
- RTK3: related-key attacks on Deoxys-BC-384.

For 13-round Deoxys-BC in the RTK3 model, we selected a non-optimal trail in terms of data complexity, which was better in time complexity. For 8-round and 9-round Deoxys-BC in the RKT1 model and 10-round Deoxys-BC in the RTK2 model, we modified the squeleton of the trail returned by the MILP because the original one was not instantiable. During other difference instantiations, we sometimes applied slight manual improvements. For instance, for minor gains, we introduced state changing bytes: fixed on the forward trail but truncated on the return trail.

Description of the attacks. In the related-tweakey model, the attacker queries two sets of $|\mathcal{D}_{\text{in}}^0|$ plaintexts (under tweaks T and T'), and each ciphertext is shifted $|\mathcal{D}_{\text{out}}^1|$ times with a new tweak ($\overline{T}$ and $\overline{T}'$ respectively). In total a structure of size $S = |\mathcal{D}_{\text{in}}^0| \times |\mathcal{D}_{\text{out}}^1|$ requires $2S$ decryption queries (or $4S$ if $|\mathcal{D}_{\text{out}}^1| = 1$) and generates S^2 quartets.

The values of ℓ and κ mentioned on the figures are the one used in our attacks, corresponding either to a 1-round or to a 2-round key-recovery. Each attack recovers a partial key, aiming for a success rate of 1/2, comparable to previous analysis; we assume that the rest of the key can be recovered efficiently afterwards. When $\tilde{\sigma} \gg 1$, the number μ of right quartets required varies from 1 to 4. In particular, if $p_b \gg \ell \cdot p_\$$, we expect no wrong quartet and $\mu = 1$ suffices, else several right quartets are needed to get the correct key ranked first.

10-round Deoxys-BC in the RTK2 model (Figure 4). Query one partial structure of $2^{93.2}$ ciphertexts, so that on average, $\mu = 2^{93.2+93.2} \cdot \tilde{p}_b = 2$ quartets follow the trail. For each element of the structure, deduce on average 1 candidate for 30 bits of key on the plaintext side: 1 candidate for $tk_0[4, 9, 14]$ and 1 representant of the 4 possible candidates each for $tk_0[3]$[1]. In total, there are on average $2^{93.2+93.2-56-30} = 2^{100.4}$ candidate quartets matching on the ciphertext bytes with a known difference and on the key candidate.

For each quartet, retrieve 2^{-8} candidates for $tk_{10}[9]$ with 2 table accesses. This costs $2^{101.4}$ table accesses, and since an encryption makes $10 \times 16 = 2^{7.3}$ S-Box calls, this step costs $2^{94.1}$ equivalent encryptions. For each of the $2^{100.4-8} = 2^{92.4}$ remaining quartets, retrieve on average 2^{-32} candidates of $tk_{10}[0, 1, 2, 3, 4, 5, 6, 7]$. Finally, recover 2^{-16} candidates for $tk_9^{eq}[1, 6]$. There remains $2^{92.4-32-16} = 2^{44.4}$ quartets with a 118-bit key candidate. The only candidate suggested twice is expected to be the right candidate. The time complexity is $2^{94.2} + 2^{94.1} \approx 2^{95.2}$, thus $(D, T, M) = (2^{94.2}, 2^{95.2}, 2^{94.2})$.

11-round Deoxys-BC in the RTK2 model. The MILP solver did not return a pertinent trail for this key setting. Instead, we use the 10-round trail and append a round at the beginning. First, query the full encryption codebook with $\overline{T}, \overline{T}'$ and store it. Then, guess the full tk_0. Perform the 10-round attack, by using the same ciphertext structure for each guess of tk_0 and simulating encryption queries with fetches in the codebook. We chose $\mu = 4$ and for each key guess, the 10-round attack with partial structures of $2^{93.7}$ elements gives $2^{45.4}$ candidates for 118 bits, for a time complexity of $2^{95.1}$. If we suppose that a fetch to the codebook costs an encryption in time complexity, we end up with $T = 2^{128}(2^{94.7} + 2^{95.1}) = 2^{223.9}$. The probability that one of the counters is at least 4 is $2^{-295+116+128} = 2^{-51}$, so in average, the correct key is ranked first. This gives $(D, T, M) = (2^{129}, 2^{223.9}, 2^{129})$.

13-round Deoxys-BC in the RTK3 model (Figure 5). Query a partial structure of $2^{125.65}$ plaintexts. On average, $\mu = 2^{125.65} \cdot 2^{125.65} \cdot \tilde{p}_b = 4$ quartets follow the trail.

1. For each element of the structure, retrieve the representant k of the 2^6 possible key values of $tk_{13}[13, 14, 15]$ that satisfy the transition $y_{12} \to x_{12}$. k defines 18 key bits.
2. Guess the value of the tweakey material $tk_0[2, 7, 8, 13]$. Set $\delta = \texttt{0x7e42c465}$ and $\delta_{\text{in}} = \texttt{0x00007a00}$, and look for collisions between:

$$\begin{aligned} v &= y_0[2,7,8,13] \quad \| \, \overline{y_0}[2,7,8,13] \quad \| \, \overline{P}[0,5,10,15] \qquad \| \, k \\ v' &= y_0'[2,7,8,13] + \delta \, \| \, \overline{y_0'}[2,7,8,13] + \delta \, \| \, \overline{P'}[0,5,10,15] + \delta_{\text{in}} \, \| \, k' \end{aligned}$$

[1] S-Box 3 on the plaintext side has two pairs $(x, x + \delta), (x', x' + \delta)$ following the transition fixed by the trail. Instead of listing four key candidates, we identify one of the 2^6 cosets of $\langle \delta, x + x' \rangle$.

This step costs $2^{32} \cdot 2 \cdot 2^{125.65} = 2^{158.65}$ in time complexity. On average, $2^{125.65} \cdot 2^{125.65} \cdot 2^{-114} = 2^{137.3}$ quartets remain for each $tk_0[2,7,8,13]$ ($2^{169.3}$ in total).

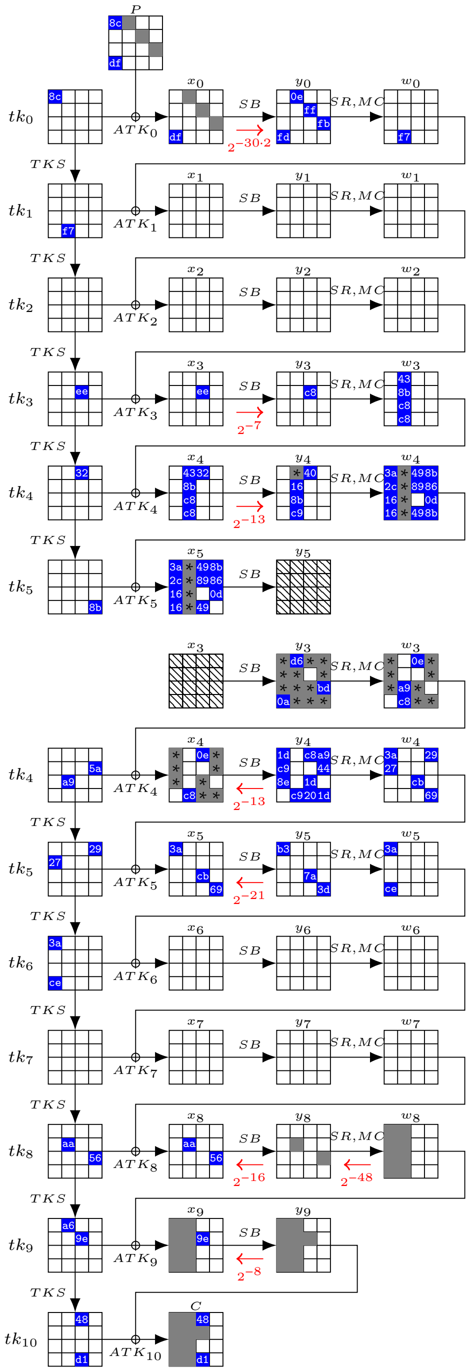
3. For each quartet, retrieve $2^{7+7-32} = 2^{-18}$ values of $tk_0[3,4,9,14]$ such that the difference in $w_0[4]$ is compatible with the S-Box transition in the next round. In order to minimize the complexity, first deduce the $2^{7+7-8} = 2^6$ pairs of column differences compatible with a key candidate for $tk_0[3]$, by only checking the first S-Box. Then, deduce the $2^{6-8} = 2^{-2}$ pairs of columns compatible with a key candidate for $tk_0[4]$ with the second S-Box. Finally deduce $tk_0[9,14]$.
   This step requires $2^8 + 2^7 = 2^{8.6}$ table accesses per quartet, therefore a total of $2^{8.6+169.3} = 2^{177.9}$ accesses; and $2^{32+137.3-18} = 2^{151.3}$ quartets remain.
4. For each quartet, retrieve $2^{7+7-32} = 2^{-18}$ values of $tk_0[1,6,11,12]$ and $2^{24+24-32} = 2^{16}$ key candidates for $tk_{13}[8,9,10,11]$. Recover $x_{12}[8,9,10,11]$ and the difference in $y_{11}[2,7,8]$. Retrieve 2^{-24} candidates for $tk_{12}^{eq}[2,7,8]$. $2^{151.3-18+16-24} = 2^{125.3}$ quartets remain.
5. For each quartet, recover the difference in $x_1[4,14]$ and the value of $w_0[4,14]$ from the known key bytes of tk_0. Retrieve $2 \cdot 2 \cdot 2^{-8} = 2^{-6}$ values of $tk_1[4]$ and 2^{-6} values for $tk_1[14]$ (2 candidates are deduced per pair because the differences are already compatible). $2^{125.3} \cdot 2^{-12} = 2^{113.3}$ quartets remain.
6. Eventually, each of the $2^{113.3}$ quartets determines in average 1 candidate of $18+32+32+32+32+24+16 = 186$ bits. We model a wrong counter with a poisson distribution with $\lambda = 2^{-72.7}$. The probability that any wrong counter is at least 3 is $(1 - e^{-\lambda}(1+\lambda+\lambda^2/2)) \cdot 2^{184} \approx 2^{-34.7}$. The correct counter follows the poisson distribution with $\lambda = 4$ and it is at least 3 with probability 0.76. Therefore, the success probability of this attack is 0.76.

Complexity analysis. The time complexity is dominated by the $2^{177.9}$ table accesses of step 3. An encryption of 13-round Deoxys-BC has 16×13 S-Boxes, so the time complexity is equivalent to $2^{177.9}/208 = 2^{170.2}$ encryptions. Thus $(D,T,M) = (2^{126.7}, 2^{170.2}, 2^{126.7})$.

Other variants. Due to space constraints, the attacks found on other variants of Deoxys-BC are detailed in the full version of this paper [6].

9 Conclusion

In this paper, we develop a framework for truncated boomerang attacks. Instead of extending a distinguisher with additional key-recovery rounds, we integrate them inside the distinguisher. This results in a simple and generic formula for the data complexity of the attack, while the classical approach to add rounds strongly depends on the shape of input/output differences of the distinguisher. In particular, the formula can be integrated in a MILP model, leading to better results than a separate search for distinguishers and attacks.



■ Truncated bytes

01 Active bytes

▨ Switched bytes

⊛ Truncated equal bytes

$$\Delta TK^1 = \begin{bmatrix} 5a & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \end{bmatrix} \qquad \nabla TK^1 = \begin{bmatrix} 00 & 00 & 00 & 00 \\ 00 & 74 & 00 & 00 \\ 00 & 00 & 00 & 2c \\ 00 & 00 & 00 & 00 \end{bmatrix}$$

$$\Delta TK^2 = \begin{bmatrix} d6 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 \end{bmatrix} \qquad \nabla TK^2 = \begin{bmatrix} 00 & 00 & 00 & 00 \\ 00 & 62 & 00 & 00 \\ 00 & 00 & 00 & b8 \\ 00 & 00 & 00 & 00 \end{bmatrix}$$

$\overleftarrow{q} = 2^{-106}$ $\quad$ $\vec{p} = 2^{-30}$

$\vec{q} = 1$ $\quad$ $r = 2^{-20}$

$p_b = 2^{-186}$ $\quad$ $p_\$ = 2^{-56}$

$\kappa = 116$ $\quad$ $\ell = 2^{-84}$

$\tilde{\sigma} = 2^{70}$ $\quad$ $\mu = 2$

$|\mathcal{D}_{in}^{up}||\mathcal{D}_{out}^{low}| = 2^{96}$ $\quad$ $D = 2^{94.5}$

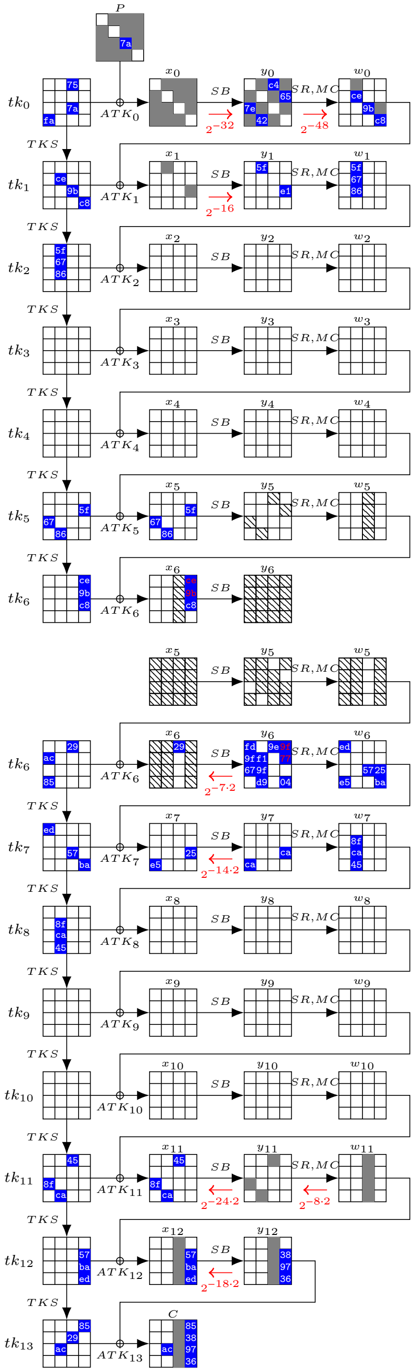
$T = 2^{95.6}$ $\quad$ $M = 2^{94.5}$

Verification $\quad x_6 \rightarrow w_2 \rightarrow x_6$
(forcing $y_5 \rightarrow x_5$)

$p_{\mathrm{th}} = 2^{-33}$ $\quad$ $p_{\exp} = 2^{-32.4}$

$\tilde{p}_b = 2^{-185.4}$

Fig. 4. Truncated boomerang attack on 10-round Deoxys-BC in the RTK2 model, starting from the ciphertext side. This attack succeeds with probability 1/2.
Middle rounds are analyzed with UBCT, LBCT and EBCT (probabilities on the trail).



▩ Truncated bytes
01 Active bytes
▨ Switched bytes

$$\Delta TK^1 = \begin{bmatrix} 00 & 00 & a9 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 8c & 00 \\ 3c & 00 & 00 & 00 \end{bmatrix} \qquad \nabla TK^1 = \begin{bmatrix} 00 & 00 & 00 & 00 \\ 00 & 53 & 00 & 00 \\ 00 & 7b & 00 & 00 \\ 00 & 28 & 00 & 00 \end{bmatrix}$$

$$\Delta TK^2 = \begin{bmatrix} 00 & 00 & 53 & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & 4f & 00 \\ 82 & 00 & 00 & 00 \end{bmatrix} \qquad \nabla TK^2 = \begin{bmatrix} 00 & 00 & 00 & 00 \\ 00 & 34 & 00 & 00 \\ 00 & c8 & 00 & 00 \\ 00 & fc & 00 & 00 \end{bmatrix}$$

$$\Delta TK^3 = \begin{bmatrix} 00 & 00 & 8f & 00 \\ 00 & 00 & 00 & 00 \\ 00 & 00 & b9 & 00 \\ 44 & 00 & 00 & 00 \end{bmatrix} \qquad \nabla TK^3 = \begin{bmatrix} 00 & 00 & 00 & 00 \\ 00 & d4 & 00 & 00 \\ 00 & 32 & 00 & 00 \\ 00 & e6 & 00 & 00 \end{bmatrix}$$

$\vec{p} = 2^{-96}$ $\qquad \overleftarrow{q} = 2^{-71}$

$\overleftarrow{p} = 1$ $\qquad r = 2^{-10.8}$

$p_b = 2^{-248.8}$ $\qquad p_{\$} = 2^{-32}$

$\kappa = 186$ $\qquad \ell = 2^{-106}$

$\tilde{\sigma} = 2^{75.2}$ $\qquad \mu = 4$

$|\mathcal{D}_{in}^{up}||\mathcal{D}_{out}^{low}| = 2^{128}$ $\qquad D = 2^{126.4}$

$T = 2^{169.7}$ $\qquad M = 2^{126.4}$

Verification $\quad x_8 \to w_4 \to x_8$
(forcing $y_7 \to x_7$)

$p_{\text{th}} = 2^{-38.8}$ $\qquad p_{\text{exp}} = 2^{-39.3}$

$\tilde{p}_b = 2^{-249.3}$

Fig. 5. Truncated boomerang attacks on 13-round Deoxys-BC in the RTK3 model, starting from the plaintext side. This attack succeeds with probability 0.76. Middle rounds are analyzed with the ladder switch and single BCT (probability r).

Acknowledgement. We would like to thank the authors of [12] for providing the code they used to generate the MILP programs. This work was supported by the French Ministry of Defence (AID), and by the French Agence Nationale de la Recherche (ANR), under grant ANR-20-CE48-0017 (project SELECT).

References

1. Bao, Z., Guo, C., Guo, J., Song, L.: TNT: How to tweak a block cipher. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 641–673. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45724-2_22
2. Bao, Z., Guo, J., List, E.: Extended truncated-differential distinguishers on round-reduced AES. IACR Trans. Symm. Cryptol. **2020**(3), 197–261 (2020). https://doi.org/10.13154/tosc.v2020.i3.197-261
3. Bar-On, A., Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: Improved key recovery attacks on reduced-round AES with practical data and memory complexities. Journal of Cryptology **33**(3), 1003–1043 (Jul 2020). https://doi.org/10.1007/s00145-019-09336-w
4. Bardeh, N.G.: A key-independent distinguisher for 6-round AES in an adaptive setting. Cryptology ePrint Archive, Report 2019/945 (2019), https://eprint.iacr.org/2019/945
5. Bardeh, N.G., Rønjom, S.: The exchange attack: How to distinguish six rounds of AES with $2^{88.2}$ chosen plaintexts. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 347–370. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_12
6. Bariant, A., Leurent, G.: Truncated boomerang attacks and application to AES-based ciphers. Cryptology ePrint Archive, Report 2022/701 (2022), https://eprint.iacr.org/2022/701
7. Bariant, A., Leurent, G.: Truncated boomerang attacks and application to AES-based ciphers — Additional data. https://github.com/AugustinBariant/Truncated_boomerangs (2023)
8. Biham, E., Dunkelman, O., Keller, N.: The rectangle attack - rectangling the Serpent. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 340–357. Springer, Heidelberg (May 2001). https://doi.org/10.1007/3-540-44987-6_21
9. Biham, E., Dunkelman, O., Keller, N.: New results on boomerang and rectangle attacks. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 1–16. Springer, Heidelberg (Feb 2002). https://doi.org/10.1007/3-540-45661-9_1
10. Biryukov, A.: The boomerang attack on 5 and 6-round reduced aes. In: International Conference on Advanced Encryption Standard. pp. 11–15. Springer (2004)
11. Boura, C., Lallemand, V., Naya-Plasencia, M., Suder, V.: Making the impossible possible. Journal of Cryptology **31**(1), 101–133 (Jan 2018). https://doi.org/10.1007/s00145-016-9251-7
12. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: A security analysis of Deoxys and its internal tweakable block ciphers. IACR Trans. Symm. Cryptol. **2017**(3), 73–107 (2017). https://doi.org/10.13154/tosc.v2017.i3.73-107
13. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: Boomerang connectivity table: A new cryptanalysis tool. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 683–714. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78375-8_22

14. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher Square. In: Biham, E. (ed.) FSE'97. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg (Jan 1997). https://doi.org/10.1007/BFb0052343
15. Daemen, J., Rijmen, V.: The design of Rijndael, vol. 2. Springer (2002)
16. Delaune, S., Derbez, P., Vavrille, M.: Catching the fastest boomerangs application to SKINNY. IACR Trans. Symm. Cryptol. **2020**(4), 104–129 (2020). https://doi.org/10.46586/tosc.v2020.i4.104-129
17. Dobraunig, C., Eichlseder, M., Mendel, F.: Square attack on 7-round kiasu-BC. In: Manulis, M., Sadeghi, A.R., Schneider, S. (eds.) ACNS 16. LNCS, vol. 9696, pp. 500–517. Springer, Heidelberg (Jun 2016). https://doi.org/10.1007/978-3-319-39555-5_27
18. Dobraunig, C., List, E.: Impossible-differential and boomerang cryptanalysis of round-reduced kiasu-BC. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 207–222. Springer, Heidelberg (Feb 2017). https://doi.org/10.1007/978-3-319-52153-4_12
19. Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: The retracing boomerang attack. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 280–309. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45721-1_11
20. Dunkelman, O., Keller, N., Shamir, A.: A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 393–410. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_21
21. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved cryptanalysis of Rijndael. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 213–230. Springer, Heidelberg (Apr 2001). https://doi.org/10.1007/3-540-44706-7_15
22. Grassi, L.: MixColumns properties and attacks on (round-reduced) AES with a single secret S-box. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 243–263. Springer, Heidelberg (Apr 2018). https://doi.org/10.1007/978-3-319-76953-0_13
23. Grassi, L., Rechberger, C., Rønjom, S.: Subspace trail cryptanalysis and its applications to AES. IACR Trans. Symm. Cryptol. **2016**(2), 192–225 (2016). https://doi.org/10.13154/tosc.v2016.i2.192-225, https://tosc.iacr.org/index.php/ToSC/article/view/571
24. Grassi, L., Rechberger, C., Rønjom, S.: A new structural-differential property of 5-round AES. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 289–317. Springer, Heidelberg (Apr / May 2017). https://doi.org/10.1007/978-3-319-56614-6_10
25. Guo, C., Guo, J., List, E., Song, L.: Towards closing the security gap of tweak-aNd-tweak (TNT). In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 567–597. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_19
26. Jean, J., Nikolic, I., Peyrin, T.: Kiasu v1. Submitted to the CAESAR competition (2014)
27. Jean, J., Nikolic, I., Peyrin, T.: Tweaks and keys for block ciphers: The TWEAKEY framework. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 274–288. Springer, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45608-8_15
28. Jean, J., Nikolic, I., Peyrin, T., Seurin, Y.: The deoxys AEAD family. Journal of Cryptology **34**(3), 31 (Jul 2021). https://doi.org/10.1007/s00145-021-09397-w

29. Mantin, I., Shamir, A.: A practical attack on broadcast RC4. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 152–164. Springer, Heidelberg (Apr 2002). https://doi.org/10.1007/3-540-45473-X_13
30. Murphy, S.: The return of the cryptographic boomerang. IEEE Trans. Inf. Theory **57**(4), 2517–2521 (2011)
31. Qin, L., Dong, X., Wang, X., Jia, K., Liu, Y.: Automated search oriented to key recovery on ciphers with linear key schedule. IACR Trans. Symm. Cryptol. **2021**(2), 249–291 (2021). https://doi.org/10.46586/tosc.v2021.i2.249-291
32. Rahman, M., Saha, D., Paul, G.: Boomeyong: Embedding yoyo within boomerang and its applications to key recovery attacks on AES and Pholkos. IACR Trans. Symm. Cryptol. **2021**(3), 137–169 (2021). https://doi.org/10.46586/tosc.v2021.i3.137-169
33. Rønjom, S., Bardeh, N.G., Helleseth, T.: Yoyo tricks with AES. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 217–243. Springer, Heidelberg (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_8
34. Sasaki, Y.: Improved related-tweakey boomerang attacks on deoxys-BC. In: Joux, A., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 18. LNCS, vol. 10831, pp. 87–106. Springer, Heidelberg (May 2018). https://doi.org/10.1007/978-3-319-89339-6_6
35. Selçuk, A.A.: On probability of success in linear and differential cryptanalysis. Journal of Cryptology **21**(1), 131–147 (Jan 2008). https://doi.org/10.1007/s00145-007-9013-7
36. Song, L., Qin, X., Hu, L.: Boomerang connectivity table revisited. IACR Trans. Symm. Cryptol. **2019**(1), 118–141 (2019). https://doi.org/10.13154/tosc.v2019.i1.118-141
37. Tiessen, T., Knudsen, L.R., Kölbl, S., Lauridsen, M.M.: Security of the AES with a secret S-box. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 175–189. Springer, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-48116-5_9
38. Tolba, M., Abdelkhalek, A., Youssef, A.M.: A meet in the middle attack on reduced round kiasu-bc. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences **99**(10), 1888–1890 (2016)
39. Wagner, D.: The boomerang attack. In: Knudsen, L.R. (ed.) FSE'99. LNCS, vol. 1636, pp. 156–170. Springer, Heidelberg (Mar 1999). https://doi.org/10.1007/3-540-48519-8_12
40. Wang, H., Peyrin, T.: Boomerang switch in multiple rounds. IACR Trans. Symm. Cryptol. **2019**(1), 142–169 (2019). https://doi.org/10.13154/tosc.v2019.i1.142-169
41. Yang, Q., Song, L., Sun, S., Shi, D., Hu, L.: New properties of the double boomerang connectivity table. IACR Transactions on Symmetric Cryptology **2022**(4), 208–242 (Dec 2022). https://doi.org/10.46586/tosc.v2022.i4.208-242, https://tosc.iacr.org/index.php/ToSC/article/view/9977
42. Zhao, B., Dong, X., Jia, K.: New related-tweakey boomerang and rectangle attacks on deoxys-bc including BDT effect. IACR Trans. Symm. Cryptol. **2019**(3), 121–151 (2019). https://doi.org/10.13154/tosc.v2019.i3.121-151
43. Zhao, B., Dong, X., Jia, K., Meier, W.: Improved related-tweakey rectangle attacks on reduced-round Deoxys-BC-384 and Deoxys-I-256-128. In: Hao, F., Ruj, S., Sen Gupta, S. (eds.) INDOCRYPT 2019. LNCS, vol. 11898, pp. 139–159. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-35423-7_7