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.

While the underlying ideas are already known, we show that a careful analysis 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 structural distinguisher with complexity 2^{87} and a key recovery attack with complexity 2^{61}.

The truncated boomerang attacks 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 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

1 Introduction

The AES [DR02] 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 proposals reuse the AES round function: the tweakable block ciphers Kiasu, Deoxys, and TNT-AES use the AES round function with a modified tweak and key schedule.

However, the additional tweak in these constructions allows an attacker to introduce a difference in the state during the computation, so that they must be evaluated in the related-tweak or related-key model. In this model,

the boomerang attack is particularly effective because it can combine two short high-probability differentials. In particular, the best known attacks against Kiasu and Deoxys 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 [BLNS18]: our goal is to provide a unified formula taking into account many details of a broad class of attacks. By integrating a set of techniques proposed in different variants of the attack, we obtain significant improvements of several attacks proposed in the literature.

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 appending extra rounds for the key recovery, we consider the truncated boomerang attack as a whole, including the key recovery thanks to the first and last round transitions. The framework integrates and improves on previous analyses, including structures of plaintexts and ciphertexts [BDK02], and truncated differentials as introduced by Wagner [Wag99].

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 attack with complexity 2^{61}, improving the previous boomerang attack with complexity 2^{71} [Bir04].

We adapt those results to 8-round Kiasu-BC (section 5), by revisiting a previous boomerang attack with complexity 2^{103} [DL17]. Using structures of ciphertexts, we obtain the best attack against Kiasu-BC, with complexity 2^{83}.

Boomerang properties can also threaten constructions using reduced-round AES as a building block (section 6). Indeed, TNT-AES [BGGS20] uses 6-round AES as an internal block cipher, and we build a marginal distinguisher with complexity slightly below 2^{128}. The attack is not competitive with generic attacks against TNT, but as far we know this is the first property of 6-round AES that can be used to target the full TNT-AES (or any construction using 6-round AES).

Finally, we apply the framework to Deoxys-BC, using a MILP model to find good parameters for the attack automatically (section 7). The MILP 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. We obtain improved attacks against most variants of Deoxys-BC.

The trails used in attacks on reduced AES (and KIASU-BC and TNT-AES), and on Deoxys-BC are quite different, but the underlying analysis is the same. In both cases, the improvement over previous works comes principally from the use of structures of ciphertexts, which is made easy by following our framework.

Distinguishers and key-recovery. 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 "inde-

pendent of the secret key" in [GRR17]). Indeed, a series of recent works have proposed complex distinguishers on 5-round [GRR17] and 6-round [RBH17,BR19,BGL20] AES, and we obtain similar results with more simple 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 key-recovery attacks.

2 Preliminaries

2.1 The AES Round Function

AES (previously Rijndael) was designed in 1998 by Daemen and Rijmen and won the NIST standardization competition in 2000 [DR02]. Three variants 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 10 rounds, 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 [JNP14b] and Deoxys [JNPS21] 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 [BGGS20] 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 summing to 256 or 384; 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:

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

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	$2^{36.6}$	[GRR17]
	6	Yoyo	$2^{122.8}$	ACC	$2^{121.8}$	[RBH17]
	6	Exchange attack	$2^{88.2}$	CP	$2^{88.2}$	[BR19]
	6	Exchange attack	2^{84}	ACC	2^{83}	[Bar19]
	6	Truncated differential	$2^{89.4}$	CP	$2^{96.5}$	[BGL20]
	6	Truncated boomerang	2^{87}	ACC	2^{87}	subsection 4.1
AES Key-recovery	6	Square	2^{32}	CP	2^{71}	[DKR97]
	6	Partial-sum	2^{32}	CP	2^{48}	[FKL+01]
	6	Boomerang	2^{71}	ACC	2^{71}	[Bir04]
	6	Mixture	2^{26}	CP	2^{80}	[BDK+20]
	6	Retracing boomerang	2^{55}	ACC	2^{80}	[DKRS20]
	6	Boomeyong	$2^{79.7}$	ACC	2^{78}	[RSP21]
	6	Truncated boomerang	2^{59}	ACC	2^{61}	Appendix B
AES Secret S-Box KR	5	Square	2^{40}	CP	2^{40}	[TKKL15]
	5	Multiple-of-n	$2^{53.3}$	CP	$2^{52.6}$	[Gra18]
	5	Retracing boomerang	$2^{25.8}$	ACC	$2^{25.8}$	[DKRS20]
	6	Square	2^{64}	CP	2^{90}	[TKKL15]
	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^{82}	2^{82}	2^{81}	Figure 7
	9						B	2^{129}	2^{168}	2^{129}	Figure 8
RTK2	8	B	2^{28}	2^{28}	2^{27}	[Sas18][a]	B	2^{27}	2^{27}	2^{27}	Figure 9
	9	B	2^{112}	2^{98}	2^{17}	[Sas18]	B	2^{55}	2^{55}	2^{55}	Figure 10
	10	B	$2^{98.4}$	$2^{109.1}$	2^{88}	[ZDJ19]	B	2^{90}	2^{90}	2^{89}	Figure 11
	11	R	$2^{122.1}$	$2^{249.9}$	$2^{128.2}$	[ZDJ19]	B	2^{129}	2^{218}	2^{129}	App. D
RTK3	10	B	2^{22}	2^{22}	2^{17}	[Sas18]	B	$2^{19.4}$	$2^{19.4}$	2^{18}	Figure 12
	11	B	2^{100}	2^{100}	2^{17}	[Sas18]	B	$2^{32.7}$	$2^{32.7}$	$2^{32.7}$	Figure 13
	12	B	2^{98}	2^{98}	2^{64}	[ZDJ19]	B	$2^{67.4}$	$2^{67.4}$	2^{65}	Figure 14
	13	R	$2^{125.2}$	$2^{186.7}$	2^{136}	[ZDJM19]	B	$2^{126.4}$	$2^{169.7}$	$2^{126.4}$	Figure 2
	14	R	$2^{125.2}$	$2^{282.7}$	2^{136}	[ZDJM19]	B	2^{129}	$2^{278.8}$	2^{129}	Figure 15

[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	$2^{48.5}$	[DEM16]
	8	Meet-in-the-Middle (KR)	2^{116}	CP	2^{116}	[TAY16]
	8	Imposs. Diff (KR)	2^{118}	CP	2^{118}	[DL17]
	8	Boomerang (KR)	2^{103}	ACK	$2^{103.1}$	[DL17]
	8	Truncated boomerang (KR)	2^{83}	ACK	2^{83}	section 5
TNT-AES	$*$-5-$*$	Boomerang (dist.)	2^{126}	ACK	2^{126}	[BGGS20]
	5-$*$-$*$	Impossible differential (KR)	$2^{113.6}$	CP	$2^{113.6}$	[GGLS20]
	$*$-$*$-$*$	Generic (dist.)	$2^{99.5}$	CP	$2^{99.5}$	[GGLS20]
	$*$-6-$*$	Truncated boomerang (dist.)	$2^{127.8}$	ACK	$2^{127.8}$	section 6

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 [JNP14b]) ensures that a byte of subtweakey 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 following the AES order:

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

A diagonal is one of the four sets of bytes with positions $(0, 5, 10, 15)$, $(4, 9, 14, 3)$, $(8, 13, 2, 7)$, $(12, 1, 6, 11)$ and the main diagonal is the first of the four. An anti-diagonal is one of the four sets of bytes $(0, 7, 10, 13)$, $(1, 4, 11, 14)$, $(2, 5, 8, 15)$, $(3, 6, 9, 12)$ and the main anti-diagonal is the first of the four.

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 $\Delta_{\text{in}} \in \{0,1\}^n$ and an output difference $\Delta_{\text{out}} \in \{0,1\}^n$. We use the notation $\Delta_{\text{in}} \xrightarrow[E]{p} \Delta_{\text{out}}$ when a differential exists with probability p, where p is defined as

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

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 $\mathcal{D}_{\text{in}} \xrightarrow[E]{\vec{p}} \mathcal{D}_{\text{out}}$ to denote the existence of a truncated differential with probability $\vec{p}$, defined as:

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

We also define the probability of the reverse truncated differential as

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

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

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

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. On this truncated differential, $\mathcal{D}_{\text{in}}$ corresponds to the vector space of elements which have zeros on all diagonals except the main one. $\mathcal{D}_{\text{out}}$ is the vector space of states which inverses through the MixColumns operation have 0 values on every anti-diagonal except the main one. Therefore, $|\mathcal{D}_{\text{out}}| = |\mathcal{D}_{\text{in}}| = 2^{32}$. 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 [Wag99], 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 $\Delta_{\text{in}} \xrightarrow[E_0]{p} \Delta_{\text{out}}$ and $\nabla_{\text{in}} \xrightarrow[E_1]{q} \nabla_{\text{out}}$. The attack proceeds as follows:

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

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

$$E_0(\overline{P}_i) + E_0(\overline{P}_i') = E_0(P_i) + E_0(P_i') + E_1^{-1}(C_i) + E_1^{-1}(\overline{C_i}) + E_1^{-1}(C_i') + E_1^{-1}(\overline{C_i'})$$

Moreover, the differentials in E_0 and E_1 imply that:

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

When the three events are satisfied, we obtain $E_0(\overline{P_i}) + E_0(\overline{P'_i}) = \Delta_{\text{out}}$ and with an additional probability p, $\overline{P}_i + \overline{P}'_i = \Delta_{\text{in}}$. Finally, assuming that all events are independent, the relation $\overline{P_i} + \overline{P'_i} = \Delta_{\text{in}}$ is verified with probability

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

Figure 3 (page 33) 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

Several variants and improvements of the boomerang attack have emerged since Wagner's original work.

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 [Mur11]. Several techniques have been proposed to improve this analysis.

Multiple Differentials. Since the differences Δ_{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 [Wag99,BDK01], this increases the probability to

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

The Sandwich Attack. Instead of splitting the cipher E into two parts $E = E_1 \circ E_0$, Dunkelman, Keller and Shamir [DKS10] 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 using the connection probability r of E_m:

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

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 [CHP+18] to analyze the case where E_m is an S-Box layer.

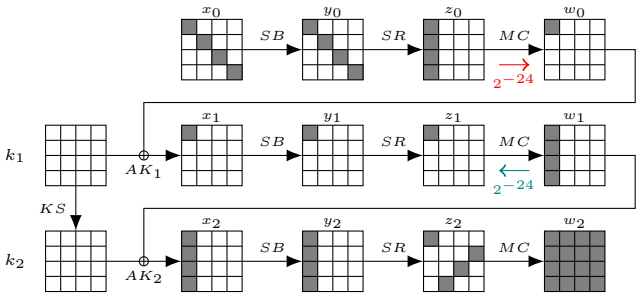


Fig. 1. Example of a truncated differential trail on 3-round AES.

Plaintext-only Attacks. The amplified boomerang attack [KKS01] and the rectangle attack [BDK01] are variants of the boomerang attack using only encryption queries (without adaptively chosen decryption queries). The complexity increases from $(pq)^{-2}$ to $2^{n/2}(pq)^{-1}$ (with the same condition that $pq \gg 2^{-n/2}$). In this paper, we focus on standard boomerang attacks.

Structures. Biham, Dunkelman and Shamir have introduced a variant of the boomerang attack using structures for the key recovery [BDK02]. They start from a boomerang distinguisher with fixed differences Δ_{in} and ∇_{out}, then they add extra rounds at the beginning and at the end. By propagating the differences Δ_{in} and ∇_{out}, they obtain a set of possible input differences $\mathcal{D}_{\text{in}}$ and output differences $\mathcal{D}_{\text{out}}$. In a typical SPN cipher, these sets are vector spaces.

The attacker builds a structure $P + \mathcal{D}_{\text{in}} = \{P + \delta : \delta \in \mathcal{D}_{\text{in}}\}$, and uses it as starting point for the attack. A structure of $|\mathcal{D}_{\text{in}}|$ elements defines $|\mathcal{D}_{\text{in}}|^2/2$ pairs, and $|\mathcal{D}_{\text{in}}|/2$ of them lead to the fixed difference Δ_{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 $\mathcal{D}_{\text{out}}$. However, many later works do not use structures on the ciphertext side.

3 Truncated Boomerang Attacks

We consider boomerang attacks with truncated differentials, as introduced by Wagner in the original paper [Wag99]. We obtain a key-recovery attack, improving on the use of structures of Biham et al. [BDK02] by considering truncated differentials for the full cipher, instead of starting from a shorter boomerang distinguisher with fixed input/output differences and adding extra key-recovery rounds. Some boomerang attacks of this type have been proposed on AES [Bir04] and Kiasu-BC [DL17], but they only use structures on the plaintext side. Our framework combines both ideas, with truncated differentials for E_0 and E_1, and structures on both sides.

3.1 Truncated Boomerang Distinguisher

Let us consider two truncated differentials $\mathcal{D}^0_{\text{in}} \xrightarrow[E_0]{\vec{p}} \mathcal{D}^0_{\text{out}}$ and $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{\vec{q}} \mathcal{D}^1_{\text{out}}$ with probabilities $\vec{p}, \breve{p}$ and $\vec{q}, \breve{q}$ 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 $(P, P', \overline{P}, \overline{P'})$ and the corresponding ciphertexts $(C, C', \overline{C}, \overline{C'})$, with $P + P' \in \mathcal{D}^0_{\text{in}}$ and $C + \overline{C}, C' + \overline{C'} \in \mathcal{D}^1_{\text{out}}$. We have:

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

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

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

If the four events hold, we have $\overline{P} + \overline{P'} \in \mathcal{D}^0_{\text{in}}$ with an additional probability $\breve{p}$. This analysis of the truncated boomerang distinguisher is the same as proposed by Wagner [Wag99], 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 $|\mathcal{D}^1_{\text{in}}|^{-1}$, which would imply $E_0(\overline{P}) + E_0(\overline{P'}) \in \mathcal{D}^0_{\text{out}}$; hence $r \geq |\mathcal{D}^1_{\text{in}}|^{-1}$. Moreover, if $\mathcal{D}^1_{\text{in}}$ is a vector subspace, then $\Sigma = E_0(P) + E_0(P') + E_0(\overline{P}) + E_0(\overline{P'}) \in \mathcal{D}^1_{\text{in}}$; in particular, $\Sigma \in \mathcal{D}^0_{\text{out}}$ with probability $r = |\mathcal{D}^0_{\text{out}} \cap \mathcal{D}^1_{\text{in}}|/|\mathcal{D}^1_{\text{in}}|$, which would imply $E_0(\overline{P}) + E_0(\overline{P'}) \in \mathcal{D}^0_{\text{out}}$.

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 with probability p_b, and randomly satisfies $\overline{P_i}^j + \overline{P_{i'}}^{j'} \in \mathcal{D}^0_{\text{in}}$ with probability $p_\$$ defined as:

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

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

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:

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

Following the analysis of [MS02, Theorem 2], we can detect the bias with $Q = \mathcal{O}(p_\$^{-1}\sigma^{-2}) = \mathcal{O}(p_b^{-1}\sigma^{-1})$ samples. 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 $\Phi(\sqrt{c}/2)$, with Φ 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 on 6-round AES in figure 4 (page 34), using the trail of figure 1 twice. $\mathcal{D}^0_{\text{in}}$ and $\mathcal{D}^1_{\text{in}}$ are the sets of all 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:

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

The full distinguisher is detailed in Algorithm 2 (page 34). 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 [GRR17]. As seen in Table 1, the complexity is slightly higher than previous distinguishers with similar properties, but 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. Indeed, in Section 6 we show an attack against TNT-AES based on this distinguisher, while previous 6-round distinguishers do not seem applicable.

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 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 $\mathcal{D}^0_{\text{in}} \xrightarrow[E_0]{\vec{p}} \mathcal{D}^0_{\text{out}}$, and two pairs of known ciphertexts $(C, \overline{C})$ and $(C', \overline{C'})$ following the truncated differential $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{\vec{q}} \mathcal{D}^1_{\text{out}}$. 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 counters 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 (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 [Sel08, 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 full attack is described in Algorithm 1 (page 33). 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 $|\mathcal{D}^0_{\text{in}}|^2 |\mathcal{D}^1_{\text{out}}| \times T_C \ll 2^n$, 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 [Sel08]:

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

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 [Sel08, section 3.2.1], the normal approximation is inaccurate in this case; instead, we approximate them by Poisson distributions to compute the success probability.

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 $\mathcal{D}^0_{\text{mid}}$:

$$E_0 = \tilde{E_0} \circ \mathsf{SB} \circ \mathsf{AK}_{K_0}$$

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 will 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 $\mathsf{ShiftRows}(\mathsf{MixColumns}(\delta))$ 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:

$$\mathcal{D}^0_{\text{in}} \xrightarrow[\mathsf{SB}]{\vec{p}_0} \mathcal{D}^0_{\text{mid}} \qquad \vec{p}_0 = |\mathcal{D}^0_{\text{mid}}|/|\mathcal{D}^0_{\text{in}}| \qquad \mathcal{D}^0_{\text{mid}} \xrightarrow[\tilde{E_0}]{\vec{p}_1} \mathcal{D}^0_{\text{out}} \qquad \vec{p} = \vec{p}_0 \times \vec{p}_1 \tag{14}$$

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 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 $\ell^0_2 = |\mathcal{D}^0_{\text{mid}}|^2/|\mathcal{D}^0_{\text{in}}|$ 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 $|\mathcal{D}^0_{\text{mid}}|^2 < |\mathcal{D}^0_{\text{in}}|$, 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 4):

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

For the key extraction, we have

$$\ell = |\mathcal{D}^0_{\text{mid}}|^2 \cdot |\mathcal{D}^1_{\text{mid}}|^2 \cdot |\mathcal{D}^0_{\text{in}}|^{-1} \cdot |\mathcal{D}^1_{\text{out}}|^{-1} = 2^{-32} \quad \kappa = \log_2(|\mathcal{D}^0_{\text{in}}| \cdot |\mathcal{D}^1_{\text{out}}|) = 64$$

Our analysis results in

$$p_b = \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-128}$$
$$p_w = |\mathcal{D}^0_{\text{in}}| \times 2^{-n} \times \ell \times 2^{-\kappa} = 2^{-192} \qquad \tilde{\sigma} = 2^{64}$$

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

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

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.

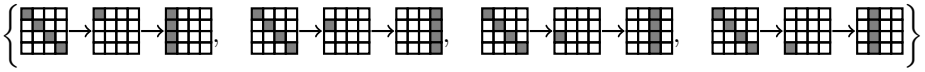
We have implemented the attack on a reduced AES with 4-bit S-Boxes, and it behaves as expected.

4 Optimized Boomerang Attacks on 6-round AES

As shown by Biryukov [Bir04], 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

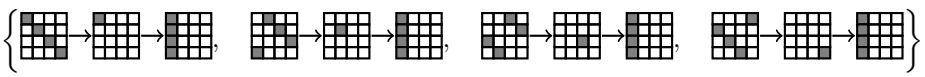
Instead of only considering the trail of Figure 1 with fixed positions for all the active bytes, we consider a collection of four different trails for upper part:



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

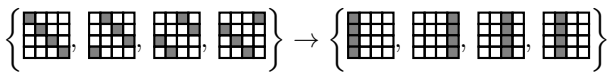


Again, this can be considered as a truncated differential $\mathcal{D}^1_{\text{in}} \xrightarrow[E_1]{\vec{q}} \mathcal{D}^1_{\text{out}}$ with

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

The analysis of the previous sections can be applied as-is with these trails. We obtain a better attack because we have increased $\vec{p}$ and $\overleftarrow{q}$ by a factor 4, even though the increase of $|\mathcal{D}^1_{\text{in}}|$ reduces the probability of the boomerang 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 $\mathcal{D}^1_{\text{out}}$. 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 $\mathcal{D}^1_{\text{in}} \xrightarrow[E_0]{\vec{p}} \mathcal{D}^1_{\text{out}}$ 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 $\overline{\mathcal{D}}^0_{\text{in}}$ for Returning Pairs. Following Biryukov [Bir04], 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 $\overline{\mathcal{D}}^0_{\text{in}} \xrightarrow[E_0]{\vec{p}} \mathcal{D}^0_{\text{out}}$ with

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

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

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

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 use the trails above, but we keep the set $\mathcal{D}^1_{\text{out}}$ active only in the first column.

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

When extracting the key, we recover information about a diagonal of k_0 from (P, P'), and information about an anti-diagonal of k_6 from $(C, \overline{C})$ and $(C', \overline{C}')$:

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

Therefore, we have the following parameters:

$$p_b = \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}^2 \times |\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-122}$$
$$p_w = |\overline{\mathcal{D}}^0_{\text{in}}| \times 2^{-n} \times \ell \times 2^{-\kappa} = 2^{-162} \qquad \tilde{\sigma} = 2^{40}$$

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

$$Q = \mu \times p_b^{-1} = 2^{124} \qquad D = 2^{63}$$

Time complexity. The complexity is dominated by the oracle queries: for each structure of 2^{64} plaintexts/ciphertexts, we filter $2^{64} \times 2^{63} \times |\overline{\mathcal{D}}^0_{\text{in}}| \times 2^{-n} = 2^{33}$ candidate quartets, and the time to extract the key candidates is negligible.

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 $\overleftarrow{q}$ (this has negligible complexity).

In Appendix B, we further reduce the complexity to $D = 2^{58.7}, T = 2^{60.8}$.

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 [GRR16], 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 4. 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 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:

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

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 [Sel08]).

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 is an instance of the TWEAKEY framework [JNP14a,JNP14b], 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 know attack on Kiasu-BC is an 8-round attack with complexity 2^{103} in data and time [DL17] following this idea; the corresponding boomerang is represented in Figure 5 (page 35). 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 $(\mathcal{D}_{\text{in}}, \mathcal{D}_{\text{tw}}) \xrightarrow[E]{\vec{p}} \mathcal{D}_{\text{out}}$, where $\mathcal{D}_{\text{in}}$ is the set of differences active on the main diagonal, $\mathcal{D}_{\text{out}}$ is the set of differences δ such that $\mathsf{MixColumns}^{-1}(\delta)$ is active on the main anti-diagonal, and $\mathcal{D}_{\text{tw}}$ is the set of differences active in the first cell of the tweak. Following the tweakey schedule of Kiasu-BC, a tweak difference in $\mathcal{D}_{\text{tw}}$ results in a tweakey difference in $\mathcal{D}_{\text{tw}}$ at each round.

The upper truncated trail is followed with probability 2^{-32}: with probability 2^{-24} the difference at the end of the first round is active only on the first cell, and with probability 2^{-8} it cancels with the tweakey difference. The same applies for the bottom truncated trail.

As in the AES case, the final round omits the MixColumns operation, therefore, we use a slightly different trail in the lower part, where the set $\mathcal{D}^1_{\text{out}}$ is active

on the main anti-diagonal. We obtain an 8-round boomerang with two 4-round differentials (Figure 5):

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

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

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

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 keep only 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 another tweakable block cipher reusing the AES round function published at Eurocrypt 2020 [BGGS20]. 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 [GGLS20] proved the bound to be at least $\mathcal{O}(2^{3n/4})$ 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

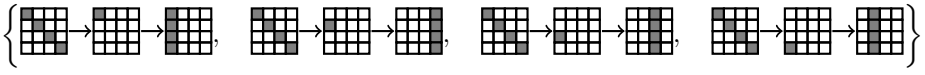
$$\begin{array}{ll} X + X' = T + T' = \Delta_{\text{in}} & \overline{X} + \overline{X}' = \overline{T} + \overline{T}' = \Delta'_{\text{in}} \\ Y + \overline{Y} = T + \overline{T} = \nabla_{\text{out}} & Y' + \overline{Y}' = T' + \overline{T}' = \nabla'_{\text{out}} \end{array}$$

When using a truncated boomerang (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 $\mathcal{D}^0_{\text{in}}$. This decreases the boomerang probability.
- We necessarily have $\Delta_{\text{in}} + \Delta'_{\text{in}} = \nabla_{\text{out}} + \nabla'_{\text{out}}$. For the 6-round AES truncated boomerang of figure 4, this implies $\Delta_{\text{in}} = \Delta'_{\text{in}}$ 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 [BGGS20].

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 $\overline{T} + \overline{T'} = \Delta^0_{\text{in}}$:

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

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 subsection C.1). Therefore, 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 Probability. We obtain:

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

As shown in section C, 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, 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,

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

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 succes 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 super-structures 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%.

7 Application to Deoxys-BC using a MILP Model

Deoxys-BC is another tweakable block cipher using the AES round function and the STK construction for the tweakey schedule [JNPS21], on which the best known attacks are based on boomerangs [CHP+17,Sas18,ZDJ19,ZDJM19].

In the single tweakey model, we assume that the adversary has access to Deoxys-BC with a fixed key and tweak. The analysis is similar to analysis of 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.

Due to the larger tweak used in Deoxys-BC, the boomerangs given in the previous sections do not give the best attacks; instead the best known boomerangs have been found using a MILP modeling. In this section we revisit these attacks using our framework for truncated boomerang.

7.1 Using a MILP Model to Search for Boomerang Characteristics

Mixed Integer Linear Programming (MILP) is a mathematical optimization tool that minimizes a linear objective function of constrained variables. The variables can be either discrete or continuous, and the constraints are given as linear inequalities. In the last years, it has proven to be a useful tool to evaluate the security of cryptographic primitives, due to the facility of encoding cryptographic properties as MILP problems, and the availability of high performance solvers. Differential trails can be modeled with a MILP program in order to obtain bounds on the probability of trails (as first used on SIMD [BFL11]), or to search for good trails (as first used by Mouha *et al.* [MWGP11]).

This method was applied to the search of boomerang distinguisher on Deoxys by Cid *et al.* [CHP+17]. 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 minimize Sbox transition probabilities, using the DDT and BCT of the AES S-Box. 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 remains in the final trail, otherwise it is unlikely to find concrete differences for the tweakey.

In 2019, Zhao et al. [ZDJ19,ZDJM19] improved this MILP model by adding two extra rounds at the end of the lower trail, containing truncated differences.

7.2 MILP Model for Truncated Boomerang Attacks

We extend the MILP model of [CHP+17] to find good truncated boomerang attacks. Previous works [Sas18,ZDJM19] have shown large differences between the complexity of an attack and the probability of a boomerang distinguisher; in particular, the best attack is not always obtained with the best distinguisher. We follow the same high-level approach as in [QDW+21]: our main objective is to cover the full boomerang attack with the MILP model. To do so, we ask the MILP solver to minimize the formula for the data complexity of the attack given in subsection 3.2. In addition, we improve the boomerang switch probability estimation compared to [CHP+17] by modeling the ladder switch in the MILP model.

On a high level, the model is given by a set of variables, a set of constraints, and an objective function. Our model is not symmetric when switching plaintext and ciphertext[1], but for simplicity we only describe the attack starting from the plaintext, though it works similarly the other way around.

State Variables. The model of Cid *et al.* considers only two types of internal differences: either it is inactive with a zero difference, or it is active with a fixed non-zero difference. In order to model truncated trails, we consider four types of differences for all intermediate state variables:

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

For the tweakey schedule variables, we use only the first two types for simplicity.

[1] The inverse AES round can be re-written with the same form as the encryption, but this requires to use equivalent round keys, and interacts badly with the introduction of sparse differences through the tweakey schedule.

Table 4. Transition probability (DDT) and connection probability (BCT) for the AES S-Box. For the transition probability of equal differences, we distinguish the first and the second pair. We omit the cases where the probability is 0.

Legend
□ $\delta = 0$
■ $\delta \neq 0$ fixed
■ unknown δ
■ equal δ

Transition probability	1st pair	2nd pair
$\Pr(□ \to □) = 1$	$\Pr(■ \to ■) \stackrel{1}{=} 1$	$\Pr(■ \to ■) \stackrel{2}{=} 2^{-7}$
$\Pr(■ \to ■) = 2^{-6}$	$\Pr(■ \to ■) \stackrel{1}{=} 1$	$\Pr(■ \to ■) \stackrel{2}{=} 2^{-8}$
$\Pr(■ \to ■) = 1$	$\Pr(■ \to ■) \stackrel{1}{=} 2^{-8}$	$\Pr(■ \to ■) \stackrel{2}{=} 2^{-7}$
$\Pr(■ \to ■) = 2^{-8}$	$\Pr(■ \to ■) \stackrel{1}{=} 1$	$\Pr(■ \to ■) \stackrel{2}{=} 1$
$\Pr(■ \to ■) = 1$	$\Pr(■ \to ■) \stackrel{1}{=} 1$	$\Pr(■ \to ■) \stackrel{2}{=} 2^{-7}$

Boomerang connexion probability			
$\Pr(□ \to □) = 1$	$\Pr(■ \to □) = 1$	$\Pr(■ \to □) = 1$	$\Pr(■ \to □) = 1$
$\Pr(□ \to ■) = 1$	$\Pr(■ \to ■) = 2^{-6}$	$\Pr(■ \to ■) = 1$	$\Pr(■ \to ■) = 2^{-8}$
$\Pr(□ \to ■) = 2^{-8}$	$\Pr(■ \to ■) = 2^{-8}$	$\Pr(■ \to ■) = 1$	$\Pr(■ \to ■) = 2^{-8}$
$\Pr(□ \to ■) = 1$	$\Pr(■ \to ■) = 2^{-8}$	$\Pr(■ \to ■) = 1$	$\Pr(■ \to ■) = 2^{-8}$

Equal Differences. 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 4; the model does not encode linear relations between active bytes (*e.g.* the set of differences for w_2 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.

In terms of analysis, we treat them specially: for the first pair they are considered as truncated bytes, but for the second pair they are considered as fixed differences (fixed to the value given by the first pair). Therefore, we explain the rest of the MILP model assuming that each trail has been duplicated, and equal differences have been replaced.

Constraints. We have constraints for each operation:

SubBytes. With truncated differences, the variables before and after the S-Box layer do not necessarily have the same type. However, an S-Box output is active (truncated or not) if and only if the input is active.

ShiftRows. ShiftRows only moves the bytes, so we have trivial equalities between the corresponding state variables.

MixColumns. The MixColumns operation operates on each column, multiplying it with an MDS matrix. Because of the MDS property, each column is either completely inactive, or has at least 5 active bytes (truncated or not) on input and output. Moreover, we have the same property with truncated bytes: either no byte is truncated, or at least 5 bytes are truncated.
We also reuse the constraints for the degree of freedom given in [CHP+17] to avoid over-defining the fixed differences (for these constraints, the equal bytes are not considered as fixed).

AddTweaKey. The AddTweaKey operation is just a XOR with the subkey. In our model, the subkey is not truncated, so it is either inactive, or active with a fixed difference. Therefore, the input state is truncated if and only if the output state is truncated. Otherwise, we use the constraints of Cid *et al.* to model XOR for the active bytes and the activity of the tweakey bytes.

Key Schedule. We follow the approach of [CHP+17] to model the key schedule. The permutation h just moves the bytes, and the LFSR construction ensures that each byte is either completely inactive, or inactive in at most $i-1$ rounds in the TK_i model.

Additional Constraint. Without additional constraint, a silly truncated trail with probability 1 is returned. To avoid that, we constrain the trail to have at maximum 3 active truncated columns in a single state, except for the first, middle, and last rounds.

Objective Function. Using the results from the previous sections, we estimate the data complexity of an attack as:

$$\begin{aligned} &D = \max(\sqrt{2Q}, 2Q \times |\mathcal{D}^0_{\text{in}}|^{-1} \times |\mathcal{D}^1_{\text{out}}|^{-1}) && \text{with} \\ &Q \approx \max(p_b^{-1}, \tilde{\sigma}^{-1} \times p_b^{-1}) && \tilde{\sigma} \approx p_b/(p_\$ \times \ell \times 2^{-\kappa}) \\ &p_b = \vec{p} \cdot \overleftarrow{p} \cdot \overleftarrow{q}_1 \cdot \overleftarrow{q}_2 \times r && p_\$ = |\mathcal{D}^0_{\text{in}}|/2^n \end{aligned}$$

Since all variables are represented logarithmically by the MILP model, these formulas only involve additions and maximums, and are easily expressed in function of the MILP variables:

- $|\mathcal{D}^0_{\text{in}}|$ and $|\mathcal{D}^1_{\text{out}}|$ are obtained by counting truncated input/output bytes;
- $\vec{p}$, $\overleftarrow{p}$, $\overleftarrow{q}_1$ and $\overleftarrow{q}_2$ are computed from S-Box and MixColumns transitions. Due to the equal bytes, the probability of the lower trail is computed twice: $\overleftarrow{q}_1$ after replacing them with truncated bytes, and $\overleftarrow{q}_2$ after replacing them with fixed bytes. Similarly $\vec{p}$ is computed after replacing them with truncated bytes, and $\overleftarrow{p}$ after replacing them with fixed bytes (see Table 4);
- r is computed from the connection probability of each S-Box, in Table 4;
- $\ell \times 2^{-\kappa}$ is estimated as $\vec{p_0}^2 \cdot \overleftarrow{q_0}^2$ following section 3.2, where the probability $\vec{p_0}$ for the first round and $\overleftarrow{q}_0$ for the last round are evaluated from the trail.

Trail Probability. With truncated differences, both MixColumns and SubBytes operations contribute to the probability of a trail. If at least one byte of a column is truncated, the probability of a MixColumns transition is 2^{-8t} where t is the number of non truncated bytes (active or not) in the output of MixColumns.

The SubBytes probability is computed by multiplying the transition probability of all the S-Boxes, following the transition probabilities given in Table 4.

Boomerang Connection Probability. In section 3, we evaluated the connection probability as $r = |\mathcal{D}^1_{\text{in}}|^{-1}$. This worked well for AES boomerangs, but when targeting Deoxys we usually have fixed differences in the middle of the cipher and we

obtain better results using the Boomerang Connectivity Table (BCT) [CHP+18]. Instead of splitting the cipher in two parts $E = E_1 \circ E_0$, we split it in three parts $E = E_1 \circ E_m \circ E_0$ with E_m an S-Box layer. Given fixed differences on both sides of E_m, the probability that the boomerang connects is given by multiplying the BCT probabilities of each S-Box. In the more general context of our MILP program, the probabilities are given in Table 4.

This analysis can be improved using the ladder switch [BK09]. Instead of splitting the cipher with E_m a full S-Box layer, we use the Super-Box representation of the two middle rounds: from one S-Box layer to the next, the AES round operates as four independent parallel transformations. Each of those four transformations can independently be considered as part of the upper or lower trail. We obtain E_0 and E_1 with partial rounds in the middle, and E_m corresponds to S-Boxes of different rounds. We model the ladder switch using a binary variable for each Super-Box in the middle, encoding whether it is part of the upper trail or the lower trail. Depending on these variables, the S-Boxes and MDS matrices in the middle are counted as part of E_0, E_m, or E_1.

7.3 Results

We run the MILP solver with different parameters and retain the trails with the lowest complexity among all the models with the same total length and attacker model. For the search, we use gurobi with 96 threads on 2x AMD EPIC 7352, 24 cores, 2.3GHz and 256 GB of RAM. The solve time varies from several minutes to several days, but the best trail is in general found much faster than the optimality proof. Memory shortage (exceeding 256 GB) was an issue for some of the largest models. After the MILP solving phase, we instantiate the trail with concrete tweakey differences, and apply slight manual improvements. For instance, for minor gains, we introduced state changing bytes: fixed on the forward trail but truncated on the return trail.

General Remarks. The best attacks found are listed in Table 2. We explain the parameters of a few attacks below and the rest are given in Appendix D. The values of ℓ and κ mentionned 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 quartets and $\mu = 1$ suffices, else several right quartets are needed to get the correct key ranked first.

In the related-tweakey model, generating a structure of S elements requires $4S$ queries (under 4 tweakeys) and produces S^2 quartets; but with structures on both sides the encryption queries are amortized and only $2S$ queries are needed.

For 13-round Deoxys-BC in the RTK3 model and 10-round Deoxys-BC in the RTK2 model, we selected a sub-optimal trail with a better time complexity.

10-round Deoxys-BC in the RTK2 model (Figure 11). Query two full structures of 2^{88} ciphertexts, so that on average, $\mu = 2 \cdot 2^{88+88} \cdot p_b = 2$ quartets follow the trail. For each element of the structure, deduce on average 1 candidate for 28 bits of key on the plaintext side: 1 candidate for $tk_0[5, 10]$ and 1 representant of the 4 possible candidates each for $tk_0[0]$ and $tk_0[15]$[2]. In total, there are on average $2^{1+88+88-56-28} = 2^{93}$ 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. For each of the $2^{93-8} = 2^{85}$ 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 $tk_9[4, 9]$. There remains $2^{85-32-16} = 2^{37}$ quartets with a 116-bit key candidate. The only candidate suggested twice is expected to be the right candidate. The time complexity is dominated by the generation of the quartets, thus $(D, T, M) = (2^{90}, 2^{90}, 2^{89})$.

13-round Deoxys-BC in the RTK3 model (Figure 2). Query a partial structure of $2^{125.4}$ plaintexts. On average, $\mu = 2^{125.4} \cdot 2^{125.4} \cdot p_b = 4$ 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 $\delta =$ `0xc4657e42` and $\delta_{\text{in}} =$ `0x00007a00`, and look for collisions between:

$$\begin{aligned} v &= y_0[2, 7, 8, 13] \quad \| \, \overline{y_0}[2, 7, 8, 13] \quad \| \, \overline{P}[0, 5, 10, 15] \quad \| \, k \\ v' &= y'_0[2, 7, 8, 13] + \delta \, \| \, \overline{y'_0}[2, 7, 8, 13] + \delta \, \| \, \overline{P'}[0, 5, 10, 15] + \delta_{\text{in}} \, \| \, k' \end{aligned}$$

   This step costs $2^{32} \cdot 2 \cdot 2^{125.4} = 2^{158.4}$ in time complexity. On average, $2^{125.4} \cdot 2^{125.4} \cdot 2^{-114} = 2^{136.8}$ quartets remain for each $tk_0[2, 7, 8, 13]$ ($2^{168.8}$ in total).
3. For each quartet, retrieve $2^{7+7-32} = 2^{-18}$ values of $tk_0[3, 4, 9, 14]$. 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 $2^{8.6+164.8} = 2^{177.4}$ accesses; and $2^{32+136.8-18} = 2^{150.8}$ 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 $tk_{12}^{eq}[2, 7, 8]$. $2^{150.8-18+16-24} = 2^{124.8}$ 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 $tk_1[4]$

[2] S-Boxes 0 and 15 on the plaintext side each have 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 $\langle \delta, x + x' \rangle$.

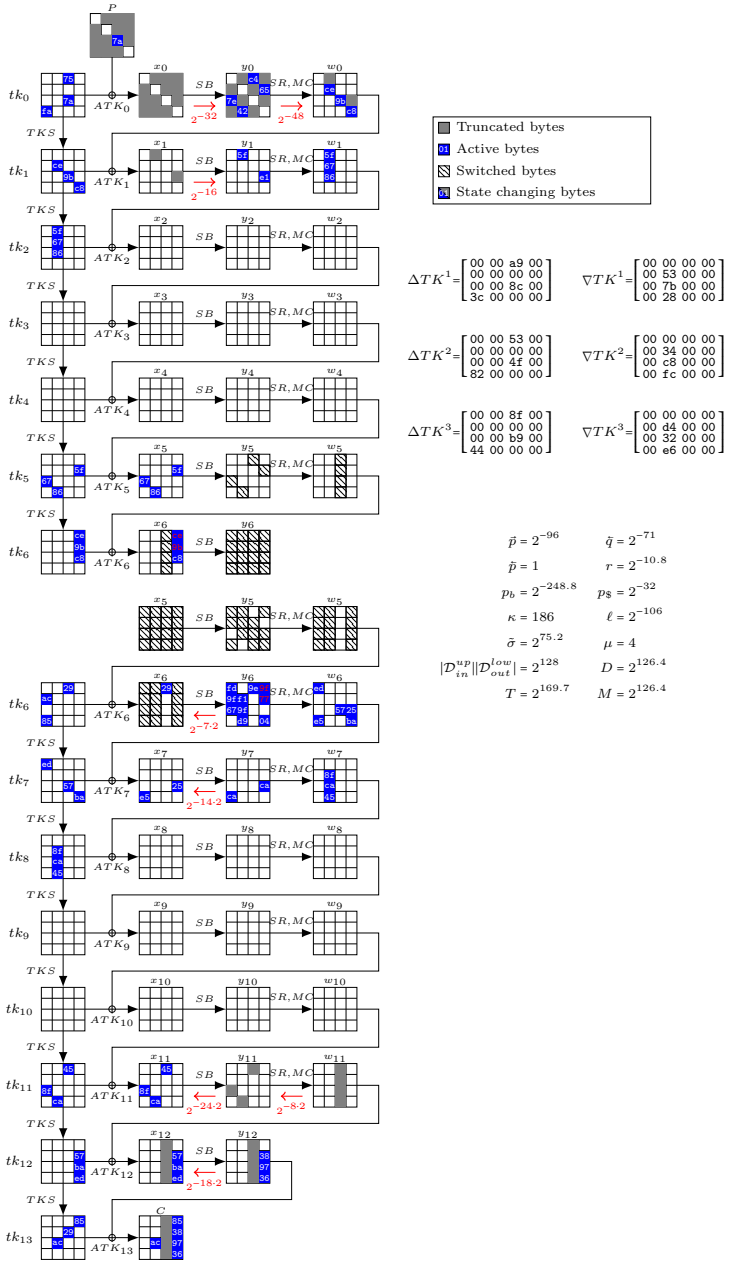


Fig. 2. Truncated boomerang attacks on 13-round Deoxys-BC in the RTK3 model, starting from the plaintext side. This attack succeeds with probability 0.76.

and 2^{-6} values for $tk_1[14]$ (2 candidates are deduced per pair because the differences are already compatible). $2^{124.8} \cdot 2^{-12} = 2^{112.8}$ quartets remain.

6. Eventually, each of the $2^{112.8}$ 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 $\lambda = 2^{-73.2}$. The probability that any wrong counter is at least 3 is $(1 - e^{-\lambda}(1 + \lambda + \lambda^2/2)) \cdot 2^{184} \approx 2^{-35}$. 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 $2^{177.4}$ table accesses of step 3. An encryption of 13-round Deoxys-BC has 16×13 S-Boxes, so the time complexity is equivalent to $2^{177.4}/208 = 2^{169.7}$ encryption. Thus $(D, T, M) = (2^{126.4}, 2^{169.7}, 2^{126.4})$.

Acknowledgement.

We would like to thank the authors of [CHP+17] for providing the code they used to generate the MILP programs.

This work was partially financially supported by the french ministry of defence – Agence de l'Innovation de Défense (AID). This work was also supported by the French Agence Nationale de la Recherche (ANR), under grant ANR-20-CE48-0017 (project SELECT).

References

Bar19. Navid Ghaedi Bardeh. 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.

BDK01. Eli Biham, Orr Dunkelman, and Nathan Keller. The rectangle attack - rectangling the Serpent. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 340–357. Springer, Heidelberg, May 2001.

BDK02. Eli Biham, Orr Dunkelman, and Nathan Keller. New results on boomerang and rectangle attacks. In Joan Daemen and Vincent Rijmen, editors, *FSE 2002*, volume 2365 of *LNCS*, pages 1–16. Springer, Heidelberg, February 2002.

BDK+20. Achiya Bar-On, Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. Improved key recovery attacks on reduced-round AES with practical data and memory complexities. *Journal of Cryptology*, 33(3):1003–1043, July 2020.

BFL11. Charles Bouillaguet, Pierre-Alain Fouque, and Gaëtan Leurent. Security analysis of SIMD. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010*, volume 6544 of *LNCS*, pages 351–368. Springer, Heidelberg, August 2011.

BGGS20. Zhenzhen Bao, Chun Guo, Jian Guo, and Ling Song. TNT: How to tweak a block cipher. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 641–673. Springer, Heidelberg, May 2020.

BGL20. Zhenzhen Bao, Jian Guo, and Eik List. Extended truncated-differential distinguishers on round-reduced AES. *IACR Trans. Symm. Cryptol.*, 2020(3):197–261, 2020.

Bir04. Alex Biryukov. The boomerang attack on 5 and 6-round reduced aes. In *International Conference on Advanced Encryption Standard*, pages 11–15. Springer, 2004.

BK09. Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, Heidelberg, December 2009.

BLNS18. Christina Boura, Virginie Lallemand, María Naya-Plasencia, and Valentin Suder. Making the impossible possible. *Journal of Cryptology*, 31(1):101–133, January 2018.

BR19. Navid Ghaedi Bardeh and Sondre Rønjom. The exchange attack: How to distinguish six rounds of AES with $2^{88.2}$ chosen plaintexts. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 347–370. Springer, Heidelberg, December 2019.

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

CHP+18. Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 683–714. Springer, Heidelberg, April / May 2018.

DEM16. Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Square attack on 7-round kiasu-BC. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *ACNS 16*, volume 9696 of *LNCS*, pages 500–517. Springer, Heidelberg, June 2016.

DKR97. Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher Square. In Eli Biham, editor, *FSE'97*, volume 1267 of *LNCS*, pages 149–165. Springer, Heidelberg, January 1997.

DKRS20. Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. The retracing boomerang attack. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 280–309. Springer, Heidelberg, May 2020.

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

DL17. Christoph Dobraunig and Eik List. Impossible-differential and boomerang cryptanalysis of round-reduced kiasu-BC. In Helena Handschuh, editor, *CT-RSA 2017*, volume 10159 of *LNCS*, pages 207–222. Springer, Heidelberg, February 2017.

DR02. Joan Daemen and Vincent Rijmen. *The design of Rijndael*, volume 2. Springer, 2002.

FKL+01. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Michael Stay, David Wagner, and Doug Whiting. Improved cryptanalysis of Rijndael. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 213–230. Springer, Heidelberg, April 2001.

GGLS20. Chun Guo, Jian Guo, Eik List, and Ling Song. Towards closing the security gap of tweak-aNd-tweak (TNT). In Shiho Moriai and Huaxiong Wang,

editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 567–597. Springer, Heidelberg, December 2020.

Gra18. Lorenzo Grassi. MixColumns properties and attacks on (round-reduced) AES with a single secret S-box. In Nigel P. Smart, editor, *CT-RSA 2018*, volume 10808 of *LNCS*, pages 243–263. Springer, Heidelberg, April 2018.

GRR16. Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace trail cryptanalysis and its applications to AES. *IACR Trans. Symm. Cryptol.*, 2016(2):192–225, 2016. https://tosc.iacr.org/index.php/ToSC/article/view/571.

GRR17. Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. A new structural-differential property of 5-round AES. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 289–317. Springer, Heidelberg, April / May 2017.

JNP14a. Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Kiasu v1. *Submitted to the CAESAR competition*, 2014.

JNP14b. Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 274–288. Springer, Heidelberg, December 2014.

JNPS21. Jérémy Jean, Ivica Nikolic, Thomas Peyrin, and Yannick Seurin. The deoxys AEAD family. *Journal of Cryptology*, 34(3):31, July 2021.

KKS01. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified boomerang attacks against reduced-round MARS and Serpent. In Bruce Schneier, editor, *FSE 2000*, volume 1978 of *LNCS*, pages 75–93. Springer, Heidelberg, April 2001.

MS02. Itsik Mantin and Adi Shamir. A practical attack on broadcast RC4. In Mitsuru Matsui, editor, *FSE 2001*, volume 2355 of *LNCS*, pages 152–164. Springer, Heidelberg, April 2002.

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

MWGP11. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *Inscrypt*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

QDW+21. Lingyue Qin, Xiaoyang Dong, Xiaoyun Wang, Keting Jia, and Yunwen Liu. Automated search oriented to key recovery on ciphers with linear key schedule. *IACR Trans. Symm. Cryptol.*, 2021(2):249–291, 2021.

RBH17. Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseth. Yoyo tricks with AES. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 217–243. Springer, Heidelberg, December 2017.

RSP21. Mostafizar Rahman, Dhiman Saha, and Goutam Paul. 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.

Sas18. Yu Sasaki. Improved related-tweakey boomerang attacks on deoxys-BC. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 18*, volume 10831 of *LNCS*, pages 87–106. Springer, Heidelberg, May 2018.

Sel08. Ali Aydin Selçuk. On probability of success in linear and differential cryptanalysis. *Journal of Cryptology*, 21(1):131–147, January 2008.

TAY16. Mohamed Tolba, Ahmed Abdelkhalek, and Amr M Youssef. 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.

TKKL15. Tyge Tiessen, Lars R. Knudsen, Stefan Kölbl, and Martin M. Lauridsen. Security of the AES with a secret S-box. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 175–189. Springer, Heidelberg, March 2015.

Wag99. David Wagner. The boomerang attack. In Lars R. Knudsen, editor, *FSE'99*, volume 1636 of *LNCS*, pages 156–170. Springer, Heidelberg, March 1999.

ZDJ19. Boxin Zhao, Xiaoyang Dong, and Keting Jia. New related-tweakey boomerang and rectangle attacks on deoxys-bc including BDT effect. *IACR Trans. Symm. Cryptol.*, 2019(3):121–151, 2019.

ZDJM19. Boxin Zhao, Xiaoyang Dong, Keting Jia, and Willi Meier. Improved related-tweakey rectangle attacks on reduced-round Deoxys-BC-384 and Deoxys-I-256-128. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *INDOCRYPT 2019*, volume 11898 of *LNCS*, pages 139–159. Springer, Heidelberg, December 2019.

A Additional Figures and Tables

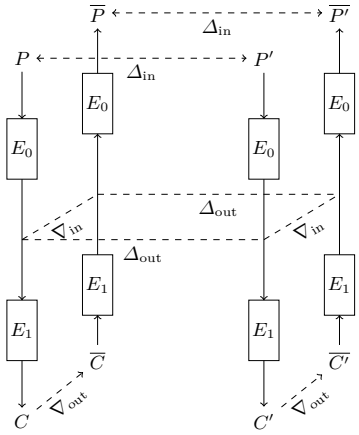


Fig. 3. Construction of a boomerang quartet.

Algorithm 1. Truncated boomerang key recover attack

Require: $\mathcal{D}^0_{\text{in}}$, $\mathcal{D}^1_{\text{out}}$
  $\mathcal{K} \leftarrow$ InitKeyCounters()
  for $i \leftarrow 1$ to N **do**
    $P_0 \leftarrow$ Rand()
    $\mathcal{P} \leftarrow [P_0 + \Delta_{\text{in}}, \text{ for } \Delta_{\text{in}} \in \mathcal{D}^0_{\text{in}}]$
    $\mathcal{C} \leftarrow [E(P), \text{ for } P \in \mathcal{P}]$ ▷ $N \times |\mathcal{D}^0_{\text{in}}|$ encryptions
    $\overline{\mathcal{P}} \leftarrow [E^{-1}(C + \Delta_{\text{out}}), \text{ for } C \in \mathcal{C},\ \Delta_{\text{out}} \in \mathcal{D}^1_{\text{out}}]$ ▷ $N \times |\mathcal{D}^0_{\text{in}}||\mathcal{D}^1_{\text{out}}|$ decryptions
    $\mathcal{H} \leftarrow$ InitHashMap()
    for $\overline{P} \in \overline{\mathcal{P}}$ **do**
      Insert the projection of $\overline{P}$ on $\{0,1\}^n/\mathcal{D}^0_{\text{in}}$ in $\mathcal{H}$ ▷ $|\mathcal{D}^0_{\text{in}}||\mathcal{D}^1_{\text{out}}|$ in memory
    if a collision occurs in $\mathcal{H}$ between $\overline{P}$ and $\overline{P}'$ **then**
      Track back to the corresponding P and P'
      if $P \neq P'$ **then** ▷ $N(p_{rand} + p_{trunc})|\mathcal{D}^0_{\text{in}}|^2|\mathcal{D}^1_{\text{out}}|^2/2$ such quartets
        for K in the ℓ key candidates induced by the quartet **do**
          Increment the counter K in $\mathcal{K}$
  Recover the key material K of the maximal counter of $\mathcal{K}$
  Recover the rest of the key by performing the attack with a different subspace $\mathcal{D}^0_{\text{in}}$

Algorithm 2. Truncated boomerang distinguisher on 6-round AES

$q \leftarrow 0$
for $0 \leq s < 2^{35}$ **do** ▷ 2^{35} structures
  $P_0 \leftarrow \$$
  for $i \in \mathcal{D}^0_{\text{in}}$ **do** ▷ Iterate over the main diagonal
    $C_i \leftarrow E(P_0 + i)$
    for $j \in \mathcal{D}^1_{\text{out}}$ **do** ▷ Iterate over the main anti-diagonal
      $\overline{P_i}^j \leftarrow E^{-1}(C_i + j)$
      Store $\overline{P_i}^j$ in a hash table indexed by three diagonals
  Count collisions in the hash table, and increment q
if $q > 2^{66} + 2^{33}$ **then**
  return AES
else
  return \$

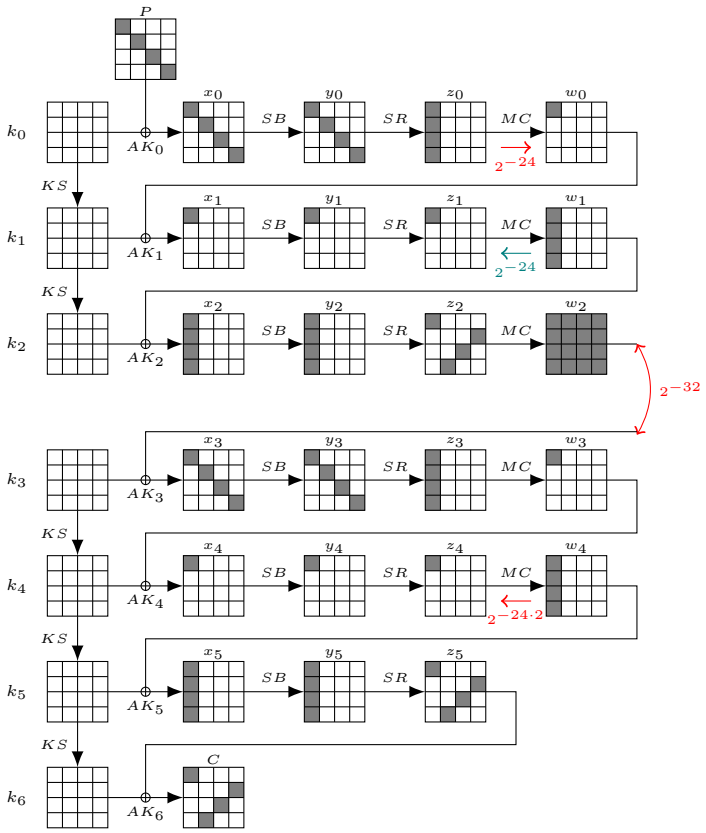


Fig. 4. A truncated boomerang trail on 6-round AES.

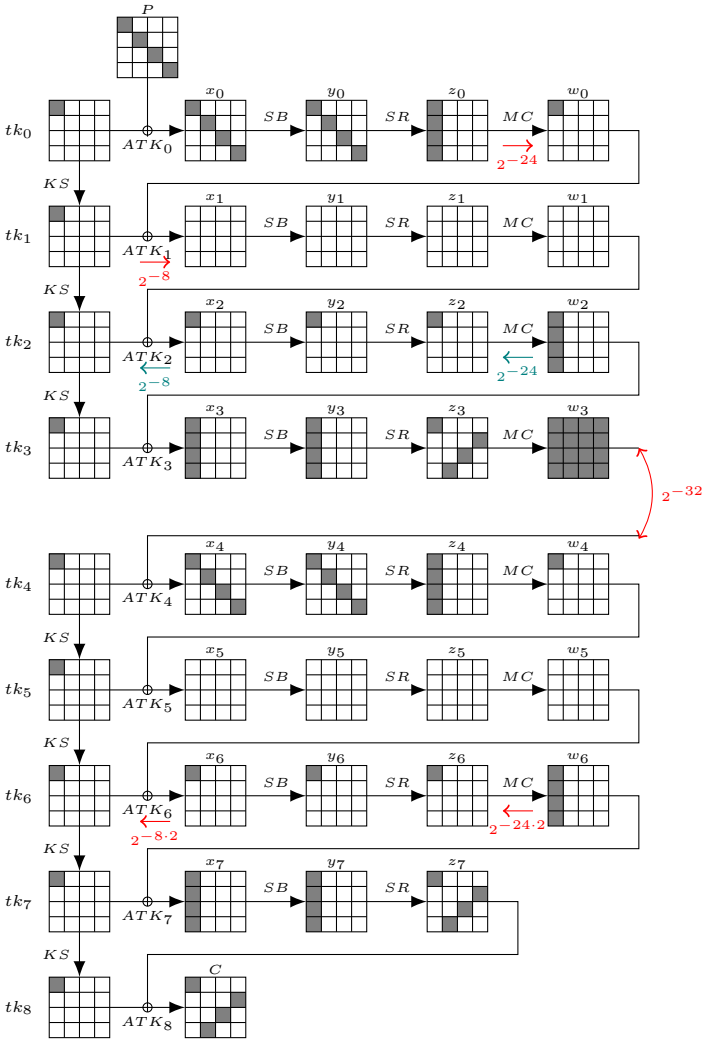


Fig. 5. Truncated boomerang trail on 8-round Kiasu-BC.

B Optimized Key-recovery Attack

In the key-recovery attack, the key extraction provides an additional filter, so we can use truncated boomerang characteristics with lower signal-to-noise ratios. Following [Bir04], 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:

$$\overline{\mathcal{D}}^0_{\text{in}} = \left\{ \right\}$$

$\vec{p} = 2^{-22}$	$\overleftarrow{p} = 2^{-24}$	$\lvert\mathcal{D}^0_{\text{in}}\rvert = 2^{32}$	$\lvert\mathcal{D}^0_{\text{out}}\rvert = 2^{34}$	$\lvert\mathcal{D}^0_{\text{mid}}\rvert = 2^{10}$
$\vec{\bar{p}} = 2^{-46}$	$\overleftarrow{\bar{p}} = 6 \times 2^{-16} = 2^{-13.4}$	$\lvert\overline{\mathcal{D}}^0_{\text{in}}\rvert = 6 \times 2^{64}$	$\lvert\mathcal{D}^0_{\text{out}}\rvert = 2^{34}$	
$\vec{q} = 2^{-24}$	$\overleftarrow{q} = 2^{-22}$	$\lvert\mathcal{D}^1_{\text{in}}\rvert = 2^{34}$	$\lvert\mathcal{D}^1_{\text{out}}\rvert = 2^{32}$	$\lvert\mathcal{D}^1_{\text{mid}}\rvert = 2^{10}$

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

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

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 $\overleftarrow{q}$. 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

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

C Optimizing the TNT-AES Boomerang

We can slightly reduce the complexity of the TNT-AES distinguisher by carefully analyzing the probability of the boomerang, and correlations between the sides.

In the lower trail, we have two pairs $(C, \overline{C})$ and $(C', \overline{C}')$ and our analysis assumes that if they both follow the trail, then the differences $E_1^{-1}(C) + E_1^{-1}(\overline{C})$ and $E_1^{-1}(C') + E_1^{-1}(\overline{C}')$ are equal with probability $|\mathcal{D}^1_{\text{in}}|^{-1} = 2^{-32}$. Actually, the differences are not uniformly distributed in $\mathcal{D}^1_{\text{in}}$, and this increases the probability that the differences are equal.

Indeed, we can split this analysis into two disjoint cases: either the differences in x_4 are equal, or they are different. If they are equal, then the 4 active S-Boxes in the fourth round have the same difference in y_3 for both pairs; therefore there are only 127 possible differences in x_3.

$$\begin{aligned}
&\Pr\left[E_1^{-1}(C) + E_1^{-1}(\overline{C}) = E_1^{-1}(C') + E_1^{-1}(\overline{C}')\right] \\
&= \bar{q}^2 \times \left(\frac{254}{255} \times 2^{-32} + \frac{1}{255} \times 127^{-4}\right) \\
&\approx \bar{q}^2 \times 2^{-31.915}
\end{aligned}$$

In the upper trail, there are similar effects. Our analysis assumes that the pair $(\overline{P}, \overline{P}')$ follows the truncated trail with probability $\bar{\bar{p}}$ independently of the pair (P, P'). However, both pairs have the same differences at the input and output of the trail and the trail does not cover the sets $\mathcal{D}^0_{\text{in}}$ and $\mathcal{D}^0_{\text{out}}$ uniformly. Let us consider a pair $\overline{C}, \overline{C}'$ with $E_1^{-1}(\overline{C}) + E_1^{-1}(\overline{C}') = E_1^{-1}(C) + E_1^{-1}(C')$. The difference in y_2 (after the S-Boxes of the third round) are the same for both pairs; the truncated trail allows a set 255 differences in x_2 (before the S-Boxes). In general the probability of a transition through four active S-Boxes is 2^{-32}, but we know that one of the 255 differences was followed by the pair P, P', therefore all differences are compatible, and the probability increases to 127^{-4}.Finally the probability of having a difference in x_2 compatible with the trail is higher than 2^{-24}:

$$1 \times 127^{-4} + 254 \times 2^{-32} \approx 2^{-23.92}$$

In the first round, we have the same analysis as in the lower trail, and the probability to obtain the same difference as the pair P, P' is higher than 2^{-32}

$$\frac{254}{255} \times 2^{-32} + \frac{1}{255} \times 127^{-4} \approx \times 2^{-31.915}$$

Finally, we obtain

$$q_b = 2^{-22} \times 2^{-23.92} \cdot 2^{-31.915} \times 2^{-20.85 \times 2} \times 2^{-31.915} = 2^{-151.45}$$

Multiple trails. The previous analysis uses a single trail for the lower part of the cipher, with the active byte of z_4 in the cell 0. We can also consider alternative trails in cell 1, 2, or 3. With this active byte in cell 1, we have the transition $\Delta^1_{\text{out}} \to (0, *, 0, 0)$ through the layer of S-Boxes with probability $\bar{q} = 256/2^{32} = 2^{-24}$ (instead of $2^{-20.85}$ for cell 0), and the probability is the same in other positions.

Finally, we sum the probabilities of the four alternate boomerangs, and we obtain:

$$q_b = 2^{-151.45} + 2^{-157.75} + 2^{-157.75} + 2^{-157.75} \approx 2^{-151.4}$$

C.1 Choice of Δ^1_{out}

We have to choose a value Δ^1_{out} to maximize the probability of having a single active byte in z_4. Therefore we analyze the S-Box layer of the last round: the output difference is equal to $\mathsf{ShiftRows}^{-1}(\mathsf{MixColumns}^{-1}(\Delta^1_{\text{out}}))$, and we want the input difference to be of one of the following types:

$$\begin{bmatrix} 2\alpha & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ 3\alpha & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} \alpha & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ 3\alpha & 0 & 0 & 0 \\ 2\alpha & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 3\alpha & 0 & 0 & 0 \\ 2\alpha & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 3\alpha & 0 & 0 & 0 \\ 2\alpha & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \end{bmatrix}$$

In particular, $\mathsf{ShiftRows}^{-1}(\mathsf{MixColumns}^{-1}(\Delta^1_{\text{out}}))$ must be active only on the first column. We experimentally tested all possible differences on the first column, and counted the number of pairs satisfying the transition. More precisely, we are interested in the joint probability that two different pairs reach an input difference of the same type, therefore we count the number of quartets of each type (using the DDT of the S-Box for each possible input and output difference).

We found that in the best case, there are $2^{22.35}$ quartets that satisfy this transition (with the same type for both pairs), compared to 2^{18} quartets expected on average (for each type of input difference, we expect on average one pair for each of the 2^8 differences, therefore 2^{8+8} quartets). The are 4×255 choices of Δ^1_{out} reaching this maximum, and we found they they all have a special form: $\mathsf{ShiftRows}^{-1}(\mathsf{MixColumns}^{-1}(\Delta^1_{\text{out}}))$ is of one of the following types, with L the linear transform inside the AES S-Box:

$$\begin{bmatrix} L(\beta/2) & 0 & 0 & 0 \\ L(\beta) & 0 & 0 & 0 \\ L(\beta) & 0 & 0 & 0 \\ L(\beta/3) & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} L(\beta) & 0 & 0 & 0 \\ L(\beta) & 0 & 0 & 0 \\ L(\beta/3) & 0 & 0 & 0 \\ L(\beta/2) & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} L(\beta) & 0 & 0 & 0 \\ L(\beta/3) & 0 & 0 & 0 \\ L(\beta/2) & 0 & 0 & 0 \\ L(\beta) & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} L(\beta/3) & 0 & 0 & 0 \\ L(\beta/2) & 0 & 0 & 0 \\ L(\beta) & 0 & 0 & 0 \\ L(\beta) & 0 & 0 & 0 \end{bmatrix}$$

This special form can be explained by the structure of the AES S-Box; it is defined as $x \mapsto L(x^{254})$, with L an invertible linear mapping. It implies the following property:

Property 1. For any $\alpha, \beta \neq 0$, $\mathsf{DDT}[\alpha, L(\beta)] = \mathsf{DDT}[\alpha\beta, L(1)]$.

Proof.

$$\begin{aligned}
\mathsf{DDT}[\alpha, L(\beta)] &= \left|\left\{x \ : \ L\left((x)^{254}\right) + L\left((x+\alpha)^{254}\right) = L(\beta)\right\}\right| \\
&= \left|\left\{x \ : \ (x)^{254} + (x+\alpha)^{254} = \beta\right\}\right| \\
&= \left|\left\{x \ : \ (\beta x)^{254} + (\beta x + \alpha\beta)^{254} = 1\right\}\right| \\
&= \left|\left\{x \ : \ (x)^{254} + (x+\alpha\beta)^{254} = 1\right\}\right| \\
&= \left|\left\{x \ : \ L\left((x)^{254}\right) + L\left((x+\alpha\beta)^{254}\right) = L(1)\right\}\right| \\
&= \mathsf{DDT}[\alpha\beta, L(1)]
\end{aligned}$$

□

Because of this property, S-Box transitions of the form

$$(\mathsf{2}\alpha, \alpha, \alpha, \mathsf{3}\alpha) \rightarrow (L(\beta/\mathsf{2}), L(\beta), L(\beta), L(\beta/\mathsf{3}))$$

have a higher probability than expected because the transition probability of the S-Boxes are not independent: for a given α and β either all transitions are possible simultaneously, or none of the transitions are possible.

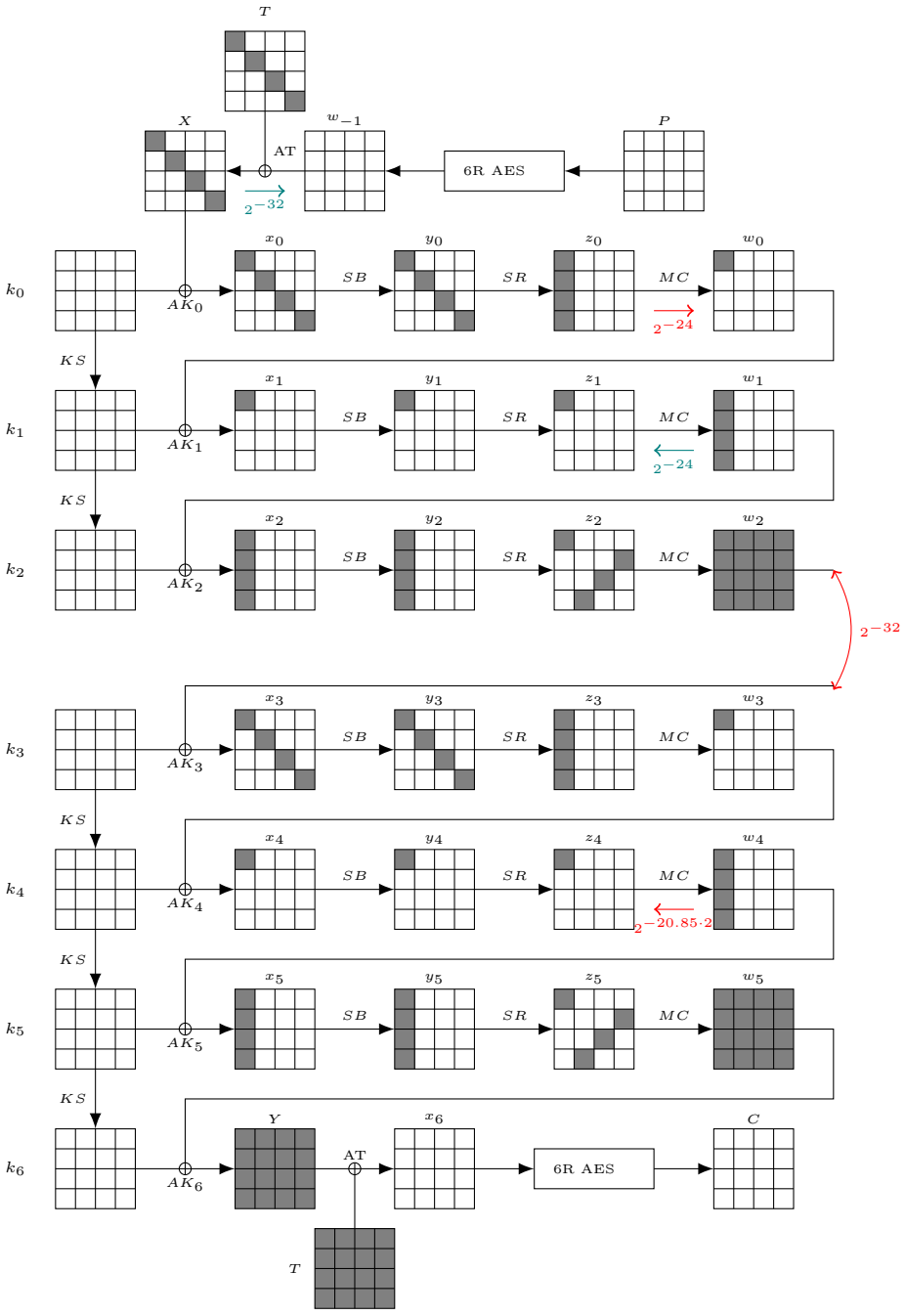


Fig. 6. Scheme of our boomerang attack on the full TNT-AES.

D Truncated Boomerang and Attacks from the MILP Model

8-round Deoxys-BC in the RTK1 model (Figure 7). Query two structures of 2^{80} elements. On average, $\mu = 2 \cdot 2^{80} \cdot 2^{80} \cdot p_b = 2$ quartets follow the trail and $2 \cdot 2^{80} \cdot 2^{80} \cdot p_\$ = 2^{81}$ random quartets are detected. For each element, retrieve in average 1 value of $tk_8[8]$ in a few table accesses. Remove quartets with incompatible candidates. $2^{81} \cdot 2^{-8} = 2^{73}$ quartets remain. For each quartet, retrieve 2^{-40} values of $tk_0[1, 6, 11, 12]$ and $tk_8[10]$. For each of the $2^{73} \times 2^{-40} = 2^{33}$ remaining quartets, deduce in average 1 key candidate for $tk_8[4, 5, 6, 7]$. For each of the 2^{33} remaining quartets, increase the counter of the retrieved 80-bit key candidate. Only the right counter is expected to be greater than 2. The complexity is dominated by the encryption queries. This gives $(D, T, M) = (2^{82}, 2^{82}, 2^{81})$.

9-round Deoxys-BC in the RTK1 model (Figure 8). Query one full structure of 2^{128} plaintexts. On average, $\mu = 2^{128} \cdot 2^{128} \cdot p_b = 4$ quartets follow the trail. For each element of the structure, deduce on average 1 candidate k for 38 bits of key on the ciphertext side: 1 candidate for $tk_9[5, 6, 8, 10]$ and 1 representant of the 4 possible candidates for $tk_9[7]$.

Guess 5 key bytes: $tk_0[0, 5, 10, 15]$ and $tk_1[1]$, so that $w_0[0, 1, 2, 3]$ and $y_1[1]$ can be evaluated from each plaintext. Set $\delta = \texttt{0x9a000000} \| \texttt{0x000000} \| \texttt{0x000000} \| \texttt{0x2d} \| \texttt{0x2d} \| 0$ and look for collisions between:

$$v = \left(\overline{P}[2, 7, 8, 13] \ \| \ w_0[0, 2, 3] \| \ \overline{w}_0[0, 2, 3] \| \ y_1[1] \| \ \overline{y}_1[1] \| \ k\right)$$
$$v' = \left(\overline{P}'[2, 7, 8, 13] \| \ w_0'[0, 2, 3] \| \ \overline{w}_0'[0, 2, 3] \| \ y_1'[1] \| \ \overline{y}_1'[1] \| \ k'\right) \ + \delta$$

Since we match on 134 bits, we expect $2^{128+128-134} = 2^{122}$ remaining quartets for each guess, or 2^{162} in total, with a complexity of $2^{128+40} = 2^{168}$.

Extract more key information from the remaining quartets. First, retrieve 2^{-16} candidates for $tk_0[3, 4, 9, 14]$; this requires about 2^9 table accesses per quartet, which is much less than to 2^6 encryptions; 2^{144} quartet remain. Then retrieve 2^{-16} candidates for $tk_0[1, 6, 11, 12]$, and 2^{-16} candidates for $tk_1[7, 13]$. We end up with 2^{112} candidates for 158 bits of key.

We model the counters for wrong keys as a Poisson distribution with $\lambda = 2^{-44}$; they reach 4 or more with probability $2^{-180.6}$, therefore the right key is expected to be ranked first. This gives $(D, T, M) = (2^{129}, 2^{168}, 2^{129})$.

8-round Deoxys-BC in the RTK2 model (Figure 9). Query a partial structure of 2^{25} ciphertexts. The only detected quartet is a right quartet. $(D, T, M) = (2^{27}, 2^{27}, 2^{27})$.

9-round Deoxys-BC in the RTK2 model (Figure 10). Query a partial structure of 2^{54} ciphertexts. On average, $\mu = 2^{54} \cdot 2^{54} \cdot p_b = 1$ quartets follow

the trail and $2^{54} \cdot 2^{54} \cdot p_\$ = 2^{20}$ random quartets are detected. For each quartet, retrieve 2^{-52} values of $tk_0[0, 5, 10, 15]$ and $tk_9[0, 1, 2, 3, 5]$. This step is of negligible complexity, and with high probability no wrong quartet remains. Thus, $(D, T, M) = (2^{55}, 2^{55}, 2^{55})$

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, this gives 2^{38} candidates for 116 bits. The probability that one of the counters is at least 4 is $2^{-300.6+116+128} = 2^{-56.6}$, so in average, the correct key is ranked first. This gives $(D, T, M) = (2^{129}, 2^{218}, 2^{129})$.

10-round Deoxys-BC in the RTK3 model (Figure 12). Query $2^{1.4}$ structures of 2^{16} ciphertexts. The only detected quartet is a right quartet. $(D, T, M) = (2^{19.4}, 2^{19.4}, 2^{18})$. This attack is equivalent to the attack given in [Sas18], but the complexity was wrongly estimated as 2^{22}.

11-round Deoxys-BC in the RTK3 model (Figure 13). Query a partial structure of $2^{30.7}$ elements. The only detected quartet is a right quartet. $(D, T, M) = (2^{32.7}, 2^{32.7}, 2^{32.7})$.

12-round Deoxys-BC in the RTK3 model (Figure 14). Query $2^{2.4}$ structures of 2^{64} ciphertexts. On average, $\mu = 2^{2.4} \cdot 2^{64} \cdot 2^{64} \cdot p_b = 2$ quartets follow the trail and $2^{2.4} \cdot 2^{64} \cdot 2^{64} \cdot p_\$ = 2^{58.4}$ random quartets are detected. For each quartet, retrieve in average 2^{-32} key candidates for $tk_0[4]$ and $tk_{12}[0, 2, 3]$ in a few table accesses. Then, for each of the $2^{58.4} \cdot 2^{-32} = 2^{26.4}$ remaining quartets, deduce in average $2^{24} \cdot 2^{24} \cdot 2^{-32} = 2^{16}$ candidates for $tk_{12}[12, 13, 14, 15]$. For each candidate, compute the values of $x_{11}[12, 13, 14, 15]$ and the differences in state y_{10}. From the transition $x_{10} \to y_{10}$, retrieve 2^{-24} key candidates for $tk_{11}^{eq}[1, 11, 12]$. Thus, $2^{26.4} \cdot 2^{16} \cdot 2^{-24} = 2^{18.4}$ quartets remain with 1 average key candidate of 88 bits. For each remaining quartet, increase the counter of the corresponding key candidate. Only the right counter is expected to be greater than 2. This gives $(D, T, M) = (2^{67.4}, 2^{67.4}, 2^{65})$.

14-round Deoxys-BC in the RTK3 model (Figure 15). We did not manage to find a 14-round trail with the MILP solver, but the 13-round attack can be extended by adding a round at the end, and guessing most of the last subkey.

We start by querying the decryption oracle over the full codebook with tweaks $\overline{T}$ and $\overline{T}'$, and storing the results in memory. Then we guess 104 bits of the last round key: $k_{14}[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$.

This allows us to essentially simulate the 13-round attack using queries from the 14-round oracle. Since 13 bytes of k_{14} are known, we can partially decrypt the corresponding S-Boxes in the last round, and the MixColumns in the first three columns (after replacing tk_{13} by an equivalent key). The 13-round trail is slightly modified with $p_b = 2^{-252.8}$, in order to limit the ciphertext difference to three columns.

We build the same type of structure as used in the 13-round attack. For each plaintext P_i, we query $C_i = E(P_i, T)$ and we partially decrypt the final round. Then we generate the set of 2^{40} values with the required difference in y_{12}, and partially encrypt them to obtain the corresponding ciphertexts $\overline{C}_i^j$. Finally, we use the stored decryption values to obtain the corresponding $\overline{P}_i^j$.

We start with a full structure of 2^{128} plaintexts, so that we expect $\mu = 2^{128+128} \cdot p_b = 2^{3.2}$ good quartets. As in the 13-round attack (steps 1 and 2), we match elements on 114 bits: we expect $2^{128+128-114} = 2^{142}$ candidate quartets for each guess of $104 + 32 = 136$ bits of key, or 2^{278} quartets in total.

Following the 13-round attack (steps 3,4, and 6), we extract on average 2^{-48} candidates for 80 additional key bits . Finally, we use the constraints of the MixColumns operation of round 11: there are only 2^{16} possible differences in w_{11} from which we deduce $2^{16+16-24} = 2^8$ candidates for $tk_{13}^{eq}[2, 5, 8]$. We end up with 2^{238} suggestions for 258 bits of key.

We keep all key candidates suggested at least 6 times. Modeling the counters for wrong keys as following a Poisson distribution with $\lambda = 2^{-20}$, we expect wrong keys to be kept with probability $2^{-129.5}$. Finally we do an exhaustive search over the 126 key bits remaining, for a cost of $2^{126+128.5} = 2^{254.5}$.

The bottleneck of the attack is the extraction of key candidates for 2^{278} quartets. Following the analysis of the 13-round attack, we estimate that it requires about $2^{8.6}$ table accesses, equivalent to $2^{0.8}$ encryption. The full attack has complexity $(D, T, M) = (2^{129}, 2^{278.8}, 2^{129})$.

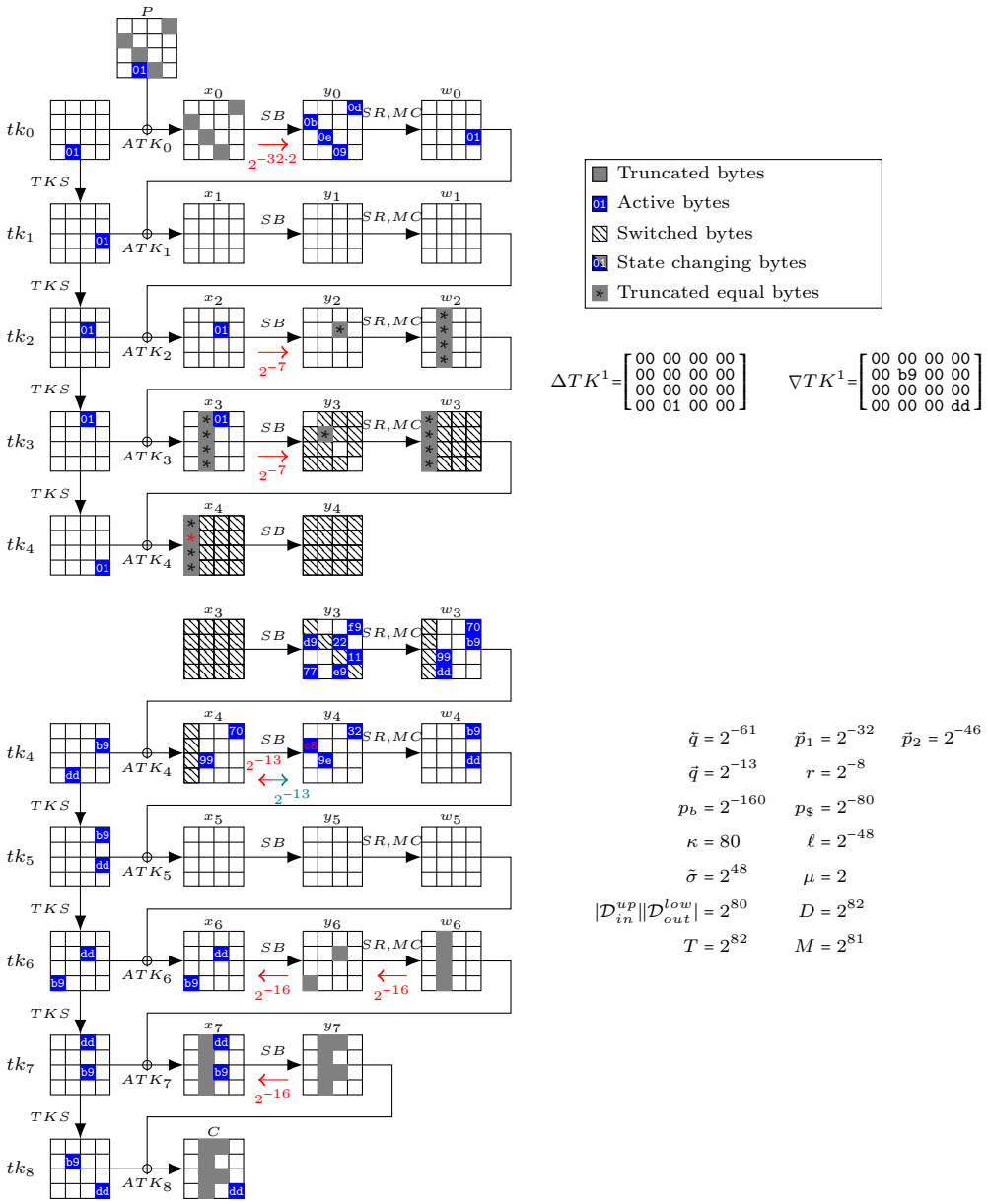


Fig. 7. Truncated boomerang attack on 8-round Deoxys-BC in the RTK1 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

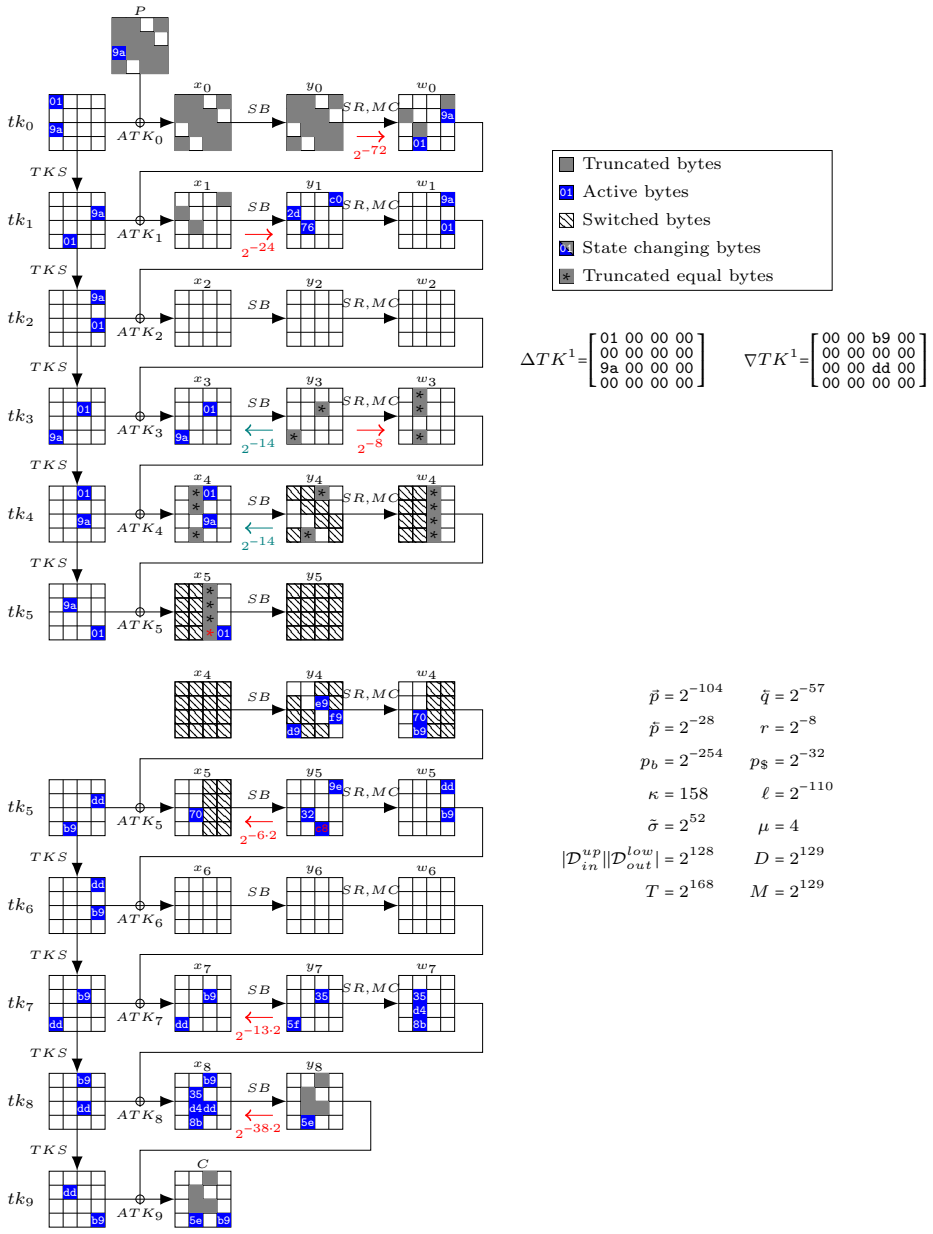


Fig. 8. Truncated boomerang attack on 9-round Deoxys-BC in the RTK1 model, starting from the plaintext side. This attack succeeds with probability 1/2.

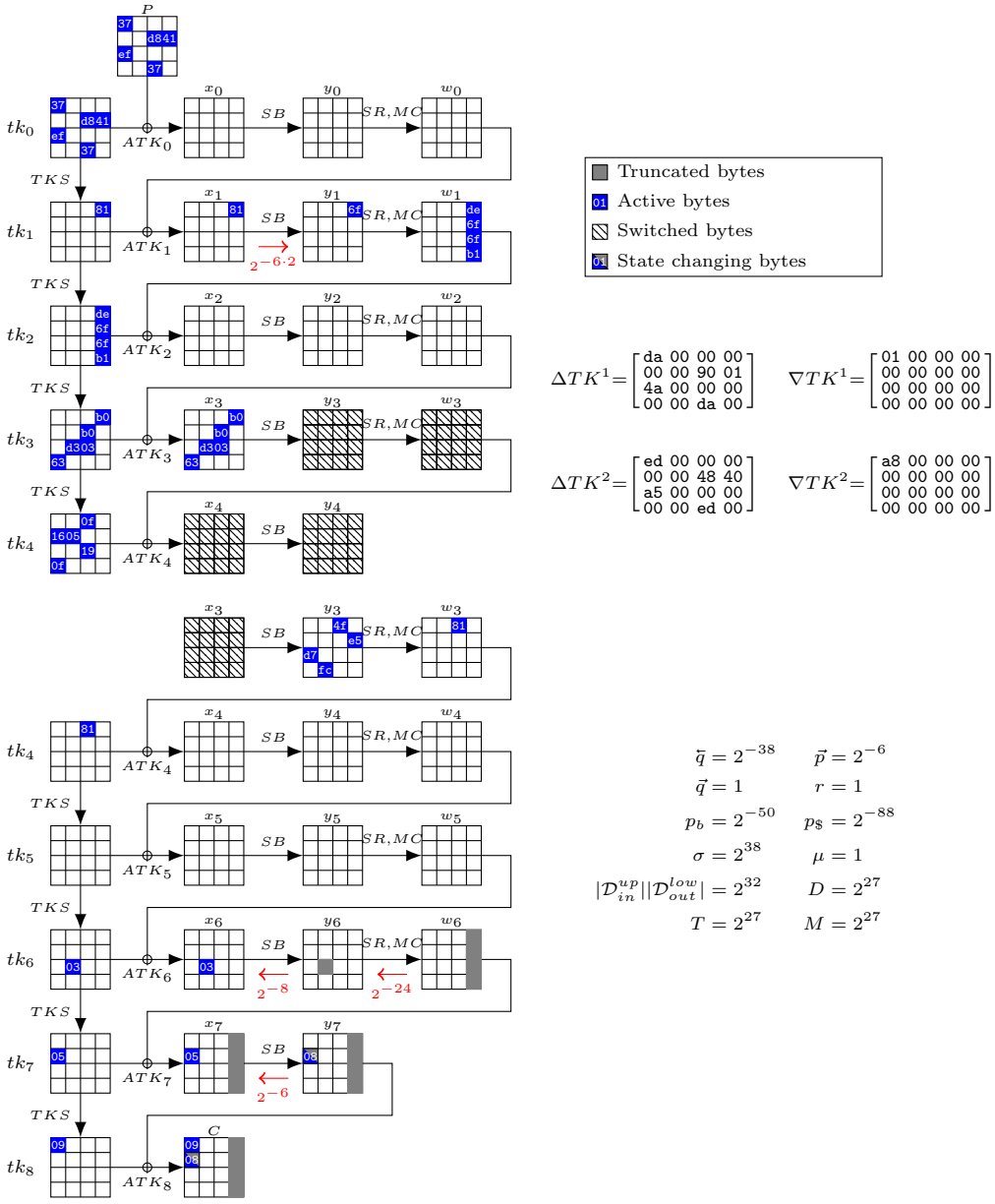


Fig. 9. Truncated boomerang attack on 8-round Deoxys-BC in the RTK2 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

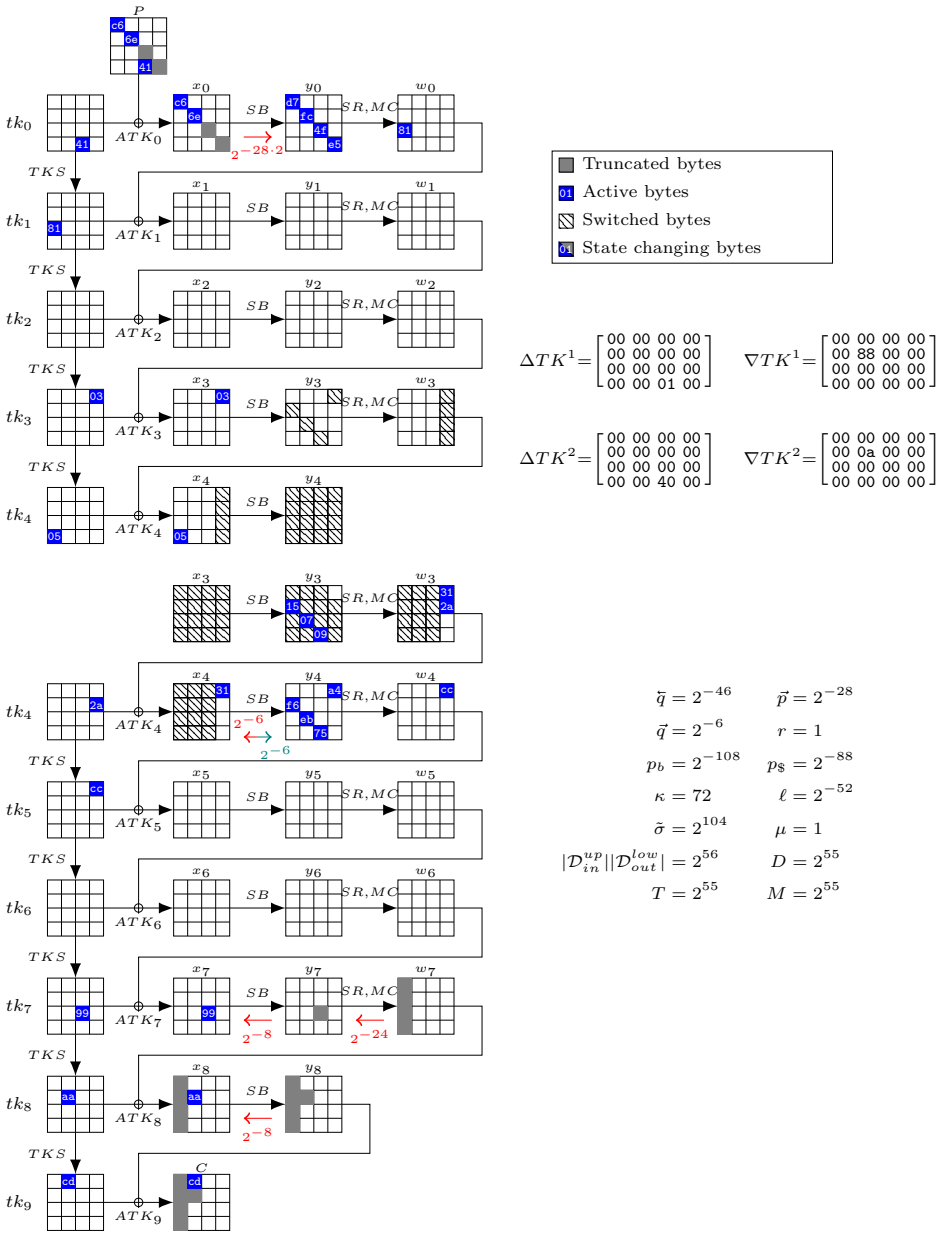


Fig. 10. Truncated boomerang attack on 9-round Deoxys-BC in the RTK2 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

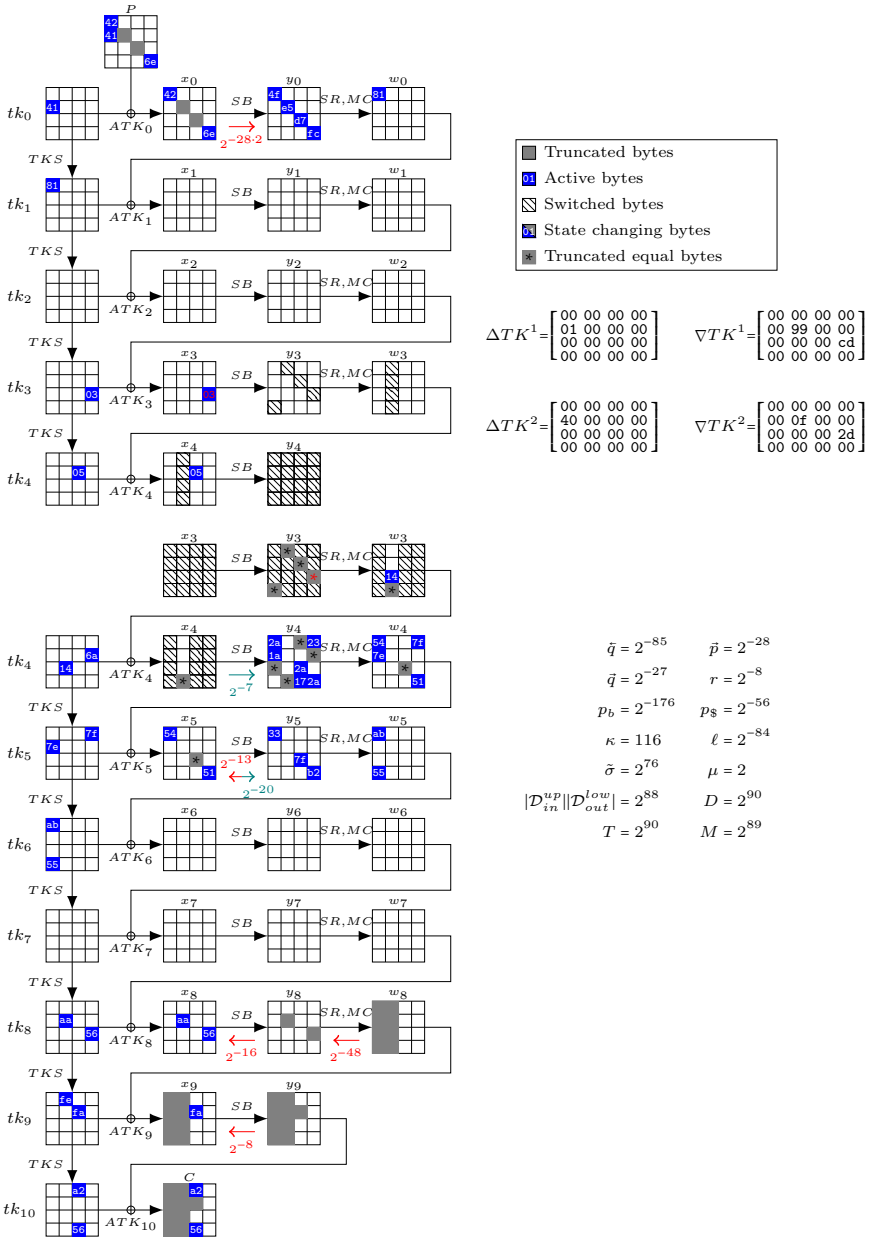


Fig. 11. Truncated boomerang attack on 10-round Deoxys-BC in the RTK2 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

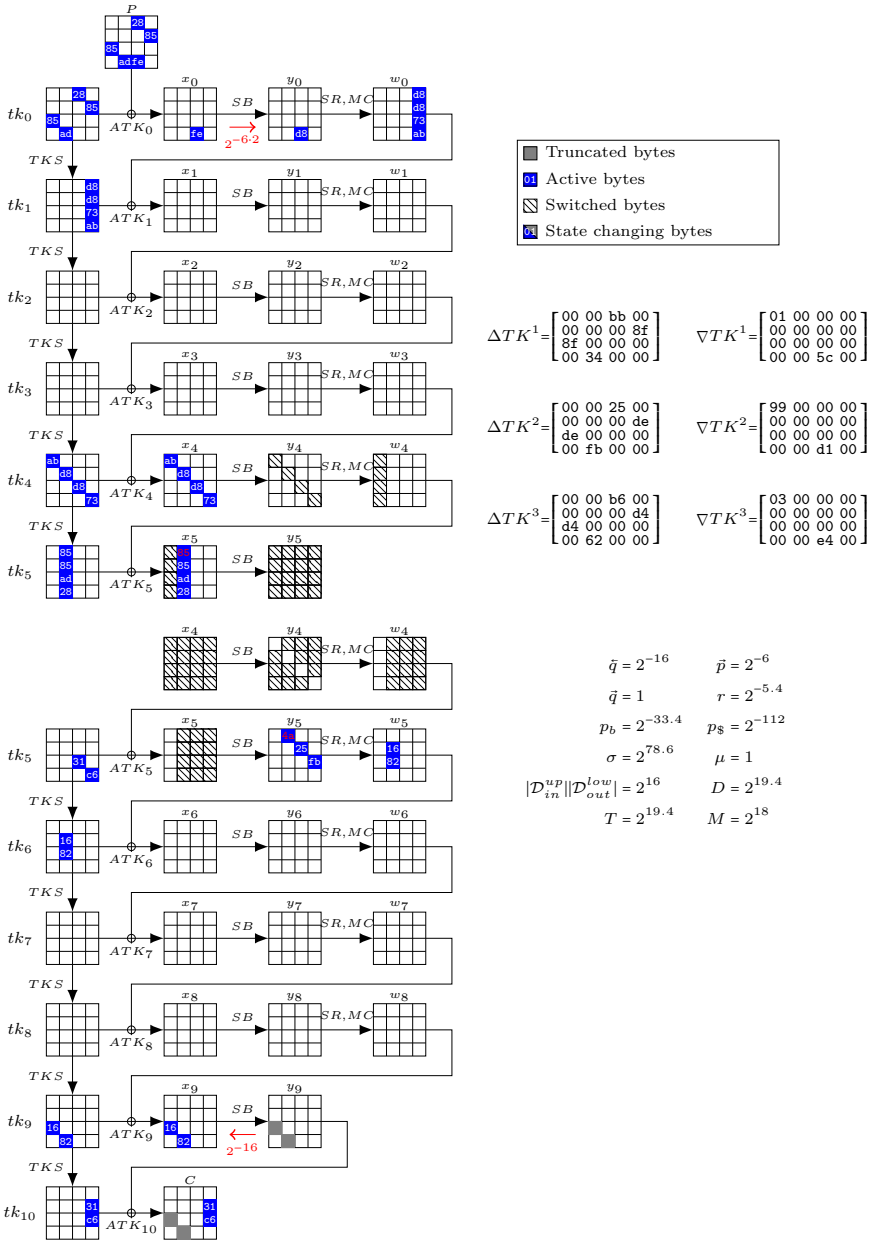


Fig. 12. Truncated boomerang attack on 10-round Deoxys-BC in the RTK3 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

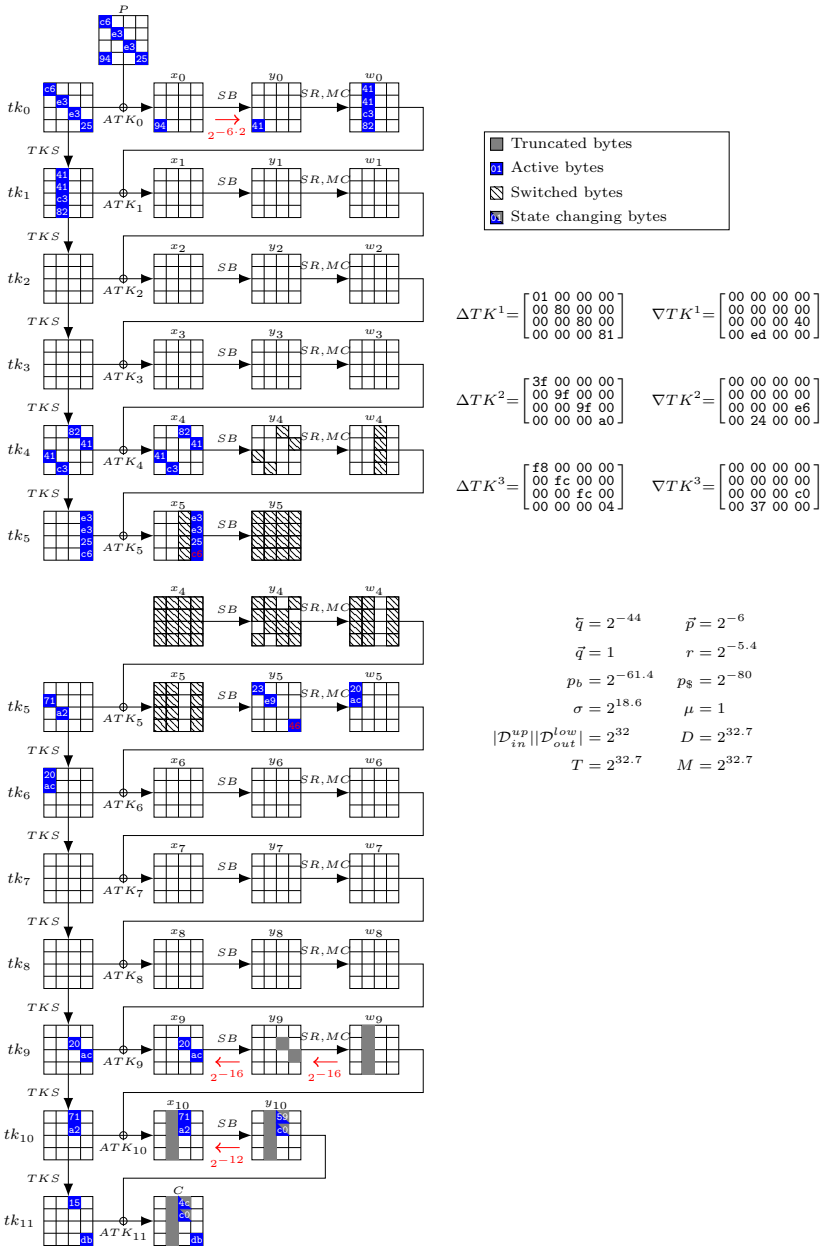


Fig. 13. Truncated boomerang attack on 11-round Deoxys-BC in the RTK3 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

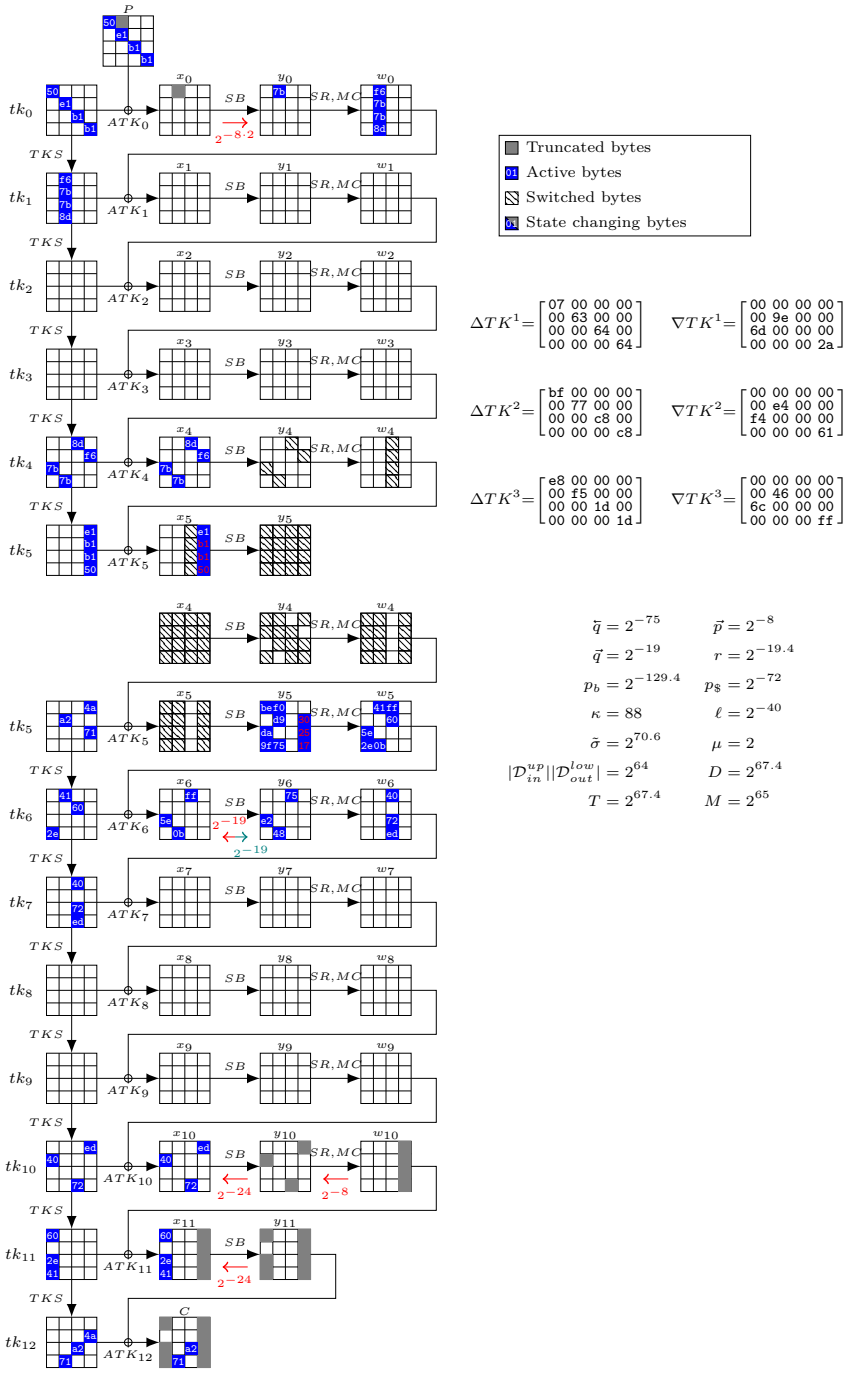


Fig. 14. Truncated boomerang attack on 12-round Deoxys-BC in the RTK3 model, starting from the ciphertext side. This attack succeeds with probability 1/2.

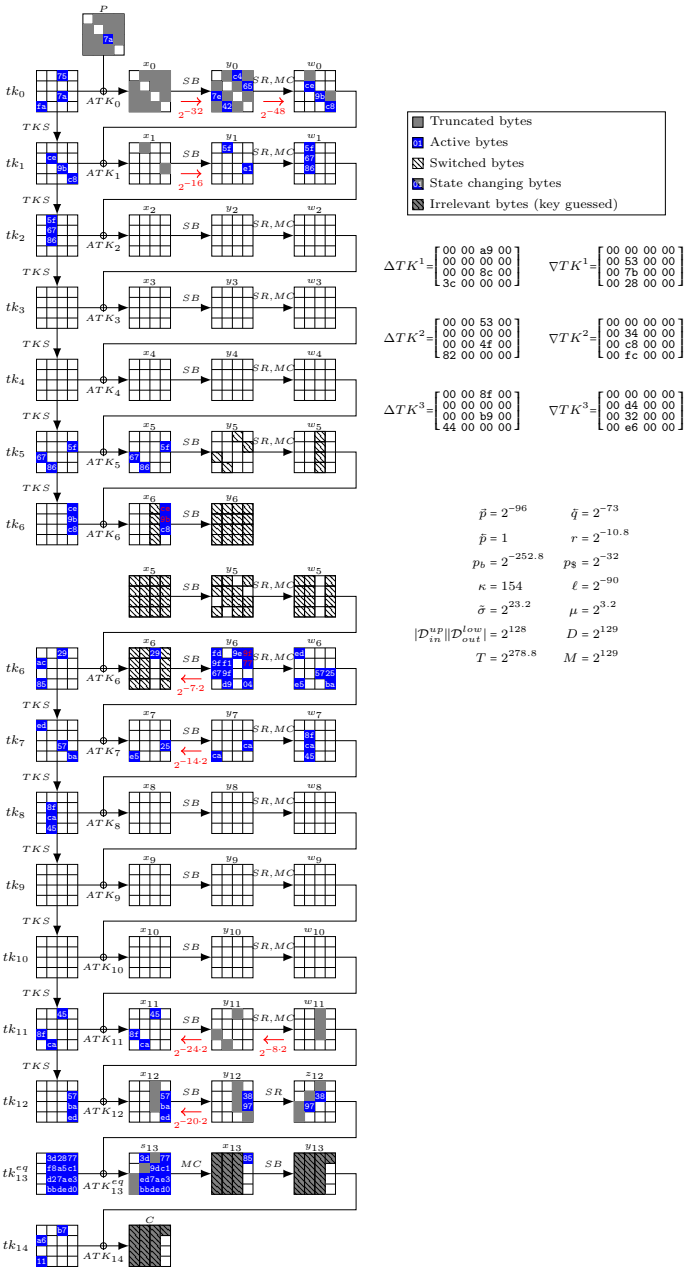


Fig. 15. Truncated boomerang attack on 14-round Deoxys-BC in the RTK3 model, starting from the plaintext side.