IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 4, pp. 28–57. DOI:10.46586/tosc.v2023.i4.28-57

Revisiting Yoyo Tricks on AES

Sandip Kumar Mondal[1], Mostafizar Rahman[2], Santanu Sarkar[3] and Avishek Adhikari[4]

[1] Department of Pure Mathematics, University of Calcutta, Kolkata, India
sandipkumarmondal80@gmail.com
[2] University of Hyogo, Kobe, Japan
mrahman454@gmail.com
[3] Department of Mathematics, Indian Institute of Technology Madras, Chennai, India
sarkar.santanu.bir1@gmail.com
[4] Department of Mathematics, Presidency University, Kolkata, India
avishek.adh@gmail.com

Abstract. At Asiacrypt 2017, Rønjom et al. presented key-independent distinguishers for different numbers of rounds of AES, ranging from 3 to 6 rounds, in their work titled "Yoyo Tricks with AES". The reported data complexities for these distinguishers were 3, 4, $2^{25.8}$, and $2^{122.83}$, respectively. In this work, we revisit those key-independent distinguishers and analyze their success probabilities.

We show that the distinguishing algorithms provided for 5 and 6 rounds of AES in the paper of Rønjom et al. are ineffective with the proposed data complexities. Our thorough theoretical analysis has revealed that the success probability of these distinguishers for both 5-round and 6-round AES is approximately 0.5, with the corresponding data complexities mentioned earlier.

We investigate the reasons behind this seemingly random behavior of those reported distinguishers. Based on our theoretical findings, we have revised the distinguishing algorithm for 5-round AES. Our revised algorithm demonstrates success probabilities of approximately 0.55 and 0.81 for 5-round AES, with data complexities of $2^{29.95}$ and $2^{30.65}$, respectively. We have also conducted experimental tests to validate our theoretical findings, which further support our findings.

Additionally, we have theoretically demonstrated that improving the success probability of the distinguisher for 6-round AES from 0.50000 to 0.50004 would require a data complexity of $2^{129.15}$. This finding invalidates the reported distinguisher by Rønjom et al. for 6-round AES.

Keywords: AES · Distinguisher · Yoyo

1 Introduction

In the modern era, the Advanced Encryption Standard (AES) [DR02] has emerged as one of the most widely used block ciphers in various applications. AES was selected through a public competition organized by the National Institute of Standards and Technology (NIST) in 2001 [NIS], with the aim of replacing the aging Data Encryption Standard (DES) [oS77]. The competition attracted 15 robust candidates from 12 countries, and ultimately, NIST announced Rijndael as the selected algorithm to become AES.

AES owes its origins to the efforts of two cryptographers, Joan Daemen and Vincent Rijmen, who developed the Rijndael algorithm. Their creation showcased remarkable security properties, efficiency, and versatility, leading to its selection as the new standard. The algorithm's strength lies in its robust security and resistance against various cryptographic attacks. AES achieves this through a series of well-defined rounds, incorporating

Received: 2023-06-01 Revised: 2023-09-01 Accepted: 2023-11-01 Published: 2023-12-08

substitution, permutation, and mixing operations. These operations introduce confusion and diffusion, ensuring that even small changes in the input produce significant changes in the output.

Over the years, many attacks have been proposed on round-reduced versions of AES. Some examples of these attacks include impossible differential [LDKK08], boomerang attack [Bir04], related-key boomerang attack [BK09], mixture-differential [Gra18], subspace attack [GRR16], yoyo [RBH17], biclique cryptanalysis [BKR11], truncated differentials [Knu94], integral cryptanalysis [DKR97], etc. Apart from these attacks, automatic search tools are also employed to improve the existing attacks. For example, Derbez et al. [DF16] improved the state-of-the-art analysis of low-complexity cryptanalysis of AES presented by Bouillaguet et al. [BDD+12] by using automated cryptanalysis.

In the realm of block ciphers, many prominent attacks are based on a distinguisher, which aims to differentiate encrypted data produced by a specific cipher from random data. If the attacker manages to create an algorithm that can accomplish this task faster than a brute force search, then it means they have successfully compromised the security of the cipher. To put it simply, let us assume an attacker designs an algorithm capable of distinguishing between a cipher (whose details are unknown to the attacker) and random data. If the input to this algorithm is a cipher, the attacker can guess the cipher with a high probability. Conversely, if the input is random data, the attacker can also guess that it is random data with a high probability. The key idea behind this attack is to exploit any patterns or biases in the cipher's output that differ from the behavior of truly random data.

Here we revisit a particular type of distinguishing attack proposed on round-reduced versions of AES which is called the yoyo attack. It was first introduced by Biham et al. to mount an attack on 16-round SKIPJACK [BBD+98]. The yoyo game is built by adaptively creating new pairs of plaintexts and ciphertexts that retain a certain property inherited from the original pair, similar to boomerang attacks [Wag99]. Zero difference between the pairs is a commonly used property. Imagine that plaintext/ciphertext has a zero difference property after some rounds of the cipher, a yoyo game verifies whether new pairs of plaintexts/ciphertexts that are formed by swapping bytes/words of the original pairs have the same zero difference after the same number of rounds. By applying the yoyo attack, Biryukov et al. [BLP16] have found a 7-round distinguisher for Feistel networks. At Asiacrypt 2017, Rønjom et al. [RBH17] analyzed the yoyo game on substitution-permutation (SP) networks. They proposed a deterministic distinguisher on two generic SP rounds. They also distinguished 5-round AES and 6-round AES by yoyo trick. In [SRP18] Saha et al. distinguished AESQ up to 16 rounds and distinguished AES up to 8 rounds in the known key setting scenario.

The success probability of a cryptographic attack, given a certain data/time/memory complexity, depends on the effectiveness of the attack algorithm. Typically, increasing the complexities involved in an attack can enhance its success probability. However, there are limitations on the allowed complexities, necessitating a trade-off to determine appropriate attack parameters. In this work, our focus centers on assessing the success probabilities of the yoyo distinguishing attacks introduced in [RBH17].

Our Contributions: In this work, we analyze the distinguisher proposed in the paper [RBH17], which claims to distinguish 5-round and 6-round AES. However, our findings reveal that the distinguishers with the data complexities presented in [RBH17] are ineffective in achieving this goal. We demonstrate that the success probability of the distinguisher, as described in [RBH17], for distinguishing 5-round and 6-round AES is only 0.5, using the data complexities mentioned in their work. Furthermore, we revise the algorithm used to distinguish 5-round AES, presenting a revised algorithm that effectively distinguishes 5-round AES from a random permutation with a significant success probability. We have

thoroughly verified this claim through experimental analysis.

Additionally, we investigate the success probability of the distinguisher for 6-round AES. Employing a similar approach, we establish that this distinguisher cannot effectively distinguish 6-round AES with a significant success probability using a data complexity lower than that of an exhaustive search. The impact of our results extends even further, rendering the impossible-differential yoyo distinguisher on `ForkAES`-*-4-4 [BBJ+19] invalid as well.

Organization of the Paper: The rest of the paper is organized as follows. In Section 2, we give some preliminary ideas about yoyo attacks. In addition to that, a brief description of AES is also provided in this section. In Section 3 we revisit the yoyo attack on AES and provide a revised distinguisher for 5-round and 6-round AES. The success probability of the revised algorithms are calculated in Section 4. Section 5 illustrates the impact of the results derived in this paper on other yoyo distinguishers. In Section 6, we give the theoretical and experimental success probabilities of the revised algorithms. Finally, the concluding remarks are furnished in Section 7.

2 Preliminaries

In this section, for the sake of completeness and comprehensibility, we first provide a brief description of AES. Next, we discuss the yoyo attack on substitution-permutation network (SPN) based primitives.

2.1 Description of AES-128

AES-128 is a block cipher with a 128-bit key length. It takes a 128-bit plaintext as input and outputs a 128-bit ciphertext. The state of AES can be represented as a 4x4 matrix over the finite field $\mathbb{F}_q$, where $q = 2^8$. One round of AES consists of the following four functions:

- **SubBytes (SB):** This function replaces each byte in the state with a new byte, using an 8-bit Sbox table.
- **ShiftRows (SR):** This function cyclically shifts each row of the state by a different amount. In general, the i-th row of the state is rotated left by i bytes (for $0 \leq i \leq 3$).
- **MixColumns (MC):** This function mixes the columns of the state using a linear transformation. The matrix of the linear transformation is

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix},$$

- **AddRoundKey (ARK):** This function adds the round subkey (generated from the secret key) to the state.

2.2 Yoyo Game on Substitution-Permutation Networks

Here, we are stating a few definitions and results (regarding the yoyo attack) adapted from [RBH17] that will be used throughout the paper.

Definition 1. Zero Difference Pattern: (Definition 1 of [RBH17]) Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}_q^n$. Define $\nu(\alpha) = (z_0, z_1, \ldots, z_{n-1}) \in \mathbb{F}_2^n$ where $z_i = 1$ if $\alpha_i = 0$ and $z_i = 0$ otherwise. Then $\nu(\alpha)$ is the Zero Difference Pattern for α.

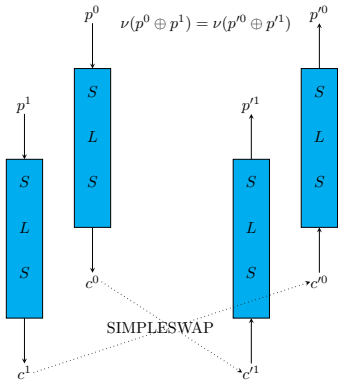

Figure 1: Generic yoyo game for 2-rounds

In the context of differential cryptanalysis where the α_i's are considered as either active or inactive bytes, $\nu(\alpha)$ represents a pattern with respect to active/inactive bytes. If α_i is active, then $z_i = 0$ and vice-versa. It is worth noting that the Hamming weight of the vector $\nu(\alpha)$, denoted as $wt(\nu(\alpha))$, corresponds to the number of non-zero elements in $\nu(\alpha)$.

Next, an important operation, named `swapping`, is defined for a simpler description of the yoyo game.

Definition 2. (Definition 2 of [RBH17]) For a vector $v \in \mathbb{F}_2^n$ and a pair of states $\alpha, \beta \in \mathbb{F}_q^n$ define a new state $\rho^v(\alpha, \beta) \in \mathbb{F}_q^n$ such that the i-th component is defined by

$$\rho^v(\alpha, \beta)_i = \begin{cases} \alpha_i, & \text{if } v_i = 1 \\ \beta_i, & \text{if } v_i = 0. \end{cases} \tag{1}$$

The following proposition describes the yoyo game for 2 generic SPN rounds (G_2'). Thus, $G_2' = L \circ S \circ L \circ S$, where S represents the substitution layer and L represents the linear layer. Note that, in the context of differential, the last linear layer has no effect. So, the following proposition considers a modified version of G_2' i. e., $G_2 = S \circ L \circ S$.

Proposition 1. *(Theorem 2 in [RBH17]) Let $G_2 = S \circ L \circ S$ be two generic SP-rounds. Let $p^0 \oplus p^1 \in \mathbb{F}_q^n$, $c^0 = G_2(p^0)$ and $c^1 = G_2(p^1)$. For any $v \in \mathbb{F}_2^n$, let $c'^0 = \rho^v(c^0, c^1)$ and $c'^1 = \rho^v(c^1, c^0)$. Then*

$$\nu(G_2^{-1}(c'^0) \oplus G_2^{-1}(c'^1)) = \nu(p'^0 \oplus p'^1) = \nu(p^0 \oplus p^1).$$

Proposition 1 states a scenario when the zero difference property remains invariant for a pair of plaintexts. This proposition is visually depicted in Figure 1. The function ρ^v is used twice for generating a new pair of states from a given pair of states. In the experiment, we use the function `SIMPLESWAP` (given in Algorithm 1) for generating a new pair from the given pair.

Algorithm 1: [RBH17] Swaps the first word where texts are different and returns one text

```
function SIMPLESWAP(x^0, x^1)
x'^0 ← x^1
for i from 0 to 3 do
    if x_i^0 ≠ x_i^1 then
        x'^0_i ← x_i^0
        return x'^0
```

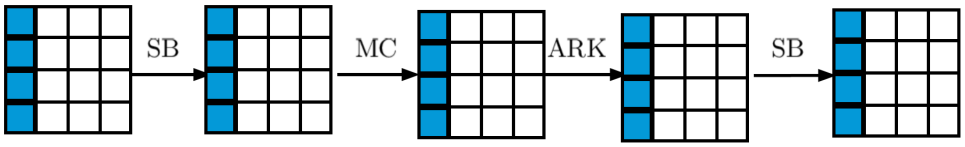

Figure 2: Super-sbox of AES

3 Revisiting the Yoyo Distinguishing Attacks on AES

In this section, we discuss some related important properties. We also provide a concise overview of our investigation into the ineffectiveness of the 5-round yoyo distinguishing attack on AES, as proposed in [RBH17]. Next, we look into the possible reasons behind the ineffectiveness of the attack and propose a revised version of it. Additionally, we undertake a thorough analysis of the 6-round attack on AES.

3.1 Overview of the Yoyo Distinguishers

In [RBH17], the 5-round and 6-round AES are denoted as $R^5 = S \circ L \circ S \circ Q$ and $R^6 = S \circ L \circ S \circ L \circ S$ where $S = SB \circ MC \circ SB$, $L = SR \circ MC \circ SR$ (this is fixed after considering that SB and SR can be interchanged in between the rounds) and $Q = SR \circ MC \circ SB$. Here, S is the `super-sbox` [DR06] of AES that acts from a column to a column (as shown in Figure 2). For more details regarding this, refer to [RBH17].

We now state two propositions that can be used to distinguish 5-round AES from a random permutation. In this context, the term "word" refers to a 4-byte long unit of data. This is the same as 32 bits. The AES algorithm uses a 4x4 matrix to store the data it is encrypting or decrypting. Each column of this matrix is referred to as a word.

Proposition 2. *(Theorem 4 of [RBH17]) Let $a \in \mathbb{F}_q^{4\times4}$ and $b \in \mathbb{F}_q^{4\times4}$ denote two states, where the zero difference pattern $\nu(Q(a) \oplus Q(b))$ has weight t. Then the probability that any $4-t$ bytes are simultaneously zero in a word in the difference $a \oplus b$ is q^{t-4}. When this happens, all bytes in the difference are zero.*

Proposition 2 states that if $wt(\nu(Q(a) \oplus Q(b))) = t$, then any column of $a \oplus b$ has either at most $4-t$ inactive bytes or all of the bytes in that column are inactive.

Proposition 3. *([RBH17]) Let p^0 and p^1 be two states such that $wt(\nu(p^0 \oplus p^1)) = t$. Let $p'^0 = \rho^v(Q^{-1}(p^0), Q^{-1}(p^1))$ and $p'^1 = \rho^v(Q^{-1}(p^1), Q^{-1}(p^0))$. Then*

$$wt(\nu(Q(p'^0) \oplus Q(p'^1))) = t.$$

Proposition 3 says that if we take two states and create two states by using the ρ function, then the weight of the XOR of these two pairs after the operation Q is unchanged.

Next, we state a proposition that gives a bound on the maximum number of inactive words (in the context of AES) in the input and output of $S \circ L \circ S$ layer. This result can be deduced by extending a *well-known* property which explains that at most 3 bytes can be inactive (out of 8 bytes) in the input and output of MC (considering all bytes are not inactive). In [RBH17], this property is primarily exploited to mount a distinguishing attack on 6-round AES.

Proposition 4. *[DR07] Let p^0 and p^1 be two different states of AES. Then*

$$wt(\nu(p^0 \oplus p^1)) + wt(\nu(S \circ L \circ S(p^0) \oplus S \circ L \circ S(p^1))) \leq 3.$$

Here we give an important remark regarding the complexity calculation in the paper [RBH17].

Remark 1. In the paper [RBH17], the authors take 1 encryption query and 1 decryption query together as 1 in the data complexity. In our paper, the data complexity is calculated by summing all individual encryption queries and decryption queries. According to our calculation, the complexities reported in the paper [RBH17] would be multiplied by a factor of 2. Henceforth we write their complexity multiplied by a factor of 2.

Yoyo Attacks on AES: Experimental Evaluation

In [RBH17], the authors reported key-independent yoyo distinguishers for 5-round and 6-round AES with data complexities of $2^{26.8}$ and $2^{123.83}$ respectively. The time complexity of the attack is $2^{26.8}$ memory accesses and $2^{24.4}$ XOR operations. The parameterized version of the 5-round AES distinguisher is outlined in Algorithm 2, with $x = 2^{13.4}$, $y = 2^{11.4}$, and $t = 2$. From the attack complexity, it is quite evident that the 5-round AES distinguisher can be practically verified. In fact, the distinguisher is able to distinguish 5-round AES with an overwhelming probability (experimentally, we have found that it always distinguishes 5-round AES). However, a challenge arises when attempting to distinguish a random permutation from a 5-round AES. In that case, also, the distinguisher is always distinguishing the random permutation as 5-round AES. Consequently, the success probability of the distinguisher is reduced to 0.5.

3.2 Detailed analysis of the Proposed Attack on 5-round AES

This section describes the revised algorithms for distinguishing the round-reduced AES from a random permutation by yoyo attack. In the revised algorithm we use x, y and t as input variables, where x represents the number of chosen plaintext pairs and y represents the number of adaptive chosen plaintext/ciphertext pairs. For a particular case, if we take $x = 2^{13.4}, y = 2^{11.4}$ and $t = 2$ then Algorithm 2 reduces to the algorithm as described in [RBH17] for distinguishing 5-round AES.

Algorithm 2: Distinguisher for 5-round AES

Input: x, y and t
Output: 1 for the AES and -1 otherwise.

$i \leftarrow 0$
while $i < x$ **do**
  $i \leftarrow i + 1$
  $p^{i,1}, p^{i,2} \leftarrow$ generate random pair with $wt(\nu(p^{i,1} \oplus p^{i,2})) = 3$
  $j \leftarrow 0$, `WrongPair` $\leftarrow$ False
  while $j < y$ & *WrongPair* $=$ *False* **do**
    $j \leftarrow j + 1$
    $c^{i,2j-1} \leftarrow enc_k(p^{i,2j-1}, 5)$, $c^{i,2j} \leftarrow enc_k(p^{i,2j}, 5)$
    $c'^0 \leftarrow$ `SIMPLESWAP`$(c^{i,2j-1}, c^{i,2j})$, $c'^1 \leftarrow$ `SIMPLESWAP`$(c^{i,2j}, c^{i,2j-1})$
    $p'^0 \leftarrow dec_k(c'^0, 5)$, $p'^1 \leftarrow dec_k(c'^1, 5)$
    $\Delta p^{i,j} \leftarrow (p'^0 \oplus p'^1)$
    for k *from* 0 *to* 3 **do**
      if $4 - t \leq wt(\nu(\Delta p^{i,j}_k)) < 4$ **then** ▷ Here $\Delta p^{i,j}_k$ is the k^{th} column of $\Delta p^{i,j}$
        `WrongPair` $\leftarrow$ True
    $p^{i,2j+1} \leftarrow$ `SIMPLESWAP`(p'^0, p'^1), $p^{i,2j+2} \leftarrow$ `SIMPLESWAP`(p'^1, p'^0)
  if *WrongPair* $=$ *False* **then**
    return 1
return -1

Here we discuss how this algorithm works for distinguishing 5-round AES from a

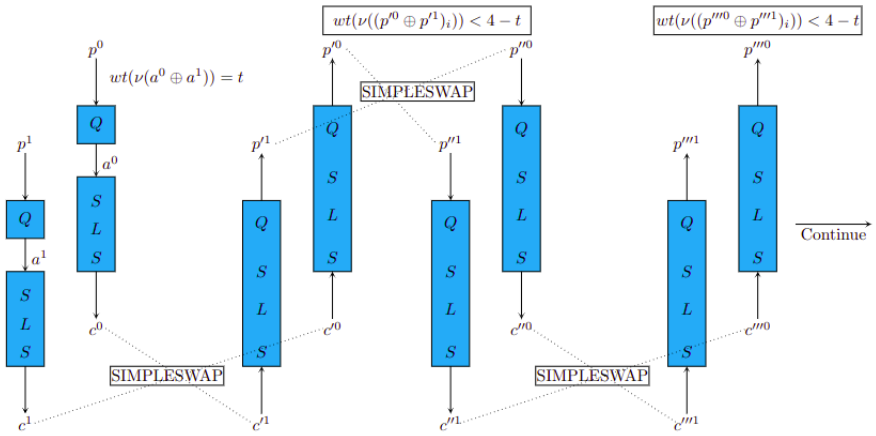

Figure 3: `RightPair` for 5-round AES Distinguisher

random permutation. We start by introducing the terms `WrongPair` and `RightPair` as used in Algorithm 2. Let $t \in \{1, 2, 3\}$ be a fixed value. When the oracle is 5-round AES, a pair p^0, p^1 is said to be a `RightPair` if $wt(\nu(Q(p^0) \oplus Q(p^1))) \geq t$. Note that, this property is required to be satisfied in the intermediate round. Based on this intermediate-round property, some probabilistic property on the final output is derived to correctly detect a `RightPair`. Conversely, a pair p^0, p^1 is considered a `WrongPair` if it does not meet the intermediate-round criteria. When the oracle is a random permutation, then every pair is supposed to be a `WrongPair`. In Figure 3 we demonstrate a `RightPair` for 5-round AES.

When we execute Algorithm 2 if one can find a `RightPair` then the algorithm returns AES. Otherwise, if all pairs are `WrongPairs`, then the algorithm returns a random permutation.

In this context, we represent a range using the notation $[a, b]$, where $[a, b]$ denotes the set of integers from a to b inclusive, with a and b being integers and $a < b$. In Algorithm 2, initially, x chosen plaintext pairs are randomly generated (as shown in line 4) which are queried to the oracle. Out of these x pairs, if at least one pair is detected as `RightPair`, then the oracle is identified as 5-round AES, otherwise it is identified as a random permutation. Consider that in Algorithm 2, $wt(\nu(Q(p^{i,1}) \oplus Q(p^{i,2}))) \geq t$ occurs for some $i \in [1, x]$ (i. e. the pair is `RightPair`). Then by Proposition 1, 2 and 3, the condition $4 - t \leq wt(\nu(\Delta p_k^{i,j})) < 4$ (in line 13 of Algorithm 2) is not satisfied for $1 \leq j \leq y$ (for a more detailed explanation, please refer to [RBH17]).

If the pair $p^{i,1}, p^{i,2}$ is a `WrongPair`, the condition $4 - t \leq wt(\nu(\Delta p_k^{i,j})) < 4$ in line 13 of Algorithm 2 is probabilistically satisfied for some sufficiently large value of y. So in Algorithm 2, the value of x is chosen such that when the oracle is 5-round AES, at least one `RightPair` is probabilistically generated. The value of y is chosen such that the condition in line 13 is satisfied at least once for a `WrongPair`.

In Remark 2 we explain when Algorithm 2 can distinguish between 5-round AES and random permutation.

Remark 2. According to Algorithm 2, the distinguisher can correctly identify 5-round AES if there exists an $i \in [1, x]$ for which the condition $4 - t \leq wt(\nu(\Delta p_k^{i,j})) < 4$ (line 13 of Algorithm 2) is not satisfied for any $j \in [1, y]$ and $k \in [0, 3]$. The distinguisher can correctly identify a random permutation if for all $i \in [1, x]$ the condition $4 - t \leq wt(\nu(\Delta p_k^{i,j})) < 4$ is satisfied for some $j \in [1, y]$ and for some $k \in [0, 3]$.

In [RBH17], 5-round AES distinguisher is shown to be effective for $t = 2$, $x = 2^{13.4}$ and $y = 2^{11.4}$. The attack starts by choosing pairs $p^{i,1}$ and $p^{i,2}$ such that $wt(\nu(p^{i,1} \oplus p^{i,2})) = 3$ for all $i \in [1, x]$. Then the probability such that $wt(\nu(Q(p^{i,1}) \oplus Q(p^{i,2}))) \geq 2$ occurs (i.e, a `RightPair` is generated) is $\binom{4}{2}(\frac{1}{2^8})^2 = \frac{1}{2^{13.4}}$. Hence, $x = 2^{13.4}$ is chosen and a `RightPair` is expected to be generated with high probability.

When $wt(\nu(Q(p^{i,1}) \oplus Q(p^{i,2}))) \geq 2$, in [RBH17] it is demonstrated using Proposition 1 and 2 that the condition $2 \leq wt(\nu(\Delta p_k^{i,j}))$ (in line 13 of Algorithm 2) is not satisfied for any value of j. As a result, Algorithm 2 detects the oracle as 5-round AES. Now for a `WrongPair`, the condition $2 \leq wt(\nu(\Delta p_k^{i,j}))$ (in line 13 of Algorithm 2) is satisfied with probability $4\binom{4}{2}(\frac{1}{2^8})^2 = \frac{1}{2^{11.4}}$. The value $y = 2^{11.4}$ determines that from the initial pair, $2^{11.4}$ pairs are generated by using the `SIMPLESWAP` operation which ensures that the zero difference property remains invariant between these pairs (when the oracle is 5-round AES). However, for a random permutation, the condition $2 \leq wt(\nu(\Delta p_k^{i,j}))$ is satisfied with high probability. Thus, if at least a pair $(p^{i,1}, p^{i,2})$ is generated out of $2^{13.4}$ possible pairs such that $wt(\nu(Q(p^{i,1}) \oplus Q(p^{i,2}))) = 2$, then for that pair the condition $2 \leq wt(\nu(\Delta p_k^{i,j}))$ is never satisfied when the oracle is 5-round AES. However, we have observed that the above condition is not complete and this issue is addressed in Remark 3.

Remark 3. In Algorithm 2, line 13, there is a discrepancy between the condition stated in [RBH17] and the correct condition. The condition given in [RBH17] is $wt(\nu(\Delta p_k^{i,j})) \geq 2$ for $t = 2$. But the correct condition should be $2 \leq wt(\nu(\Delta p_k^{i,j})) < 4$. Since from Proposition 2 it may happen that for some pair $p^{i,1}$, $p^{i,2}$ satisfying $wt(\nu(Q(p^{i,1}) \oplus Q(p^{i,2}))) = 2$ but there exists $k \in \{0, 1, 2, 3\}$ such that $wt(\nu(\Delta p_k^{i,j})) = 4$ [1]. The probability of obtaining a pair such that $wt(\nu(\Delta p_k^{i,j})) = 4$ is $4 \times (2^{-8})^4 = 2^{-30}$, which is considered negligible. However, this condition is essential in the algorithm as it prevents some `RightPair` from being detected as `WrongPair`.

However, the above-mentioned claims are not entirely correct in the distinguishing attack setting. In Observation 1, we identify the shortcomings of the distinguisher, and in Observation 2, we propose a possible correction for the 5-round AES distinguisher.

Observation 1. *In [RBH17], it is explained that if for a pair of plaintexts p^0 and p^1, $wt(\nu(Q(p^0) \oplus Q(p^1))) = 2$ is satisfied, then 5-round AES is distinguished by Algorithm 2. However, the authors did not mention the probability of correctly identifying a random permutation using Algorithm 2. Algorithm 2 returns a random permutation if for each $i \in [1, x]$, the condition in line 13 of Algorithm 2 is satisfied for some $j \in [1, y]$ and $k \in [0, 3]$. But such a value of y is chosen such that the condition in line 13 of Algorithm 2 is satisfied only for a fixed value of $i \in [1, x]$.*

Observation 2. *In Table 1 we give our experimental results for $x = 2^{13.4}$, $y = 2^{11.4}$ and $t = 2$. We see that the distinguisher identifies 5-round AES with probability 1 and identifies a random permutation with probability 0. So the overall success probability for distinguishing 5-round AES is 0.5. In Section 4, our theoretical results align with the experimental findings. It is worth noting that in every case, we get at least one pair (from x pairs created in line 4 of Algorithm 2) for which the condition in line 13 does not hold. In Observation 1, we point out that the value of y should be chosen such that for every pair (from x pairs) the condition in line 13 is satisfied at least once in the loop of line 6. For this reason, we increase the value of y continuously and find the success probability of Algorithm 2.*

[1] Note that in the source code of [RBH17], they consider the condition $2 \leq wt(\nu(\Delta p_k^{i,j})) < 4$.

Table 1: Experimental results for 5-round AES when t=2. Here, $\#N$ denotes the number of experiments.

$\#N$	Blackbox Primitive	x	y	Detected as AES	Detected as Random Permutation	Experimental Success Probability
100	AES	$2^{13.4}$	$2^{11.4}$	100	0	0.5
100	Random Permutation	$2^{13.4}$	$2^{11.4}$	100	0	

3.3 Detailed analysis of the Proposed Attack on 6-round AES

The distinguisher for 6-round AES is given in Algorithm 3. In the revised algorithm we use x, y and t as input variables. For a particular case, if we take $x = 2^{61.4}, y = 2^{60.4}$ and $t = 2$ then Algorithm 3 reduces to the algorithm as described in [RBH17] for distinguishing 6-round AES.

Algorithm 3: Distinguisher for 6-round AES

Input: x, y and t
Output: 1 for the AES and -1 otherwise.

1 $i \leftarrow 0$
2 **while** $i < x$ **do**
3 $\quad i \leftarrow i + 1$
4 $\quad p^{i,1}, p^{i,2} \leftarrow$ generate random pair with $p^{i,1} \neq p^{i,2}$
5 $\quad j \leftarrow 0$, `WrongPair` $\leftarrow$ False
6 $\quad$ **while** $j < y$ & *WrongPair* = *False* **do**
7 $\quad\quad j \leftarrow j + 1$
8 $\quad\quad \Delta p^{i,j} \leftarrow p^{i,2j-1} \oplus p^{i,2j}$
9 $\quad\quad$ **if** $wt(\nu(\Delta p^{i,j})) \geq 4 - t$ **then**
10 $\quad\quad\quad$ `WrongPair` = True
11 $\quad\quad c^{i,2j-1} \leftarrow enc_k(p^{i,2j-1}, 6)$, $c^{i,2j} \leftarrow enc_k(p^{i,2j}, 6)$
12 $\quad\quad \Delta c^{i,j} \leftarrow c^{i,2j-1} \oplus c^{i,2j}$
13 $\quad\quad$ **if** $wt(\nu(\Delta c^{i,j})) \geq 4 - t$ **then**
14 $\quad\quad\quad$ `WrongPair` = True
15 $\quad\quad c'^0 \leftarrow$ `SIMPLESWAP`$(c^{i,2j-1}, c^{i,2j})$, $c'^1 \leftarrow$ `SIMPLESWAP`$(c^{i,2j}, c^{i,2j-1})$
16 $\quad\quad p'^0 \leftarrow dec_k(c'^0, 6)$, $p'^1 \leftarrow dec_k(c'^1, 6)$
17 $\quad\quad p^{i,2j+1} \leftarrow$ `SIMPLESWAP`(p'^0, p'^1), $p^{i,2j+2} \leftarrow$ `SIMPLESWAP`(p'^1, p'^0)
18 $\quad$ **if** *WrongPair* = *False* **then**
19 $\quad\quad$ **return** 1
20 **return** -1

Like the 5-round AES distinguisher, the success of the 6-round AES distinguisher also depends on an intermediate relation $wt(\nu(L \circ S(p^{i,1}) \oplus L \circ S(p^{i,2}))) \geq t$, where $t \in \{1, 2, 3\}$. If the above relation happens for a pair $(p^{i,1}, p^{i,2})$ then for that pair the conditions $(wt(\nu(\Delta p^{i,j})) \geq 4 - t$ and $wt(\nu(\Delta c^{i,j})) \geq 4 - t)$ in lines 9 and 13 of Algorithm 3 are not satisfied for any value of y (for a detailed explanation, refer to [RBH17]). In this case, a pair p^0, p^1 is said to be a `RightPair` if $wt(\nu(L \circ S(p^0) \oplus L \circ S(p^1))) \geq t$. In Figure 4, a `RightPair` is shown for a 6-round AES distinguisher. In Section 4, we calculate the success probability of Algorithm 3.

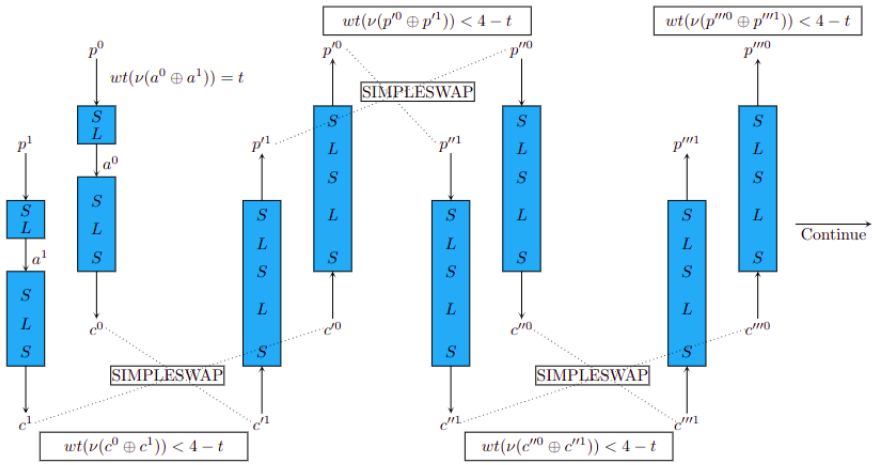

Figure 4: `RightPair` for 6-round Distinguisher

4 Success Probability of the Revised Yoyo Distinguishers

In this section, we critically analyze the 5-round and 6-round AES distinguishers reported in [RBH17] and compute their respective success probabilities. First, we analyze the success probability of the distinguisher for 5-round AES. Then, the 6-round AES distinguisher is analyzed.

4.1 Success Probability of the Yoyo Distinguisher on 5-round AES

Algorithm 2 is a distinguisher for 5-round AES. Since Algorithm 2 depends on the values of x, y and t we called this distinguisher as $\mathcal{D}^{x,y,t}_{AES_5}$. We say that a distinguisher is effective if its success probability is high. Here we use the notation $p^{x,y,t}_{AES_5}$ and $p^{x,y,t}_{\mathcal{RP}_5}$ for the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ can correctly identify 5-round AES and random permutation respectively. To find the above two probabilities, first of all, we introduce some lemmas. Note that we denote the probability of an event $\mathcal{A}$ by $P(\mathcal{A})$.

Lemma 1. *Consider* $s \xleftarrow{\$} \mathbb{F}^4_{2^8}$. *Then*

$$P[wt(\nu(s)) = k] = \binom{4}{k}(q^{-1})^k(1-q^{-1})^{(4-k)},$$

where $q = 2^8$.

Proof. Since k bytes from 4 bytes can be chosen in $\binom{4}{k}$ ways, the probability that s has exactly k inactive bytes is

$$P[wt(\nu(s)) = k] = \binom{4}{k}(q^{-1})^k(1-q^{-1})^{(4-k)}.$$

□

Lemma 2. *Consider* $p^0, p^1 \xleftarrow{\$} \mathbb{F}^{4\times 4}_{2^8}$ *with* $wt(\nu(p^0 \oplus p^1)) = 3$. *Let* A_i *be the event that* $wt(\nu(Q(p^0) \oplus Q(p^1))) = i$ *where* $i \in \{0,1,2,3\}$ *and* $Q = SR \circ MC \circ SB$. *Then* $P(A_i) = \binom{4}{i}(q^{-1})^i(1-q^{-1})^{4-i}$ *where* $q = 2^8$.

Proof. Inputs p^0, p^1 are randomly chosen with $wt(\nu(p^0 \oplus p^1)) = 3$. As SB transforms a non-zero byte difference into a non-zero byte difference, $wt(\nu(SB(p^0) \oplus SB(p^1))) = 3$. It is clear that after MC operation if there are i active bytes in the active column then after SR operation we get $wt(\nu(Q(p^0) \oplus Q(p^1))) = i$. Therefore the required probability depends on the MC operations only. So $P(A_i) = \binom{4}{i}(q^{-1})^i(1-q^{-1})^{4-i}$. □

Now we aim to determine the success probability of Algorithm 2. Let us define a set $\mathcal{X}_i = \{\Delta p^{i,1}, \Delta p^{i,2}, \ldots, \Delta p^{i,y}\}$. The set $\mathcal{X}_i$ is deduced from the yoyo game. The event $\mathcal{A}^t_{\mathcal{X}_i}$ denotes that $wt(\nu(\Delta p^{i,j}_k)) \in [0, 3-t] \cup \{4\}$, for all $1 \le j \le y$ and $0 \le k \le 3$. Let $\omega_{i,j,k}$ and $\omega'_{i,j}$ denote $wt(\nu(\Delta p^{i,j}_k))$ and $wt(\nu(Q(p^{i,2j-1}) \oplus Q(p^{i,2j})))$ respectively. Note that due to Propositions 1 and 3, for a fixed i, $\omega'_{i,1} = \omega'_{i,2} = \cdots = \omega'_{i,y}$.

The distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ returns 5-round AES if there exists $i \in [1, x]$ such that the event $\mathcal{A}^t_{\mathcal{X}_i}$ occurs. Therefore the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ returns 5-round AES is $P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i})$. First, we calculate the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ can correctly identify 5-round AES, i.e. $p^{x,y,t}_{AES_5}$.

Probability of the event $(\mathcal{A}^t_{\mathcal{X}_i})$ when the encryption function (the oracle) is 5-round AES:

In this context, we represent $\sum_{r\in[a,b]} f(r) = f(a) + f(a+1) + \cdots + f(b)$, where f is a function with r as a variable.

Let $\mathcal{X}_i = \{\Delta p^{i,1}, \Delta p^{i,2}, \ldots, \Delta p^{i,y}\}$. Then the event $(\mathcal{A}^t_{\mathcal{X}_i})^c$ occurs if $\omega_{i,j,k} \in [4-t, 3]$ for some $j \in [1, y]$ and for some $k \in [0, 3]$. The occurrence of this event depends on the value of $\omega'_{i,1} = \omega'_{i,2} = \cdots = \omega'_{i,y} = m$, where $m \in \{0, 1, 2, 3\}$.

Case 1: $\mathbf{m \ge t}$. From Proposition 2 we know that $\omega_{i,j,k} \notin [4-t, 3]$ for every $j \in [1, y]$ and $k \in [0, 3]$. Therefore, the probability that the event $(\mathcal{A}^t_{\mathcal{X}i})^c$ occurs is 0.

Case 2: $\mathbf{m < t}$. From Proposition 2 we know that $\omega_{i,j,k} \in [0, 3-m] \cup \{4\}$ for every $j \in [1, y]$ and $k \in [0, 3]$. So the event $(\mathcal{A}^t_{\mathcal{X}_i})^c$ occurs if there exist $j \in [1, y]$ and $k \in [0, 3]$ such that $\omega_{i,j,k} \in [4-t, 3-m]$. Let us analyze the probabilities related to the event $(\mathcal{A}^t_{\mathcal{X}_i})$:

- Using Lemma 1, the probability that, for a fixed j and k, $\omega_{i,j,k} \in [4-t, 3-m]$ is given by $\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)}$.
- The probability that, for a fixed j and k, $\omega_{i,j,k} \notin [4-t, 3-m]$ is $1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)}$.
- The probability that, for every $j \in [1, y]$ and $k \in [0, 3]$, $\omega_{i,j,k} \notin [4-t, 3-m]$ is $(1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)})^{4y}$.
- Therefore, the probability that the event $(\mathcal{A}^t_{\mathcal{X}i})$ occurs is $(1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)})^{4y}$.

Therefore from Case 1 and Case 2, we have

$$
\begin{aligned}
P(\mathcal{A}^t_{\mathcal{X}_i}) &= \sum_{m\in[0,3]} P(\mathcal{A}^t_{\mathcal{X}_i}/A_m)P(A_m) \text{ (where } A_m \text{ is the event that } \omega'_{i,1} = m) \\
&= \sum_{\substack{m<t \\ m\in[0,3]}} P(\mathcal{A}^t_{\mathcal{X}_i}/A_m)P(A_m) + \sum_{\substack{m\ge t \\ m\in[0,3]}} P(\mathcal{A}^t_{\mathcal{X}_i}/A_m)P(A_m) \\
&= \sum_{\substack{m<t \\ m\in[0,3]}} P(\mathcal{A}^t_{\mathcal{X}_i}/A_m)P(A_m) + \sum_{\substack{m\ge t \\ m\in[0,3]}} P(A_m) \\
&= \sum_{\substack{m<t \\ m\in[0,3]}} (1 - \sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)})^{4y} \times \kappa_m + \sum_{\substack{m\ge t \\ m\in[0,3]}} \kappa_m
\end{aligned}
$$

where we take $\kappa_m = P(A_m) = \binom{4}{m}(q^{-1})^m(1-q^{-1})^{4-m}$ (from Lemma 2).

Probability of the event $\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i}$ when the encryption function (the oracle) is 5-round AES:

$$
\begin{aligned}
p^{x,y,t}_{AES_5} &= P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i}) \\
&= 1 - P((\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i})^c) \\
&= 1 - P(\bigcap_{i=1}^{x} (\mathcal{A}^t_{\mathcal{X}_i})^c) \\
&= 1 - \prod_{i=1}^{x} P((\mathcal{A}^t_{\mathcal{X}_i})^c) \text{ [Since the events } \mathcal{A}^t_{\mathcal{X}_i} \text{are independent events]} \\
&= 1 - \prod_{i=1}^{x} (1 - P(\mathcal{A}^t_{\mathcal{X}_i})) \\
&= 1 - (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x \qquad (2) \\
&= 1 - (1 - (\sum_{\substack{m<t \\ m\in[0,3]}} (1 - \sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)})^{4y} \times \kappa_m + \sum_{\substack{m\geq t \\ m\in[0,3]}} \kappa_m))^x.
\end{aligned}
$$

Now we find the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ can correctly identify a random permutation, i.e. $p^{x,y,t}_{\mathcal{RP}_5}$. The probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ returns a random permutation is given by $1 - P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i})$.

Probability of the event $(\mathcal{A}^t_{\mathcal{X}_i})$ when the encryption function (the oracle) is a random permutation:

- Let $\mathcal{X}_i = \{\Delta p^{i,1}, \Delta p^{i,2}, \ldots, \Delta p^{i,y}\}$.
- The event $(\mathcal{A}^t_{\mathcal{X}i})^c$ occurs if there exists $j \in [1, y]$ and $k \in [0, 3]$ such that $\omega_{i,j,k} \in [4-t, 3]$.
- Using Lemma 1 the probability that, for a fixed j and k, $\omega_{i,j,k} \in [4-t, 3]$ is $\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)}$.
- The probability that, for a fixed j and k,$\omega_{i,j,k} \notin [4-t, 3]$ is $1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)}$.
- The probability that, for every $j \in [1, y]$ and $k \in [0, 3]$,$\omega_{i,j,k} \notin [4-t, 3]$ is $(1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)})^{4y}$.
- The probability that the event $(\mathcal{A}^t_{\mathcal{X}_i})$ occurs is $(1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-1})^r(1-q^{-1})^{(4-r)})^{4y}$.

Probability of the event $\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i}$ when the encryption function is a random permutation:

From the calculation of Equation (2), we get

$$P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i}) = 1 - (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x$$

Therefore

$$\begin{aligned} p^{x,y,t}_{\mathcal{RP}_5} &= 1 - (1 - (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x) \\ &= (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x \\ &= (1 - (1 - \sum_{r \in [4-t,3]} \binom{4}{r} (q^{-1})^r (1 - q^{-1})^{(4-r)})^{4y})^x. \end{aligned}$$

Note that the 5-round yoyo distinguisher of [RBH17] always detects a random permutation as AES. In their algorithm, they choose $y = 2^{11.4}$ (as the expected number of adaptively chosen pairs required for identifying a `WrongPair` is $2^{11.4}$). For $x = 2^{13.4}$, the distinguisher is required to identify $2^{13.4}$ many `WrongPairs` (one `WrongPair` for each value of x) to correctly detect a random permutation. However, it is observed that $2^{11.4}$ pairs are sometimes not sufficient to detect a `WrongPair`. Thus the distinguisher incorrectly detects a random permutation as AES. This can also be confirmed by computing the value of $p^{x,y,t}_{\mathcal{RP}_5}$ (where $x = 2^{13.4}$, $y = 2^{11.4}$ and $t = 2$) which is close to zero.

Lemma 3. ***(Success Probability of Algorithm 2)*** *Consider a black-box function whose n_1 and n_2 instances act as 5-round AES and random permutation, respectively. Let us assume that the distinguisher described in Algorithm 2 is employed to distinguish this black-box function. Then the success probability of the distinguisher is given by*

$$\frac{n_1 \times p^{x,y,t}_{AES_5} + n_2 \times p^{x,y,t}_{\mathcal{RP}_5}}{n_1 + n_2}.$$

Proof. Let n_1 be the number of instances that act as 5-round AES and n_2 be the number of instances that act as a random permutation. The total number of instances is $n_1 + n_2$.

As the success probability of correctly identifying 5-round AES is $p^{x,y,t}_{AES_5}$, the success of Algorithm 2 from n_1 instances is $n_1 \times p^{x,y,t}_{AES_5}$. Similarly the success of Algorithm 2 from n_2 instances is $n_2 \times p^{x,y,t}_{\mathcal{RP}_5}$. Therefore the success probability of the distinguisher described in Algorithm 2 is

$$\frac{n_1 \times p^{x,y,t}_{AES_5} + n_2 \times p^{x,y,t}_{\mathcal{RP}_5}}{n_1 + n_2}.$$

□

Consider the scenario where an adversary employs the distinguisher $\mathcal{D}^{x,y,t}_{AES_5}$ to distinguish between a 5-round AES and a random permutation.

For numerical estimation of the success probability in the subsequent sections, we consider $n_1 = n_2$ and thus the success probability of Algorithm 2 becomes

$$\frac{p^{x,y,t}_{AES_5} + p^{x,y,t}_{\mathcal{RP}_5}}{2}.$$

In Algorithm 2, there are a total of xy iterations. In each iteration, we perform one encryption and one decryption, resulting in a total of $2(2xy) = 4xy$ operations (as each iteration involves encrypting a pair of plaintexts and ciphertexts). So the data complexity is $4xy$. Within each iteration, we access a total of 4 states (comprising two ciphertexts and two plaintexts) and perform XOR operations on two plaintexts. This results in a time complexity of $4xy$ for memory accesses and xy for XOR operations.

Based on our probability calculation in Remark 4, we show that the algorithm described in [RBH17] can not distinguish 5-round AES. In Remark 5, we give a possible data complexity for a revised algorithm that successfully distinguishes 5-round AES from a random permutation.

Remark 4. From Lemma 3 we can find the success probability of the distinguisher for the 5-round AES described in [RBH17] by taking $x = 2^{13.4}$, $y = 2^{11.4}$ and $t = 2$. For these values, the success probability of Algorithm 2 is approximately 0.5.

Remark 5. When $x = 2^{13.4}$, $y = 2^{14.55}$ and $t = 2$ the success probability of Algorithm 2 is 0.55. In this case the data complexity is $2 \times 2 \times 2^{13.4} \times 2^{14.55} = 2^{29.95}$. The time complexity of the attack is $2^{29.95}$ memory accesses and $2^{27.95}$ XOR operations.

Similarly, when $x = 2^{13.4}$, $y = 2^{15.25}$ and $t = 2$, the success probability of Algorithm 2 is 0.81. In this case the data complexity is $2 \times 2 \times 2^{13.4} \times 2^{15.25} = 2^{30.65}$. The time complexity of the attack is $2^{30.65}$ memory accesses and $2^{28.65}$ XOR operations.

Table 4 lists the theoretical and experimental success probabilities of the yoyo distinguisher for 5-round AES. It can be observed in the table that when the value of y is increasing (the value of x is fixed) the success probability converges around 0.8. Next, we discuss the possible reason behind this convergence (Remark 6) and the influence of the value of x (Remark 7).

Remark 6. We consider a scenario where the adversary has access to a total of x chosen plaintext pairs, and for each chosen plaintext pair, they can adaptively choose y plaintexts/ciphertexts. For the 5-round case when $t = 2$, we know that the probability of getting a `RightPair` is $\binom{4}{2}(2^{-8})^2 = \frac{1}{2^{13.4}}$. If we choose $x = 2^{13.4}$, then we get a `RightPair` with probability $1 - (1 - \frac{1}{2^{13.4}})^{2^{13.4}} = 0.63$. In the case of a random permutation, all the chosen pairs are `WrongPairs`. Now when we increase y, the probability of identifying a `WrongPair` increases and tends to 1. In the case of 5-round AES, the success probability is 0.63, and in the case of random permutation, the success probability tends to 1. Therefore, the overall probability tends to $\frac{1+0.63}{2} = 0.815$. Now for the case of 6-round AES, the probability of getting a `RightPair` is $\binom{4}{2}(2^{-32})^2 = \frac{1}{2^{61.41}}$. If we choose $x = 2^{61.41}$, then we get a `RightPair` with probability $1 - (1 - \frac{1}{2^{61.41}})^{2^{61.41}} = 0.63$. Similar to the above case, in this case, the success probability also tends to 0.815 as y increases.

Remark 7. From Remark 6, we see that if we increase x then to eliminate all `WrongPairs` we need to increase y also. Also, it is clear that if we increase x then the probability of the presence of `RightPair` will increase, and hence the probability of identifying AES increases. So by increasing x, we may increase the success probability but we also have to increase data complexity. For example, with $(x, y) = (2^{14.3}, 2^{15})$, the estimated (practical) success probability is 0.8684(0.865), while with $(2^{13.4}, 2^{15})$, it is 0.7918(0.775). So we choose x sufficiently large and fix it so that the algorithm has a high success probability.

4.2 Success Probability of the Yoyo Distinguisher on 6-round AES

Now, we proceed to compute the success probability of Algorithm 3, which serves as a distinguisher for 6-round AES. Given that this algorithm relies on the values of x, y, and t, we denote it as $\mathcal{D}^{x,y,t}_{AES_6}$. We use the notations $p^{x,y,t}_{AES_6}$ and $p^{x,y,t}_{\mathcal{RP}_6}$ to represent the probabilities that the distinguisher $\mathcal{D}^{x,y,t}_{AES_6}$ correctly identifies 6-round AES and a random permutation, respectively.

To find the above two probabilities, we first introduce some lemmas.

Lemma 4. *Consider* $s \xleftarrow{\$} \mathbb{F}^4_{2^{32}}$. *Then*

$$P[wt(\nu(s)) = k] = \binom{4}{k}(q^{-4})^k(1 - q^{-4})^{(4-k)},$$

where $q = 2^8$.

Proof. Let $P[wt(\nu(s)) = k]$ denote the probability that the state s has k inactive columns. Since k columns from 4 columns can be chosen in $\binom{4}{k}$ ways, the probability of the state s

with exactly k inactive columns is

$$P[wt(\nu(s)) = k] = \binom{4}{k}(q^{-4})^k(1-q^{-4})^{(4-k)}.$$

□

Lemma 5. *Consider $p^0, p^1 \xleftarrow{\$} \mathbb{F}_{2^8}^{4\times 4}$ with $p^0 \neq p^1$. Let B_i be the event that $wt(\nu(L \circ S(p^0) \oplus L \circ S(p^1))) = i$, where $i \in \{0,1,2,3\}$. Then $P(B_i) = \binom{4}{i}(q^{-4})^i(1-q^{-4})^{4-i}$, where $q = 2^8$.*

Proof. Given that p^0, p^1 are chosen randomly, we have

$$P(B_i) = P[wt(\nu(L \circ S(p^0) \oplus L \circ S(p^1))) = i] = \binom{4}{i}(q^{-4})^i(1-q^{-4})^{4-i}.$$

□

Lemma 6. *[RBH17] Let p^0, p^1 be a random pair such that $wt(\nu(L \circ S(p^0) \oplus L \circ S(p^1))) = i$. Then $p^0 \oplus p^1$ contains at most $3-i$ inactive columns and $c^0 \oplus c^1$ contains at most $3-i$ inactive columns where $c^0 = S \circ L \circ S \circ L \circ S(p^0)$ and $c^1 = S \circ L \circ S \circ L \circ S(p^1)$.*

Proof. It is given that $wt(\nu(L \circ S(p^0) \oplus L \circ S(p^1))) = i$. Then using Proposition 4, we can say that $wt(\nu(c^0 \oplus c^1)) \leq 3-i$. Again it is clear that $wt(\nu(S \circ L \circ S(p^0) \oplus S \circ L \circ S(p^1))) = i$ as $wt(\nu(L \circ S(p^0) \oplus L \circ S(p^1))) = i$. Then using Proposition 4 we can say that $wt(\nu(p^0 \oplus p^1)) \leq 3-i$. Therefore $(p^0 \oplus p^1)$ and $(c^0 \oplus c^1)$ contain at most $3-i$ inactive columns. □

Now we find the success probability of Algorithm 3 which distinguishes 6-round AES from a random permutation. For this purpose, we define a set

$$\mathcal{X}_i = \{\Delta p^{i,1}, \Delta c^{i,1}, \Delta p^{i,2}, \Delta c^{i,2}, \ldots, \Delta p^{i,y}, \Delta c^{i,y}\}.$$

The event $\mathcal{A}^t_{\mathcal{X}_i}$ denotes that $wt(\nu(\Delta p^{i,j}))$ and $wt(\nu(\Delta c^{i,j})) \in [0, 3-t]$, for all $1 \leq j \leq y$. Let $\omega_{p,i,j}$, $\omega_{c,i,j}$ and $\omega'_{i,j}$ denote $wt(\nu(\Delta p^{i,j}))$, $wt(\nu(\Delta c^{i,j}))$ and $wt(\nu(L \circ S(p^{i,2j-1}) \oplus L \circ S(p^{i,2j})))$ respectively. Note that due to Propositions 1 and 4, for a fixed i, $\omega'_{i,1} = \omega'_{i,2} = \cdots = \omega'_{i,y}$.

Now the distinguisher $\mathcal{D}^{x,y,t}_{AES_6}$ returns 6-round AES if there exists $i \in [1,x]$ such that the event $\mathcal{A}^t_{\mathcal{X}_i}$ occurs. Therefore the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_6}$ returns 6-round AES is $P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i})$. First, we find the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_6}$ can correctly identify 6-round AES, i.e. $p^{x,y,t}_{AES_6}$.

Probability of the event $(\mathcal{A}^t_{\mathcal{X}_i})$ when the encryption function (oracle) is 6-round AES:

Let $\mathcal{X}_i = \{\Delta p^{i,1}, \Delta c^{i,1}, \Delta p^{i,2}, \Delta c^{i,2}, \ldots, \Delta p^{i,y}, \Delta c^{i,y}\}$. Then the event $(\mathcal{A}^t_{\mathcal{X}_i})^c$ occurs if $\omega_{p,i,j} \in [4-t, 3]$ or $\omega_{c,i,j} \in [4-t, 3]$ for some $j \in [1, y]$. This is dependent on the value of $\omega'_{i,1} = \omega'_{i,2} = \cdots = \omega'_{i,y} = m$, where $m \in \{0,1,2,3\}$.

Case 1: $\mathbf{m \geq t}$. In this case, according to Lemma 6 we know that $\omega_{p,i,j} \notin [4-t, 3]$ and $\omega_{c,i,j} \notin [4-t, 3]$ for every $j \in [1, y]$. Then the probability that the event $(\mathcal{A}^t_{\mathcal{X}_i})^c$ occurs is '0'.

Case 2: $\mathbf{m < t}$. Then from Lemma 6, we know that $\omega_{p,i,j} \in [0, 3-m]$ and $\omega_{c,i,j} \in [0, 3-m]$ for every $j \in [1, y]$. So the event $(\mathcal{A}^t_{\mathcal{X}_i})^c$ occurs if there exists $j \in [1, y]$ such that $\omega_{p,i,j} \in [4-t, 3-m]$ or $\omega_{c,i,j} \in [4-t, 3-m]$.

- Using Lemma 4

  - The probability that for a fixed j, $\omega_{p,i,j} \in [4-t, 3-m]$ is $\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
  - The probability that, for a fixed j, $\omega_{c,i,j} \in [4-t, 3-m]$ is $\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
-
  - The probability that, for a fixed j, $\omega_{p,i,j} \notin [4-t, 3-m]$ is $1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
  - The probability that, for a fixed j, $\omega_{c,i,j} \notin [4-t, 3-m]$ is $1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
- The probability that, for a fixed j, $\omega_{p,i,j} \notin [4-t, 3-m]$ and $\omega_{c,i,j} \notin [4-t, 3-m]$ is $(1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^2$.
- The probability that the event $(\mathcal{A}^t_{\mathcal{X}_i})$ occur is $(1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^{2y}$.

Therefore from Case 1 and Case 2, we have

$$\begin{aligned}
P(\mathcal{A}^t_{\mathcal{X}_i}) &= \sum_{m\in[0,3]} P(\mathcal{A}^t_{\mathcal{X}_i}/B_m)P(B_m)(\text{where } B_m \text{ is the event that } \omega'_{i,1}=m)\\
&= \sum_{\substack{m<t\\ m\in[0,3]}} P(\mathcal{A}^t_{\mathcal{X}_i}/B_m)P(B_m) + \sum_{\substack{m\geq t\\ m\in[0,3]}} P(\mathcal{A}^t_{\mathcal{X}_i}/B_m)P(B_m)\\
&= \sum_{\substack{m<t\\ m\in[0,3]}} P(\mathcal{A}^t_{\mathcal{X}_i}/B_m)P(B_m) + \sum_{\substack{m\geq t\\ m\in[0,3]}} P(B_m).\\
&= \sum_{\substack{m<t\\ m\in[0,3]}} (1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^{2y}\times\mu_m + \sum_{\substack{m\geq t\\ m\in[0,3]}} \mu_m
\end{aligned}$$

where we take $\mu_m = P(B_m) = \binom{4}{m}(q^{-4})^m(1-q^{-4})^{4-m}$ from Lemma 5.

Probability of the event $\bigcup_{i=1}^x \mathcal{A}^t_{\mathcal{X}_i}$ when the encryption function (oracle) is 6-round AES

From the calculation of Equation (2) we get

$$\begin{aligned}
p^{x,y,t}_{AES_6} &= P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i})\\
&= 1-(1-P(\mathcal{A}^t_{\mathcal{X}_i}))^x\\
&= 1-(1-(\sum_{\substack{m<t\\ m\in[0,3]}} (1-\sum_{r\in[4-t,3-m]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^{2y}\times\mu_m + \sum_{\substack{m\geq t\\ m\in[0,3]}} \mu_m))^x.
\end{aligned}$$

Now we calculate the probability that the distinguisher $\mathcal{D}^{x,y,t}_{AES_6}$ can correctly identify random permutation, i.e. $p^{x,y,t}_{\mathcal{RP}_6}$. The probability that distinguisher $\mathcal{D}^{x,y,t}_{AES_6}$ returns random permutation is $1-P(\bigcup_{i=1}^x \mathcal{A}^t_{\mathcal{X}_i})$.

Probability of the event $(\mathcal{A}^t_{\mathcal{X}_i})$ when the encryption function (oracle) is a random permutation:

- Let $\mathcal{X}_i = \{\Delta p^{i,1}, \Delta c^{i,1}, \Delta p^{i,2}, \Delta c^{i,2}, \ldots, \Delta p^{i,y}, \Delta c^{i,y}\}$.

- Then the event $(\mathcal{A}^t_{\mathcal{X}_i})^c$ occurs for a fixed $j \in [1, y]$, if $\omega_{p,i,j} \in [4-t, 3]$ or $\omega_{c,i,j} \in [4-t, 3]$.
- Using Lemma 4
  - The probability that, for a fixed j, $\omega_{p,i,j} \in [4-t, 3]$ is $\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
  - The probability that, for a fixed j, $\omega_{c,i,j} \in [4-t, 3]$ is $\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
-
  - The probability that, for a fixed j, $\omega_{p,i,j} \notin [4-t, 3]$ is $1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
  - The probability that, for a fixed j, $\omega_{c,i,j} \notin [4-t, 3]$ is $1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}$.
- The probability that, for a fixed j, $\omega_{p,i,j} \notin [4-t, 3]$ and $\omega_{c,i,j} \notin [4-t, 3]$ is $(1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^2$.
- The probability that the event $(\mathcal{A}^t_{\mathcal{X}_i})$ occurs is $(1-\sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^{2y}$.

Probability of the event $\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i}$ when the encryption function (oracle) is a random permutation:
From the calculation of Equation 2, we get

$$P(\bigcup_{i=1}^{x} \mathcal{A}^t_{\mathcal{X}_i}) = 1 - (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x$$

Therefore

$$\begin{aligned} p^{x,y,t}_{\mathcal{RP}_6} &= 1 - (1 - (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x) \\ &= (1 - P(\mathcal{A}^t_{\mathcal{X}_i}))^x \\ &= (1 - (1 - \sum_{r\in[4-t,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)})^{2y})^x. \end{aligned}$$

Similar to 5-round distinguisher, the success probability of Algorithm 3 is

$$\frac{p^{x,y,t}_{AES_6} + p^{x,y,t}_{\mathcal{RP}_6}}{2}.$$

In Algorithm 3, there are a total of xy iterations. In each iteration, we perform one encryption and one decryption, resulting in a total of $2(2xy) = 4xy$ operations (as each iteration involves encrypting a pair of plaintexts and ciphertexts). So the data complexity is $4xy$. Within each iteration, we access a total of 4 states (comprising two ciphertexts and two plaintexts) and perform XOR operations on two plaintexts and two ciphertexts. This results in a time complexity of $4xy$ for memory accesses and $2xy$ for XOR operations.

Based on our probability calculation in Remark 8, we show that the algorithm described in [RBH17] is not able to distinguish 6-round AES. In Remark 9, we establish that our method is unable to distinguish 6-round AES with a data complexity lower than the full codebook.

Remark 8. From Lemma 3, the success probability of the distinguisher for the 6-round AES described in [RBH17] can be computed by taking $x = 2^{61.4}$, $y = 2^{60.4}$ and $t = 2$. For these values, the success probability of Algorithm 2 becomes approximately 0.5.

Remark 9. When $x = 2^{61.4}$, $y = 2^{65.75}$ and $t = 2$, the success probability of Algorithm 3 is 0.50004. In this case the data complexity is given by $2 \times 2 \times 2^{61.4} \times 2^{65.75} = 2^{129.15}$ (which exceeds the exhaustive search). The time complexity of the attack is $2^{129.15}$ memory accesses and $2^{128.15}$ XOR operations.

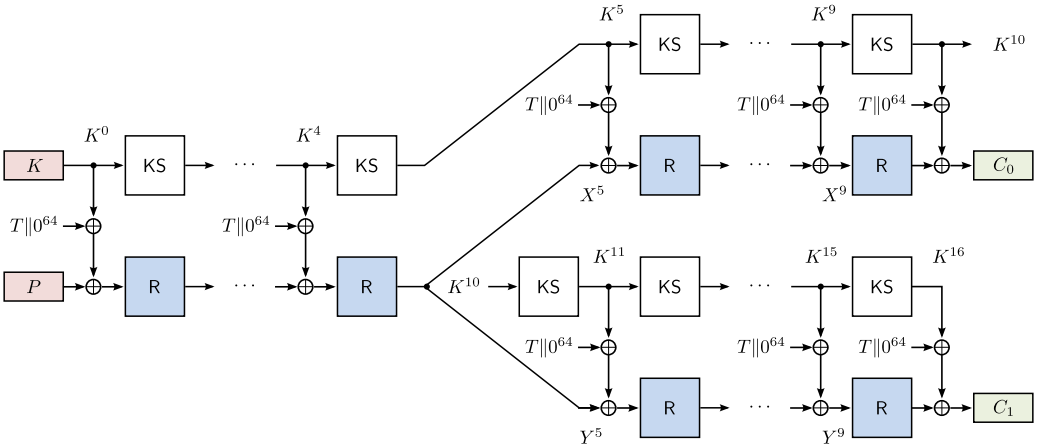

Figure 5: Construction of `ForkAES` [BBJ+19]. Here, **R** and **KS** denote the AES-128 round function and round of a key schedule function, respectively.

5 Impact on other Yoyo Distinguishers

The obtained results can be generalized and it can be used to determine the validity of other existing yoyo distinguishers. Here, we discuss the impact of our result on the impossible yoyo distinguisher of `ForkAES` proposed in [BBJ+19] and yoyo key recovery attack on 5-round AES [RBH17].

5.1 Impossible Yoyo Distinguisher on `ForkAES-*-4-4`

The design of `ForkAES` [ARVV18] is based on the design of KIASU-BC [JNP14] and it follows the forkcipher design strategy proposed at Asiacrypt 2019 [ALP+19]. The construction of `ForkAES` is depicted in Figure 5. First of all, a 128-bit state, P, is encrypted using five rounds of AES. Then the state is forked into two equivalent states and both states are again encrypted using 5-round `AES` to output C_0 and C_1. In `ForkAES`, in addition to a 128-bit key K, a 64-bit tweak T is also used. Note that, the designers of `ForkAES` introduced the notion of *reconstruction query* in addition to normal encryption queries. In the reconstruction query, C_0 (or C_1) can be queried to obtain C_1 (or C_0). In [BBJ+19], the notation `ForkAES-*-x-y` is introduced which denotes the variant of `ForkAES` with x and y number of rounds in its forked branches. The yoyo distinguishing attack on `ForkAES`-*-4-4 is based on the reconstruction queries. For more details on the construction of `ForkAES`, refer to [ARVV18].

Attack Overview. In [BBJ+19], for `ForkAES-*-4-4` the underlying structure involved in the computation of reconstruction queries can be written as $S_1 \circ L_1 \circ S_2 \circ L_2 \circ S_3$ where S_1, S_3 corresponds to super-sbox, S_2 corresponds to megasbox and L_1, L_2 corresponds to MixColumn operation (linear layer)(depicted in Figure 6).

In [BBJ+19], Banik et al. mounted an impossible-differential yoyo distinguishing attack on `ForkAES`-*-4-4. This attack is quite similar to the one which is mounted on a 6-round AES. The only difference between these two attacks is in the definition of the intermediate non-linear layer. For 6-round AES, the intermediate non-linear layer is a super-sbox whereas for `ForkAES`-*-4-4 it is a megasbox (as depicted in Figure 6). However, this difference has no impact on the attack complexities in these two attacks as the primary notions behind devising these two distinguishers are similar.

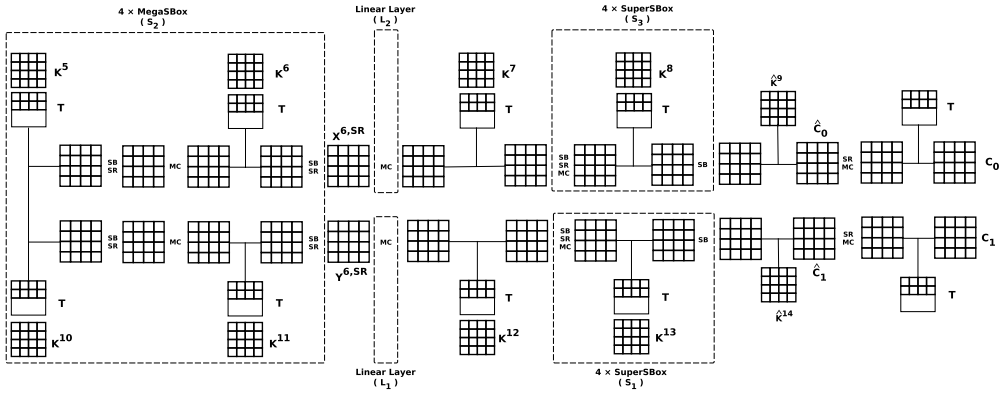

Figure 6: Underlying construction of $S_1 \circ L_1 \circ S_2 \circ L_2 \circ S_3$ in `ForkAES-*-4-4` [BBJ+19]

Invalidation of the Distinguisher. The distinguisher outlined in Algorithm 3 is quite similar to the one proposed to distinguish `ForkAES`-*-4-4. In particular, the attack parameters ((x, y, t) as in Algorithm 3) also remain the same. Note that, the computation of success probability depends only on (x, y, t), and thus the success probability of both of these distinguishers is the same. This invalidates the impossible-differential yoyo distinguisher of `ForkAES`-*-4-4.

5.2 Yoyo Key Recovery Attack on 5-round AES

Here, we discuss the impact of our results on yoyo key recovery attack on 5-round AES. First, we re-introduce some notations from [RBH17]. Let p^i represent a state of AES and $p^i = (p^i_0, p^i_1, p^i_2, p^i_3)$ written word wise. Let k be the secret key and $k_{i,j}$ be the byte at position (i, j) of the key matrix.

In the 5-round key recovery attack, two pairs of plaintexts p^0 and p^1 are chosen whose first words are given by $p^0_0 = (0, i, 0, 0)$ and $p^1_0 = (z, z \oplus i, 0, 0)$, respectively, where z is a known value. The other words are the same. Then it is shown that at least for $i \in \{k_{0,0} \oplus k_{0,1}, z \oplus k_{0,0} \oplus k_{0,1}\}$ the third byte of the first word of $MC \circ SB \circ ARK(p^0) \oplus MC \circ SB \circ ARK(p^1)$ is 0.

Now, plaintext pairs p^0 and p^1 are generated for each i. The attack consists of encrypting those plaintexts and getting the ciphertexts c^0 and c^1. Then, five new ciphertext pairs are generated using ρ function i.e., $(c'^0, c'^1) = (\rho^v(c^0, c^1), \rho^v(c^1, c^0))$. Now those ciphertext pairs are decrypted and the corresponding plaintext pairs (p'^0,p'^1) are stored. Then the third byte of the first word of $MC \circ SB \circ ARK(p'^0) \oplus MC \circ SB \circ ARK(p'^1)$ is 0 for all five pairs (p'^0, p'^1).

The first word of the secret key is guessed. It is known that $k_{0,0} \oplus k_{0,1} \in \{i, i \oplus z\}$. So there are $2 \cdot 2^{24}$ possibilities. But 2^{24} possibilities are chosen because it is sufficient to test $k_{0,1} = k_{0,0} \oplus i$ and there is no need to test $k_{0,1} = k_{0,0} \oplus i \oplus z$ as i will run over all possible 2^8 values. After guessing the first word of the secret key, all five stored pairs are checked to see if they satisfy the condition that the third byte of the first word of $MC \circ SB \circ ARK(p'^0) \oplus MC \circ SB \circ ARK(p'^1)$ is 0 or not. If the condition is satisfied, it is expected that the first word of the secret key is correctly guessed.

This occurs because if the third byte of the first word of $MC \circ SB \circ ARK(p^0) \oplus MC \circ SB \circ ARK(p^1)$ is zero and other words are also inactive, so after the SR operation the third word of the difference would be zero. Now using Proposition 1, we assert that the third word of $SR \circ MC \circ SB \circ ARK(p'^0) \oplus SR \circ MC \circ SB \circ ARK(p'^1)$ is zero for all five pairs (p'^0, p'^1). Therefore, the third byte of first word of $MC \circ SB \circ ARK(p'^0) \oplus MC \circ SB \circ ARK(p'^1)$ is

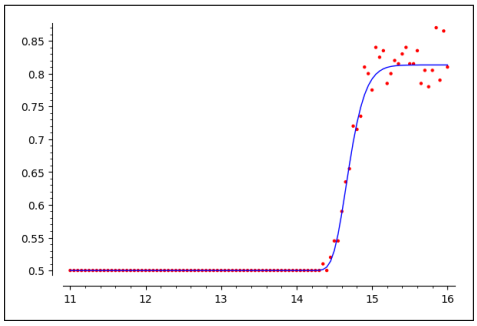

Figure 7: Success probability of the 5-round distinguisher (Algorithm 2) as a function of y. The x-axis represents the value of $\log_2 y$ and the y-axis represents the success probability of Algorithm 2 when t = 2. The blue line represents the theoretical probability while the red dot represents the experimental probability.

inactive.

From the above discussion, it is evident that the yoyo key recovery attack on 5-round AES is based on deterministic properties. As the results (presented in this paper) have no impact on the deterministic distinguisher, the yoyo key recovery attack on 5-round AES works as expected.

6 Theoretical and Experimental Results

Based on the above analysis, in this section, we present both experimental and theoretical success probabilities of the yoyo distinguisher for 5-round AES. All the experiments are conducted in the C language [2]. Additionally, we also provide the theoretical success probability for 6-round AES for different values of x and y. In our experiments, we generate a new key for each experiment using the drand48() function, which is a pseudorandom number generator. We use two methods to generate random permutations. In one program, we use drand48() to generate a random permutation. In another program, we use 20-round AES encryption to generate a random permutation.

From the result of Table 1, we say that when $x = 2^{13.4}$ and $y = 2^{11.4}$ the 5-round AES distinguisher cannot distinguish AES from a random permutation. From the analysis in Section 4, it is clear that the value of y should be greater than $2^{11.4}$. So we increase the value of y by $2^{0.05}$ and calculate the experimental and theoretical success probabilities. It is observed that when the value of $y \leq 2^{14.20}$, the success probability of Algorithm 2 is 0.5. After that, success probability of Algorithm 2 increases. So the value of y should be greater than $2^{14.20}$ to distinguish 5-round AES. Therefore the data complexity of Algorithm 2 when $x = 2^{13.4}$ and $y = 2^{14.20}$ is $2 \times 2 \times 2^{13.4} \times 2^{14.20} = 2^{29.6}$. The time complexity of the attack is $2^{29.6}$ memory accesses and $2^{13.4} \times 2^{14.20} = 2^{27.6}$ XOR operations. In Table 4, we list the experimental and theoretical success probabilities of Algorithm 2 for some values of y and in Figure 7 we give a comparison between the theoretical and experimental results.

From Table 4, we can obtain the experimental success probabilities of $p^{x,y,2}_{AES_5}$ and $p^{x,y,2}_{\mathcal{RP}_5}$. When $n_1 = 0$, we compute the probability $p^{x,y,2}_{\mathcal{RP}_5}$ by dividing the cell value under the column labeled "Found as Random" (when the black box cipher is either "RANDOM (AES20)" or "RANDOM(drand48)") by $n_2 = 100$. When $n_2 = 0$, the probability of $p^{x,y,2}_{AES_5}$ can be calculated by dividing the cell value under the column labeled “Found as AES”

[2]The codes are available at https://github.com/sandipkumarmondal/Revisiting-Yoyo-Tricks-on-AES

Table 2: Some theoretical and experimental results of Algorithm 2 for larger values of x when $t = 2$.

Value of x	Value of y	Data Complexity	Theoretical Success Probability	Experimental Success Probability
2^{15}	$2^{14.4}$	$2^{31.4}$	0.5	0.5
	$2^{14.6}$	$2^{31.6}$	0.51	0.51
	$2^{14.8}$	$2^{31.8}$	0.6739	0.68
	2^{15}	2^{32}	0.8832	0.89
	$2^{15.2}$	$2^{32.2}$	0.9578	0.955
	$2^{15.4}$	$2^{32.4}$	0.9725	0.965
	$2^{15.6}$	$2^{32.6}$	0.9745	0.945
	$2^{15.8}$	$2^{32.8}$	0.9747	0.975
$2^{15.3}$	$2^{14.4}$	$2^{31.7}$	0.5	0.5
	$2^{14.6}$	$2^{31.9}$	0.5042	0.51
	$2^{14.8}$	$2^{32.1}$	0.6415	0.67
	2^{15}	$2^{32.3}$	0.8744	0.86
	$2^{15.2}$	$2^{32.5}$	0.966	0.97
	$2^{15.4}$	$2^{32.3}$	0.9845	0.995
	$2^{15.6}$	$2^{32.9}$	0.987	0.995
	$2^{15.8}$	$2^{33.1}$	0.9873	0.98
$2^{15.6}$	$2^{15.37}$	$2^{32.97}$	0.99	0.995

(when black box cipher is "AES") by $n_1 = 100$. The graphical representation for these cases is shown in Figure 8.

Note that, we can achieve a success probability close to 100% by choosing suitable values of x and y. In that case, we have to choose x sufficiently large such that there always presents a `RightPair` in the case of AES. For that x, we have to choose y sufficiently large so that all the `WrongPair` is eliminated for the case of random permutation. We know that the probability of getting a `RightPair` is $\frac{1}{2^{13.4}}$. If we choose x such that $1-(1-\frac{1}{2^{13.4}})^x \geq 0.99$ i.e., $x \geq 2^{15.6032}$ then we get $p^{x,y,2}_{AES_5} \geq 0.99$. Now, for this x, if we choose $y \geq 2^{15.37}$ then we get $p^{x,y,2}_{\mathcal{RP}_5} \geq 0.99$. So the overall success probability is greater than or equal to 0.99. In Table 2, we give some theoretical and experimental results for larger x. Table 2 shows that for larger x, we can achieve a success probability close to 100%.

Based on the results presented in Table 5, it can be concluded that a 6-round AES distinguisher is unable to differentiate 6-round AES from a random permutation when the values of x and y are set to $x = 2^{61.4}$ and $y = 2^{60.4}$. However, according to the analysis in Section 4, it is evident that the value of y should be greater than $2^{60.4}$. Therefore, we increased the value of y by $2^{0.05}$ and calculated the theoretical success probability. It is observed that as long as y remains less than or equal to $2^{65.70}$, the success probability of Algorithm 3 remains at 0.5. Beyond this point, the success probability starts to increase. To provide more detailed insights, we adjusted the value of y by increments of $2^{0.01}$ at the point where the theoretical success probability changed rapidly. Our final observation indicates that for values of y up to $2^{65.74}$, the success probability of Algorithm 3 remains at 0.5. Beyond this threshold, the success probability starts to increase again. Therefore, it can be inferred that the value of y should be greater than or equal to $2^{65.75}$ in order

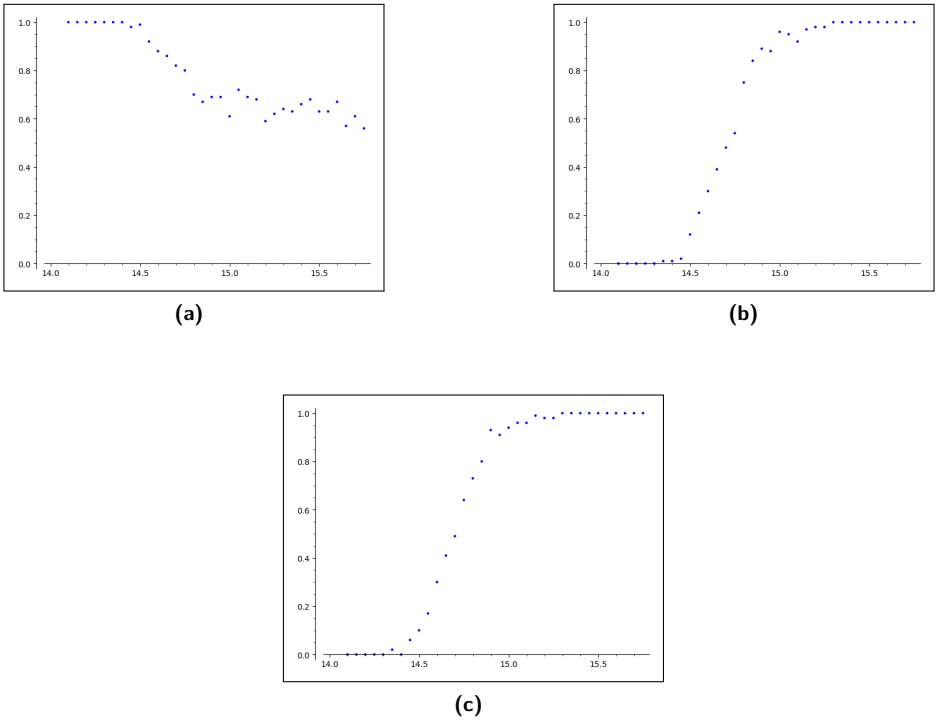

Figure 8: Practical success probability of the 5-round distinguisher (Algorithm 2) as a function of y. The x-axis represents the value of $\log_2 y$ and the y-axis represents the practical success probability of Algorithm 2 when $t = 2$. In the figures, the practical success probability of Algorithm 2 are shown when blackbox cipher is 5-round AES, random(drand48()) and random(AES20) respectively.

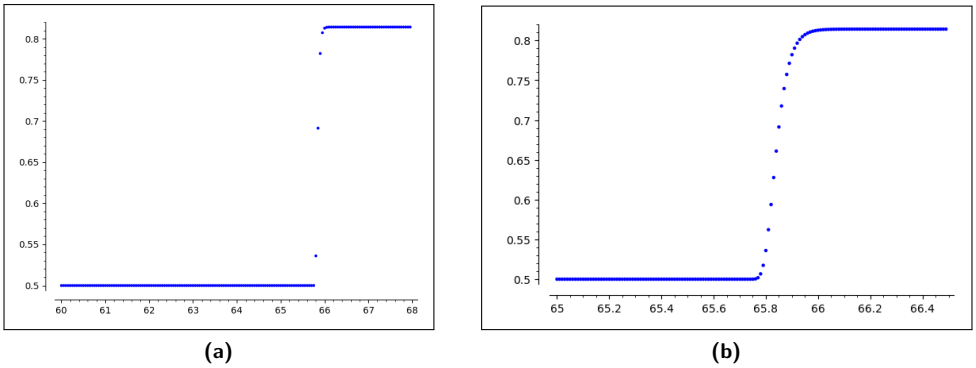

Figure 9: Success probability of the 6-round distinguisher (Algorithm 3) as a function of y. The x-axis represents the value of $\log_2 y$ and the y-axis represents the theoretical success probability of Algorithm 3 when t = 2. In the first figure, the points are spaced 0.05 units apart, while in the second figure, the points are spaced 0.01 units apart.

Table 3: Success probability and data complexity of Algorithm 2 and 3 for different values of x and y when $t = 2$. Here, MAs refer to memory accesses.

Round	Value of x	Value of y	Data Complexity	Time Complexity	Success Probability	Paper
5	$2^{13.4}$	$2^{11.4}$	$2^{26.8}$	$2^{24.8}$ XOR + $2^{26.8}$ MAs	0.5	[RBH17]
	$2^{13.4}$	$2^{15.25}$	$2^{30.65}$	$2^{28.65}$ XOR + $2^{30.65}$ MAs	0.81	our
	$2^{15.60}$	$2^{15.37}$	$2^{32.97}$	$2^{30.97}$ XOR + $2^{32.97}$ MAs	0.99	
6	$2^{61.4}$	$2^{60.4}$	$2^{123.8}$	$2^{122.8}$ XOR + $2^{123.8}$ MAs	0.5	[RBH17]
	$2^{61.4}$	$2^{65.76}$	$2^{129.15}$	$2^{128.15}$ XOR + $2^{129.15}$ MAs	0.50004	our

to distinguish 6-round AES effectively. In Table 5, we provide the theoretical success probability of Algorithm 3 for various values of y, and in Figure 9, we present a graphical representation of these theoretical results. Consequently, when setting $x = 2^{61.4}$ and $y = 2^{65.75}$, the data complexity of Algorithm 3 amounts to $2 \times 2 \times 2^{61.4} \times 2^{65.75} = 2^{129.15}$. The time complexity of the attack is $2^{129.15}$ memory accesses and $2^{128.15}$ XOR operations. In Table 3 we give the success probability and data complexity of Algorithm 2 and 3 for different values of x and y.

Now, we calculate the average complexity of 5-round and 6-round distinguishers of AES. For the average complexity, we first have to ensure that all the `WrongPairs` are identified in both the cases i.e., for AES and random permutation. Now we see that theoretically and experimentally (for 5 rounds) if we choose $y \geq 2^{15.70}$ (refer to Table 4), then all the `WrongPairs` can be identified. Now in the case of 5-round, when $t = 2$ we know that a random pair of states is a `WrongPair` with probability $p = 1 - (1 - \sum_{r\in[2,3]} \binom{4}{r}(q^{-1})^r(1 - q^{-1})^{(4-r)})^4 = 2^{-11.42}$. So, it is expected that for $y = 2^{11.42}$, a `WrongPair` can be detected. So the average complexity is $x \times \frac{1}{p} \times 2 \times 2 = 2^{13.4} \times 2^{11.42} \times 2^2 = 2^{26.82}$. Note that, only when $y \geq 2^{15.70}$ all the `WrongPairs` can be detected with a probability close to one.

Now, in the case of 6-round AES distinguisher, for $t = 2$ a random pair is a `WrongPair` with probability $p = 1-(1-\sum_{r\in[2,3]} \binom{4}{r}(q^{-4})^r(1-q^{-4})^{(4-r)}) = \frac{1}{2^{61.42}}$. So, it is expected that for $y = 2^{61.42}$, a `WrongPair` can be detected. In this case, for identifying all `WrongPairs` y should be greater than or equal to $2^{66.13}$ (refer to Table 5). So the average complexity is $x \times \frac{1}{p} \times 2 = 2^{61.4} \times 2^{61.42} \times 2 = 2^{123.82}$.

7 Conclusion

In this work, we revisit the work of Rønjom et al. and analyze the reported 5-round and 6-round AES distinguishers. We observed that the success probabilities of both these distinguishers are quite low. Based on this, we propose a revised distinguisher for

5-round AES by increasing the data complexity. However, the 6-round AES can not be modified in order to gain significant success probability, the data complexity of the attack exceeds the full codebook complexity. Similarly, our investigation also invalidates the impossible-differential yoyo distinguisher on `ForkAES-*-4-4`.

We would like to emphasize the significance of the success probability in cryptographic attack algorithms. It is crucial to establish the validity of these attacks by demonstrating a substantial success probability while maintaining a complexity lower than that of an exhaustive search.

Acknowledgments

We thank Virginie Lallemand and the anonymous reviewers of ToSC for their insightful and valuable comments, which helped to improve the technical as well as the editorial quality of our manuscript. Sandip Kumar Mondal is thankful to the University Grants Commission(UGC) for providing a research fellowship. Research of Dr. Avishek Adhikari is partially supported by the DST-SERB project MATRICS, vide sanction order: MTR/2019/001573, and DST-FIST project, Govt. of India, vide sanction order: SR/FST/MS-I/2019/41.

References

[ALP+19] Elena Andreeva, Virginie Lallemand, Antoon Purnal, Reza Reyhanitabar, Arnab Roy, and Damian Vizár. Forkcipher: A new primitive for authenticated encryption of very short messages. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part II*, volume 11922 of *Lecture Notes in Computer Science*, pages 153–182. Springer, 2019.

[ARVV18] Elena Andreeva, Reza Reyhanitabar, Kerem Varici, and Damian Vizár. Forking a blockcipher for authenticated encryption of very short messages. *IACR Cryptol. ePrint Arch.*, page 916, 2018.

[BBD+98] Eli Biham, Alex Biryukov, Orr Dunkelman, Eran Richardson, and Adi Shamir. Initial observations on skipjack: Cryptanalysis of skipjack-3xor. In Stafford E. Tavares and Henk Meijer, editors, *Selected Areas in Cryptography '98, SAC'98, Kingston, Ontario, Canada, August 17-18, 1998, Proceedings*, volume 1556 of *Lecture Notes in Computer Science*, pages 362–376. Springer, 1998.

[BBJ+19] Subhadeep Banik, Jannis Bossert, Amit Jana, Eik List, Stefan Lucks, Willi Meier, Mostafizar Rahman, Dhiman Saha, and Yu Sasaki. Cryptanalysis of ForkAES. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 43–63, Cham, 2019. Springer International Publishing.

[BDD+12] Charles Bouillaguet, Patrick Derbez, Orr Dunkelman, Pierre-Alain Fouque, Nathan Keller, and Vincent Rijmen. Low-data complexity attacks on AES. *IEEE Trans. Inf. Theory*, 58(11):7002–7017, 2012.

[Bir04] Alex Biryukov. The boomerang attack on 5 and 6-round reduced AES. In Hans Dobbertin, Vincent Rijmen, and Aleksandra Sowa, editors, *Advanced Encryption Standard - AES, 4th International Conference, AES 2004, Bonn, Germany, May 10-12, 2004, Revised Selected and Invited Papers*, volume 3373 of *Lecture Notes in Computer Science*, 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, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[BKR11] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.

[BLP16] Alex Biryukov, Gaëtan Leurent, and Léo Perrin. Cryptanalysis of feistel networks with secret round functions. In Orr Dunkelman and Liam Keliher, editors, *Selected Areas in Cryptography – SAC 2015*, pages 102–121, Cham, 2016. Springer International Publishing.

[DF16] Patrick Derbez and Pierre-Alain Fouque. Automatic search of meet-in-the-middle and impossible differential attacks. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 157–184. Springer, 2016.

[DKR97] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher square. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 1997.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[DR06] Joan Daemen and Vincent Rijmen. Understanding two-round differentials in AES. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings*, volume 4116 of *Lecture Notes in Computer Science*, pages 78–94. Springer, 2006.

[DR07] Joan Daemen and Vincent Rijmen. Plateau characteristics. *IET Inf. Secur.*, 1(1):11–17, 2007.

[Gra18] Lorenzo Grassi. Mixture differential cryptanalysis: a new approach to distinguishers and attacks on round-reduced AES. *IACR Trans. Symmetric Cryptol.*, 2018(2):133–160, 2018.

[GRR16] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. Subspace trail cryptanalysis and its applications to AES. *IACR Trans. Symmetric Cryptol.*, 2016(2):192–225, 2016.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and keys for block ciphers: The TWEAKEY framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung,*

Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.

[Knu94] Lars R. Knudsen. Truncated and higher order differentials. In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.

[LDKK08] Jiqiang Lu, Orr Dunkelman, Nathan Keller, and Jongsung Kim. New impossible differential attacks on aes. In Dipanwita Roy Chowdhury, Vincent Rijmen, and Abhijit Das, editors, *Progress in Cryptology - INDOCRYPT 2008*, pages 279–293, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[NIS] NIST. National Institute of Standards and Technology: Lightweight Cryptography (LWC) Standardization Project. https://csrc.nist.gov/Projects/lightweight-cryptography.

[oS77] National Bureau of Standards. Data encryption standard. federal information processing standards publication 46. January 1977.

[RBH17] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseth. Yoyo tricks with AES. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 217–243. Springer, 2017.

[SRP18] Dhiman Saha, Mostafizar Rahman, and Goutam Paul. New yoyo tricks with AES-based permutations. *IACR Trans. Symmetric Cryptol.*, 2018(4):102–127, 2018.

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

Table 4: Experimental and theoretical results for 5-round distinguisher when t=2 and $x = 2^{13.4}$. Here experimental success probability of Algorithm 2 is based on the results of random permutation AES20 rounds.

Number of Experiments	Blackbox Cipher	Value of y	Found as AES	Found as Random	Success Probability (Theoretical)	Overall Success Probability (Experimental)	Overall Success Probability (Theoretical)
100	AES	$2^{14.1}$	100	0	1.0	0.5	0.5
100	RANDOM (AES20)	$2^{14.1}$	100	0	0.0		
100	RANDOM (drand48)	$2^{14.1}$	100	0			
100	AES	$2^{14.15}$	100	0	1.0	0.5	0.5
100	RANDOM (AES20)	$2^{14.15}$	100	0	0.0		
100	RANDOM (drand48)	$2^{14.15}$	100	0			
100	AES	$2^{14.2}$	100	0	1.0	0.5	0.5
100	RANDOM (AES20)	$2^{14.2}$	100	0	0.0		
100	RANDOM (drand48)	$2^{14.2}$	100	0			
100	AES	$2^{14.25}$	100	0	1.0	0.5	0.5
100	RANDOM (AES20)	$2^{14.25}$	100	0	0.0001		
100	RANDOM (drand48)	$2^{14.25}$	100	0			
100	AES	$2^{14.3}$	100	0	0.9996	0.5	0.5003
100	RANDOM (AES20)	$2^{14.3}$	100	0	0.001		
100	RANDOM (drand48)	$2^{14.3}$	100	0			
100	AES	$2^{14.35}$	100	0	0.9983	0.51	0.5015
100	RANDOM (AES20)	$2^{14.35}$	98	2	0.0047		
100	RANDOM (drand48)	$2^{14.35}$	99	1			
100	AES	$2^{14.4}$	100	0	0.9938	0.5	0.5052
100	RANDOM (AES20)	$2^{14.4}$	100	0	0.0165		
100	RANDOM (drand48)	$2^{14.4}$	99	1			
100	AES	$2^{14.45}$	98	2	0.9833	0.52	0.514
100	RANDOM (AES20)	$2^{14.45}$	94	6	0.0446		
100	RANDOM (drand48)	$2^{14.45}$	98	2			
100	AES	$2^{14.5}$	99	1	0.9638	0.545	0.5304
100	RANDOM (AES20)	$2^{14.5}$	90	10	0.0971		
100	RANDOM (drand48)	$2^{14.5}$	88	12			
100	AES	$2^{14.55}$	92	8	0.9339	0.545	0.5555
100	RANDOM (AES20)	$2^{14.55}$	83	17	0.177		
100	RANDOM (drand48)	$2^{14.55}$	79	21			
100	AES	$2^{14.6}$	88	12	0.8954	0.59	0.5878
100	RANDOM (AES20)	$2^{14.6}$	70	30	0.2802		
100	RANDOM (drand48)	$2^{14.6}$	70	30			
100	AES	$2^{14.65}$	86	14	0.8519	0.635	0.6242
100	RANDOM (AES20)	$2^{14.65}$	59	41	0.3966		
100	RANDOM (drand48)	$2^{14.65}$	61	39			
100	AES	$2^{14.7}$	82	18	0.8078	0.655	0.6612
100	RANDOM (AES20)	$2^{14.7}$	51	49	0.5145		
100	RANDOM (drand48)	$2^{14.7}$	52	48			
100	AES	$2^{14.75}$	80	20	0.767	0.72	0.6954
100	RANDOM (AES20)	$2^{14.75}$	36	64	0.6237		
100	RANDOM (drand48)	$2^{14.75}$	46	54			
100	AES	$2^{14.8}$	70	30	0.7318	0.715	0.7249
100	RANDOM (AES20)	$2^{14.8}$	27	73	0.7179		
100	RANDOM (drand48)	$2^{14.8}$	25	75			

Number of Experiments	Blackbox Cipher	Value of y	Found as AES	Found as Random	Success Probability (Theoretical)	Overall Success Probability (Experimental)	Overall Success Probability (Theoretical)
100	AES	$2^{14.85}$	67	33	0.7031	0.735	0.7489
100	RANDOM (AES20)	$2^{14.85}$	20	80	0.7948		
100	RANDOM (drand48)	$2^{14.85}$	16	84			
100	AES	$2^{14.9}$	69	31	0.6807	0.81	0.7677
100	RANDOM (AES20)	$2^{14.9}$	7	93	0.8546		
100	RANDOM (drand48)	$2^{14.9}$	11	89			
100	AES	$2^{14.95}$	69	31	0.664	0.8	0.7817
100	RANDOM (AES20)	$2^{14.95}$	9	91	0.8993		
100	RANDOM (drand48)	$2^{14.95}$	12	88			
100	AES	$2^{15.0}$	61	39	0.6519	0.775	0.7918
100	RANDOM (AES20)	$2^{15.0}$	6	94	0.9318		
100	RANDOM (drand48)	$2^{15.0}$	4	96			
100	AES	$2^{15.05}$	72	28	0.6433	0.84	0.799
100	RANDOM (AES20)	$2^{15.05}$	4	96	0.9547		
100	RANDOM (drand48)	$2^{15.05}$	5	95			
100	AES	$2^{15.1}$	69	31	0.6374	0.825	0.8039
100	RANDOM (AES20)	$2^{15.1}$	4	96	0.9705		
100	RANDOM (drand48)	$2^{15.1}$	8	92			
100	AES	$2^{15.15}$	68	32	0.6335	0.835	0.8073
100	RANDOM (AES20)	$2^{15.15}$	1	99	0.9811		
100	RANDOM (drand48)	$2^{15.15}$	3	97			
100	AES	$2^{15.2}$	59	41	0.6308	0.785	0.8095
100	RANDOM (AES20)	$2^{15.2}$	2	98	0.9881		
100	RANDOM (drand48)	$2^{15.2}$	2	98			
100	AES	$2^{15.25}$	62	38	0.6291	0.8	0.8109
100	RANDOM (AES20)	$2^{15.25}$	2	98	0.9926		
100	RANDOM (drand48)	$2^{15.25}$	2	98			
100	AES	$2^{15.3}$	64	36	0.6281	0.82	0.8118
100	RANDOM (AES20)	$2^{15.3}$	0	100	0.9955		
100	RANDOM (drand48)	$2^{15.3}$	0	100			
100	AES	$2^{15.35}$	63	37	0.6274	0.815	0.8124
100	RANDOM (AES20)	$2^{15.35}$	0	100	0.9973		
100	RANDOM (drand48)	$2^{15.35}$	0	100			
100	AES	$2^{15.4}$	66	34	0.627	0.83	0.8127
100	RANDOM (AES20)	$2^{15.4}$	0	100	0.9984		
100	RANDOM (drand48)	$2^{15.4}$	0	100			
100	AES	$2^{15.45}$	68	32	0.6267	0.84	0.8129
100	RANDOM (AES20)	$2^{15.45}$	0	100	0.9991		
100	RANDOM (drand48)	$2^{15.45}$	0	100			
100	AES	$2^{15.5}$	63	37	0.6266	0.815	0.813
100	RANDOM (AES20)	$2^{15.5}$	0	100	0.9995		
100	RANDOM (drand48)	$2^{15.5}$	0	100			

Number of Experiments	Blackbox Cipher	Value of y	Found as AES	Found as Random	Success Probability (Theoretical)	Overall Success Probability (Experimental)	Overall Success Probability (Theoretical)
100	AES	$2^{15.55}$	63	37	0.6265	0.815	0.8131
100	RANDOM (AES20)	$2^{15.55}$	0	100	0.9997		
100	RANDOM (drand48)	$2^{15.55}$	0	100			
100	AES	$2^{15.6}$	67	33	0.6264	0.835	0.8131
100	RANDOM (AES20)	$2^{15.6}$	0	100	0.9999		
100	RANDOM (drand48)	$2^{15.6}$	0	100			
100	AES	$2^{15.65}$	57	43	0.6264	0.785	0.8132
100	RANDOM (AES20)	$2^{15.65}$	0	100	0.9999		
100	RANDOM (drand48)	$2^{15.65}$	0	100			
100	AES	$2^{15.7}$	61	39	0.6264	0.805	0.8132
100	RANDOM (AES20)	$2^{15.7}$	0	100	1.0		
100	RANDOM (drand48)	$2^{15.7}$	0	100			
100	AES	$2^{15.75}$	56	44	0.6264	0.78	0.8132
100	RANDOM (AES20)	$2^{15.75}$	0	100	1.0		
100	RANDOM (drand48)	$2^{15.75}$	0	100			

Table 5: Theoretical results for 6-round distinguishers when t=2 and $x = 2^{61.4}$

SL No	Value of y	Success Probability (When Oracle is AES)	Success Probability (When Oracle is Random Permutation)	Success Probability (Overall)	SL No	Value of y	Success Probability (When Oracle is AES)	Success Probability (When Oracle is Random Permutation)	Success Probability (Overall)
1	$2^{65.6}$	1.0	0.0	0.5	41	$2^{66.0}$	0.6299	0.9957	0.8128
2	$2^{65.61}$	1.0	0.0	0.5	42	$2^{66.01}$	0.6294	0.9969	0.8132
3	$2^{65.62}$	1.0	0.0	0.5	43	$2^{66.02}$	0.6291	0.9978	0.8134
4	$2^{65.63}$	1.0	0.0	0.5	44	$2^{66.03}$	0.6289	0.9984	0.8136
5	$2^{65.64}$	1.0	0.0	0.5	45	$2^{66.04}$	0.6287	0.9989	0.8138
6	$2^{65.65}$	1.0	0.0	0.5	46	$2^{66.05}$	0.6286	0.9992	0.8139
7	$2^{65.66}$	1.0	0.0	0.5	47	$2^{66.06}$	0.6285	0.9994	0.814
8	$2^{65.67}$	1.0	0.0	0.5	48	$2^{66.07}$	0.6284	0.9996	0.814
9	$2^{65.68}$	1.0	0.0	0.5	49	$2^{66.08}$	0.6284	0.9997	0.8141
10	$2^{65.69}$	1.0	0.0	0.5	50	$2^{66.09}$	0.6284	0.9998	0.8141
11	$2^{65.7}$	1.0	0.0	0.5	51	$2^{66.1}$	0.6283	0.9999	0.8141
12	$2^{65.71}$	1.0	0.0	0.5	52	$2^{66.11}$	0.6283	0.9999	0.8141
13	$2^{65.72}$	1.0	0.0	0.5	53	$2^{66.12}$	0.6283	0.9999	0.8141
14	$2^{65.73}$	1.0	0.0	0.5	54	$2^{66.13}$	0.6283	1.0	0.8141
15	$2^{65.74}$	1.0	0.0	0.5	55	$2^{66.14}$	0.6283	1.0	0.8141
16	$2^{65.75}$	1.0	0.0001	0.5	56	$2^{66.15}$	0.6283	1.0	0.8141
17	$2^{65.76}$	0.9996	0.0011	0.5004	57	$2^{66.16}$	0.6283	1.0	0.8141
18	$2^{65.77}$	0.9978	0.006	0.5019	58	$2^{66.17}$	0.6283	1.0	0.8141
19	$2^{65.78}$	0.9921	0.0213	0.5067	59	$2^{66.18}$	0.6283	1.0	0.8141
20	$2^{65.79}$	0.9794	0.0555	0.5174	60	$2^{66.19}$	0.6283	1.0	0.8141
21	$2^{65.8}$	0.9574	0.1147	0.536	61	$2^{66.2}$	0.6283	1.0	0.8141
22	$2^{65.81}$	0.9264	0.198	0.5622	62	$2^{66.21}$	0.6283	1.0	0.8141
23	$2^{65.82}$	0.889	0.2986	0.5938	63	$2^{66.22}$	0.6283	1.0	0.8141
24	$2^{65.83}$	0.8489	0.4065	0.6277	64	$2^{66.23}$	0.6283	1.0	0.8141
25	$2^{65.84}$	0.8096	0.5122	0.6609	65	$2^{66.24}$	0.6283	1.0	0.8141
26	$2^{65.85}$	0.7737	0.6088	0.6913	66	$2^{66.25}$	0.6283	1.0	0.8141
27	$2^{65.86}$	0.7425	0.6926	0.7176	67	$2^{66.26}$	0.6283	1.0	0.8141
28	$2^{65.87}$	0.7166	0.7624	0.7395	68	$2^{66.27}$	0.6283	1.0	0.8141
29	$2^{65.88}$	0.6956	0.8188	0.7572	69	$2^{66.28}$	0.6283	1.0	0.8141
30	$2^{65.89}$	0.6791	0.8633	0.7712	70	$2^{66.29}$	0.6283	1.0	0.8141
31	$2^{65.9}$	0.6663	0.8977	0.782	71	$2^{66.3}$	0.6283	1.0	0.8141
32	$2^{65.91}$	0.6565	0.924	0.7903	72	$2^{66.31}$	0.6283	1.0	0.8141
33	$2^{65.92}$	0.6491	0.9439	0.7965	73	$2^{66.32}$	0.6283	1.0	0.8141
34	$2^{65.93}$	0.6436	0.9588	0.8012	74	$2^{66.33}$	0.6283	1.0	0.8141
35	$2^{65.94}$	0.6395	0.9698	0.8047	75	$2^{66.34}$	0.6283	1.0	0.8141
36	$2^{65.95}$	0.6365	0.978	0.8072	76	$2^{66.35}$	0.6283	1.0	0.8141
37	$2^{65.96}$	0.6342	0.984	0.8091	77	$2^{66.36}$	0.6283	1.0	0.8141
38	$2^{65.97}$	0.6326	0.9884	0.8105	78	$2^{66.37}$	0.6283	1.0	0.8141
39	$2^{65.98}$	0.6314	0.9916	0.8115	79	$2^{66.38}$	0.6283	1.0	0.8141
40	$2^{65.99}$	0.6305	0.994	0.8123	80	$2^{66.39}$	0.6283	1.0	0.8141