CS5760: Topics in Cryptanalysis **16 April 2025**

Lecture 7: Yoyo Tricks with AES

Instructors: Maria Francis and M. V. Panduranga Rao *Scribe: Gautam Singh*

7.1 Introduction

The yoyo game was introduced by Biham et al. in the cryptanalysis of SKIPJACK in 1998. It is based on making new pairs of plaintexts and ciphertexts that preserve a certain property inherited from the original pair. This leads to a partition of the plaintext and ciphertext spaces where each partition is closed under exchange operations.

The yoyo game is quite similar to the boomerang attack, and has been used to build distinguishers for Feistel networks. They can also attack substitution permutation networks (SPNs) that iterate a round function $A \circ S$ where A is an affine transformation and S is a non-linear S-box layer.

7.2 Yoyo Analysis of Generic SPNs

For simplicity, we analyse permutations on $\mathbb{F}_q^n$ for $q = 2^k$ of the form $F(x) = S \circ L \circ S \circ L \circ S$, where L is a linear transformation as opposed to an affine transformation. An element of $\mathbb{F}_q^n$ is of the form $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_n)$ where each $\alpha_i \in \mathbb{F}_q$ is called a *word*.

To compare two differences according to their zero positions, we use the following.

Definition 7.1 (Zero Difference Pattern)**.** Let $\alpha \in \mathbb{F}_q^n$. Then, the zero difference pattern of α is given by

$$\nu(\alpha) \triangleq (z_0, z_1, \ldots, z_{n-1}) \tag{7.1}$$

where $z_i = 1$ if $\alpha_i = 0$ or $z_i = 0$ otherwise.

Clearly, $\nu(\alpha) \in \mathbb{F}_2^n$ and the complement of the zero-difference pattern is called the *activity pattern*. Linear transformations do not preserve the zero difference pattern, but permutations do.

Lemma 7.1. *For two states $\alpha, \beta \in \mathbb{F}_q^n$, the zero pattern of their difference is preserved through S. Mathematically,*

$$\nu(\alpha \oplus \beta) = \nu(S(\alpha) \oplus S(\beta)). \tag{7.2}$$

Proof. This is evident from the fact that $\alpha_i \oplus \beta_i = 0 \iff s(\alpha_i) \oplus s(\beta_i) = 0$ since s is a permutation. □

We make extensive use of the following definition.

Definition 7.2. For a vector $v \in \mathbb{F}_2^n$ and a pair of states $\alpha, \beta \in \mathbb{F}_q^n$ define $\rho^v(\alpha, \beta) \in \mathbb{F}_q^n$ where

$$\rho^v(\alpha, \beta)_i \triangleq \alpha_i v_i \oplus \beta_i (v_i \oplus 1) = \begin{cases} \alpha_i & v_i = 1 \\ \beta_i & v_i = 0 \end{cases}. \tag{7.3}$$

From the definition it is evident that

$$\rho^v(\alpha, \beta) \oplus \rho^v(\beta, \alpha) = \alpha \oplus \beta. \tag{7.4}$$

The function ρ^v has some interesting properties which are stated and proved below.

Lemma 7.2. *Let $\alpha, \beta \in \mathbb{F}_q^n$ and $v \in \mathbb{F}_2^n$. Then, ρ commutes with the S-box layer. Mathematically,*

$$\rho^v(S(\alpha), S(\beta)) = S(\rho^v(\alpha, \beta)) \tag{7.5}$$

and thus

$$S(\alpha) \oplus S(\beta) = S(\rho^v(\alpha, \beta)) \oplus S(\rho^v(\beta, \alpha)). \tag{7.6}$$

Proof. S operates on each word independently and the result follows immediately from Definition 7.2. □

Lemma 7.3. *For a linear transformation $L(x) = L(x_0, x_1, \ldots, x_{n-1})$ acting on n words we have*

$$L(\alpha) \oplus L(\beta) = L(\rho^v(\alpha, \beta)) \oplus L(\rho^v(\beta, \alpha)) \tag{7.7}$$

for any $v \in \mathbb{F}_2^n$.

Proof. Using (7.4) and the linearity of L, we have

$$L(\alpha) \oplus L(\beta) = L(\alpha \oplus \beta) = L(\rho^v(\alpha, \beta) \oplus \rho^v(\beta, \alpha)) = L(\rho^v(\alpha, \beta)) \oplus L(\rho^v(\beta, \alpha)) \tag{7.8}$$

as desired. □

Using Lemma 7.2 and Lemma 7.3, we have

$$L(S(\alpha)) \oplus L(S(\beta)) = L(S(\rho^v(\alpha, \beta))) \oplus L(S(\rho^v(\beta, \alpha))), \tag{7.9}$$

however switching S and L does not guarantee equality in (7.9).

Observe that the zero difference pattern does not change when we apply L or S to any pair $\alpha' = \rho^v(\alpha, \beta)$ and $\beta' = \rho^v(\beta, \alpha)$. Thus, unlike (7.9), it can be shown that

$$\nu(S(L(\alpha)) \oplus S(L(\beta))) = \nu(S(L(\rho^v(\alpha, \beta))) \oplus S(L(\rho^v(\beta, \alpha)))). \tag{7.10}$$

In other words, although equality may not hold, the differences are zero in exactly the same positions when $S \circ L$ is applied.

The above results can be summarised as Theorem 7.1, which is heavily used in the yoyo attack.

Theorem 7.1. *Let $\alpha, \beta \in \mathbb{F}_q^n$ and $\alpha' = \rho^v(\alpha, \beta), \beta' = \rho^v(\beta, \alpha)$. Then,*

$$\nu(S \circ L \circ S(\alpha) \oplus S \circ L \circ S(\beta)) = \nu(S \circ L \circ S(\alpha') \oplus S \circ L \circ S(\beta')). \tag{7.11}$$

Proof. The proof follows from the following observations.

1. Lemma 7.2 gives $S(\alpha) \oplus S(\beta) = S(\alpha') \oplus S(\beta')$.
2. The linearity of L gives $L(S(\alpha)) \oplus L(S(\beta)) = L(S(\alpha')) \oplus L(S(\beta'))$.
3. Finally, Lemma 7.1 gives (7.11).

□

7.2.1 Yoyo Distinguisher for Two Generic SP-Rounds

Two generic SP rounds can be represented as $G_2' = L \circ S \circ L \circ S$, where the last linear layer can be removed to instead represent it as $G_2 = S \circ L \circ S$. If we fix a pair of plaintexts p^0, p^1 with a paritcular zero difference pattern $\nu(p^0 \oplus p^1)$, then from the corresponding ciphertexts c^0, c^1, we can construct another pair of new ciphertexts c'^0, c'^1 such that their decrypted plaintexts p'^0, p'^1 also have the same zero difference pattern. This follows directly from Theorem 7.1 and holds with probability 1.

Theorem 7.2 (Generic Yoyo Game for Two SP-Rounds)**.** *Let $p^0 \oplus p^1 \in \mathbb{F}_q^n$, $c^0 = G_2(p^0)$ and $c^1 = G_2(p^1)$. Then for any $v \in bF_2^n$, let $c'^0 = \rho^v(c^0, c^1)$ and $c'^1 = \rho^v(c^1, c^0)$. Then,*

$$\nu(G_2^{-1}(c'^0) \oplus G_2^{-1}(c'^1)) = \nu(p'^0 \oplus p'^1) = \nu(p^0 \oplus p^1). \tag{7.12}$$

Proof. Since S^{-1} is also a permutation and L^{-1} is a linear transformation, we invoke Theorem 7.1 on $G_2^{-1} = S^{-1} \circ L^{-1} \circ S^{-1}$ to obtain (7.12). □

Theorem 7.2 gives us a straightforward distinguisher for two generic SP-rounds requiring two plaintexts and two adaptively chosen ciphertexts. A random permutation would not give back a pair of decrypted plaintexts that still have the same zero difference pattern with very high probability. One can go the other way to generate two ciphertexts and then observe the ciphertexts of the adaptively chosen plaintexts.

7.2.2 Analysis of Three Generic SP-Rounds

As before, three SP rounds can be modeled as $G_3 = S \circ L \circ S \circ L \circ S$. For two states α and β, using Theorem 7.2, it follows that

$$\nu(G_2^{-1}(\rho^v(G_2(\alpha), G_2(\beta))) \oplus G_2^{-1}(\rho^v(G_2(\beta), G_2(\alpha)))) = \nu(\alpha \oplus \beta). \tag{7.13}$$

Since G_2 and G_2^{-1} have identical forms, we have

$$\nu(G_2(\rho^v(G_2^{-1}(\alpha), G_2^{-1}(\beta))) \oplus G_2(\rho^v(G_2^{-1}(\beta), G_2^{-1}(\alpha)))) = \nu(\alpha \oplus \beta). \tag{7.14}$$

Finally, from Lemma 7.2, zero difference patterns are preserved through an S-box layer. Putting it all together gives us the following theorem.

Theorem 7.3 (Generic Yoyo Game for 3 SP-Rounds)**.** *Let $G_3 = S \circ L \circ S \circ L \circ S$. If $p^0, p^1 \in \mathbb{F}_q^n$ and $c^0 = G_3(p^0)$, $c^1 = G_3(p^1)$, then*

$$\nu(G_2(\rho^{v_1}(p^0, p^1)) \oplus G_2(\rho^{v_1}(p^1, p^0))) = \nu(G_2^1(\rho^{v_2}(c^0, c^1)) \oplus G_2^{-1}(\rho^{v_2}(c^1, c^0))) \tag{7.15}$$

for any $v_1, v_2 \in \mathbb{F}_2^n$. Moreover, for any $z \in \mathbb{F}_2^n$, define $R_P(z) \triangleq \{(p^0, p^1) \mid \nu(G_2(p^0) \oplus G_2(p^1)) = z\}$ and $R_C(z) \triangleq \{(c^0, c^1) \mid \nu(G_2^{-1}(c^0) \oplus G_2^{-1}(c^1)) = z\}$. Then,

$$(G_3(\rho^v(p^0, p^1)), G_3(\rho^v(p^1, p^0))) \in R_C(z) \tag{7.16}$$

for any $(p^0, p^1) \in R_P(z)$, while

$$(G_3^{-1}(\rho^v(c^0, c^1)), G_3^{-1}(\rho^v(c^1, c^0))) \in R_C(z) \tag{7.17}$$

for any $(c^0, c^1) \in R_C(z)$.

Thus, given a pair in $R_P(z)$, we can generate new pairs that belong to $R_P(z)$ and $R_C(z)$ with probability 1.

The key idea behind a distinguisher for three SP-rounds is to get a pair with a particular Hamming weight of the zero difference pattern and then detect this occurence. The probability that a random pair of plaintexts has a sum with nonzero difference pattern containing exactly m zeros is $\binom{n}{m}\frac{(q-1)^m}{q^n}$ where $q = 2^k$. Thus, we need to test approximately the inverse of that number of pairs to find one correct pair.

Detecting a correct pair is more involved. Suppose $(p_1, p_2) \in R_P(z)$ and let the respective ciphertexts be (c_1, c_2). Let A be the affine layer in an SASAS construction. Assume that $S^{-1}(c^0) = x \oplus z$ and $S^{-1}(c^1) = y \oplus z$, where $A^{-1}(x), A^{-1}(y)$ and $A^{-1}(z)$ are non-zero only in the positions where z is zero. It follows that x and y belong to a linear subspace U of dimension $n-m$ while z belongs to the complementary linear subspace V of dimension m such that $U \oplus V = \mathbb{F}_q^n$. Thus, we need to investiage whether $c_1 \oplus c_2 = S(x \oplus z) \oplus S(y \oplus z)$ has some distinguishing properties.

7.3 Applications to AES

7.3.1 Preliminaries

The round function in AES is represented as operations over $\mathbb{F}_q^{4\times 4}$ where $q = 2^8$. One round of AES can be written as $R = AK \circ MC \circ SR \circ SB$. Since we are working with differences, we can strip AK operations. Further, SR and SB commute. Thus, two rounds of AES can be written as

$$R^{2\prime} = MC \circ SR \circ (SB \circ MC \circ SB) \circ SR \tag{7.18}$$

where $S = SB \circ MC \circ SB$ can be thought of as four parallel 32-bit super S-boxes. Finally, the initial SR has no effect, thus we can rewrite two rounds of AES as

$$R^2 = MC \circ SR \circ S. \tag{7.19}$$

Now, considering $S = SB \circ MC \circ SB$ and $L = SR \circ MC \circ SR$, four rounds of AES can be represented using (7.18) as

$$R^{4\prime} = MC \circ SR \circ S \circ L \circ S \circ SR \tag{7.20}$$

which ends up becoming

$$R^4 = S \circ L \circ S. \tag{7.21}$$

This can also be used to show that a lower bound on the number of active S boxes over four rounds is 25. This is because the number of active super S-boxes is 5 due to the linear layer and there are at least 5 active S boxes insided a super S-box due to the MixColumns matrix.

Similarly, six rounds of AES can be written as

$$R^6 = S \circ L \circ S \circ L \circ S. \tag{7.22}$$

For convenience, we introduce the following definition.

Definition 7.3. Let $Q \triangleq SB \circ MC \circ SR$ and $Q' \triangleq SR \circ MC \circ SB$.

Since two rounds of AES correspond to one generic SPN round, we must exploit the properties of one AES round to create distinguishers for an odd number of rounds. Adding another round at the end of (7.18), three rounds of AES can be written as $Q \circ S$ and similarly, five rounds of AES can be written as $S \circ L \circ S \circ Q'$.

We now consider some properties of Q and Q'. For a binary vector $z \in \mathbb{F}_4^2$ of weight t, let V_z denote the subspace of $q^{4\cdot(4-t)}$ states $x = (x_0, x_1, x_2, x_3)$ where $x_i \in \mathbb{F}_q^4$ if $z_i = 0$ or $x_i = 0$ otherwise. For any state $a = (a_0, a_1, a_2, a_3)$, let

$$T_{z,a} \triangleq \{Q(a \oplus x) \mid x \in V_z\}. \tag{7.23}$$

Note that the sets $T_{z,a}$ depend on keyed functions such as those depicted in Definition 7.3. Let H_i denote the image of the i-th word in $SR(a \oplus x)$ for $x \in V_z$. Notice that $|H_i| = q^{4-t}$. Define

$$T_i^{z,a} \triangleq SB \circ MC(H_i). \tag{7.24}$$

Since SB and MC operate on each word individually, we obtain the following.

Lemma 7.4. *The set $T_{z,a}$ satisfies*

$$T_{z,a} = T_0^{z,a} \times T_1^{z,a} \times T_2^{z,a} \times T_3^{z,a} \tag{7.25}$$

where $|T_i^{z,a}| = q^{4-hw(z)}$, with $hw(z)$ denoting the Hamming weight of z.

Proof. Notice that each word of $Q(a \oplus x)$ contributes one byte to each word after SR. Thus, if $4-t$ words are nonzero, it follows that each word after SR can take exactly q^{4-t} values. Thus, $T_i^{z,a} = SB \circ MC(H_i)$. □

A similar property can be derived for Q' and its inverse as well.

Later algorithms make use of the primitive SimpleSWAP algorithm shown in Algorithm 7.1 to perform the yoyo itself. Further, note that we consider r-round AES encryption wihtout first SR and last $SR \circ MC$.

Algorithm 7.1 Swaps the first word where texts are different and returns one word.

```
1: function SimpleSWAP(x^0, x^1)                    ▷ x^0 ≠ x^1
2:     x'^0 ← x'^1
3:     for i from 0 to 3 do
4:         if x_i^0 ≠ x_i^1 then
5:             x'^0_i ← x'^1_i
6:             return x'^0
```

7.3.2 Yoyo Distinguisher for Three Rounds of AES

We have seen that three rounds of AES can be written as $R^3 = Q \circ S$. We use Lemma 7.4 to create the distinguisher. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$. Using Lemma 7.1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$. Then, from Lemma 7.4, $Q(S(p^0)) = c^0$ and $Q(S(p^1)) = c^1$ also belong to $T_{z,a}$. Further, each word is drawn from the subsets $T_i^{z,a}$. In paritcular, we have

$$T'_{z,a} = \{c_0^0, c_0^1\} \times \{c_1^0, c_1^1\} \times \{c_2^0, c_2^1\} \times \times \{c_3^0, c_3^1\} \subset T_{z,a}. \tag{7.26}$$

where the size of $T'_{z,a}$ is at most 2^4 and $\{c_i^0, c_i^1\} \subset T_i^{z,a}$. Thus, any other ciphertext $c' \neq c^0, c^1$ from $T'_{z,a}$ satisfies $\nu(Q^{-1}(c') \oplus S(p^0)) = \nu(Q^{-1}(c') \oplus S(p^1)) = \nu(S(p^0) \oplus S(p^1))$. In particular, we have $\nu(R^{-3}(c') \oplus p^0) = \nu(R^{-3}(c') \oplus p^1) = \nu(p^0 \oplus p^1)$. If a random permutation were used, the chosen ciphertext c' would satisfy this condition with probability 2^{-96}. The distinguisher is summarised in Algorithm 7.2. Clearly, only two plaintexts and one adaptively chosen ciphertext is required.

Algorithm 7.2 Distinguisher for Three Rounds of AES

Input: Plaintexts p^0, p^1 with $hw(\nu(p^0 \oplus p^1)) = 3$
Output: 1 for AES, -1 otherwise
1: $c^0 \leftarrow enc_k(p^0, 3)$, $c^1 \leftarrow enc_k(p^1, 3)$
2: $c' \leftarrow \text{SIMPLESWAP}(c^0, c^1)$
3: $p' \leftarrow dec_k(c', 3)$
4: **if** $\nu(p^0 \oplus p^1) = \nu(p' \oplus p^1)$ **then**
5: **return** 1
6: **else**
7: **return** -1

7.3.3 Yoyo Distinguisher for Four Rounds of AES

Four rounds of AES can be represented as $R^4 = S \circ L \circ S$ after simplification. We make use of Theorem 7.2 to create the distinguisher. Again, the new ciphertexts are created by simply exchanging words between the two obtined ciphertexts, as shown in Algorithm 7.3. This distinguisher requires two plaintexts and two adaptively chosen ciphertexts.

Algorithm 7.3 Distinguisher for Four Rounds of AES

Input: Plaintexts p^0, p^1 with $hw(\nu(p^0 \oplus p^1)) = 3$
Output: 1 for AES, -1 otherwise
1: $c^0 \leftarrow enc_k(p^0, 4)$, $c^1 \leftarrow enc_k(p^1, 4)$
2: $c'^0 \leftarrow \text{SIMPLESWAP}(c^0, c^1)$, $c'^1 \leftarrow \text{SIMPLESWAP}(c^1, c^0)$
3: $p'^0 \leftarrow dec_k(c'^0, 4)$, $p'^1 \leftarrow dec_k(c'^1, 4)$
4: **if** $\nu(p^0 \oplus p^1) = \nu(p'^0 \oplus p'^1)$ **then**
5: **return** 1
6: **else**
7: **return** -1

7.3.4 Yoyo Distinguisher for Five Rounds of AES

Five rounds of AES can be written as $R^5 = S \circ L \circ S \circ Q' = R^4 \circ Q'$. If the difference between two plaintexts after Q' is zero in t words, we can apply the yoyo game and get new plaintext pairs that are zero in exactly the same words after Q' and thus, reside in the same sets by Lemma 7.4. In paritcular, if a pair of plaintexts p^0, p^1 are encrypted through Q' to a pair of intermediate states with zero difference in 3 out of 4 words, then they have probability q^{-1} of having the same value in a particular word, since $|T_i^{z,a}| = q^{4-3} = q$ by Lemma 7.4. Another property, this time of the MixColumns matrix can be exploited to get a tighter bound, which is stated as Lemma 7.5.

Lemma 7.5. *Let M denote a 4×4 MixColumns matrix and $x \in \mathbb{F}_q^4$. If t bytes in x are zero, then $x \cdot M$ or $x \cdot M^{-1}$ cannot contain $4 - t$ or more zeros.*

This is used to prove Theorem 7.4.

Theorem 7.4. *Let a and b denote two states where $\nu(Q'(a) \oplus Q'(b))$ 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.*

Proof. From Lemma 7.4, words in the same position of a and b are drawn from $T_i^{z,a}$ with size q^{4-t}. Thus, words in the same position are equal with probability q^{t-4}. Since t words are zero in $Q'(a) \oplus Q'(b)$, t bytes are zero in each word of $SR^{-1}(Q'(a)) \oplus SR^{-1}(Q'(b))$. From Lemma 7.5, we see that $4-t$ bytes cannot be zero in each word after MC^{-1}. This is preserved through SB^{-1} and XOR with the round key. □

To build the distinguisher, we need to create enough plaintext pairs so that there will be exactly t zeros after the application of Q'. Notice that two equal columns remain equal on applying $MC \circ SB$. Thus, the adversary chooses pairs (p^0, p^1) which are nonzero in exactly one word and tries enough pairs until the corresponding active word after applying $MC \circ SB$ on that word has t zero bytes. This would imply $Q'(p^0) \oplus Q'(p^1)$ has t zero words. Playing the yoyo game on R^4 will return at most 7 new plaintext pairs which have the same zero difference pattern after one round and obey Theorem 7.4. This is used to distinguish five-round AES from a random permutation.

The probability that a pair (p^0, p^1) with a zero difference pattern of weight 3 has a zero difference pattern of weight t when encrypted through Q' is given by

$$p_b(t) = \binom{4}{t} q^{-t} \tag{7.27}$$

where $q = 2^8$. Thus, we require $p_b(t)^{-1}$ pairs to get one such pair. To distinguish a correct pair, notice that for a random pair of plaintexts, the probability that $4-t$ bytes are zero simultaneously in any of the 4 words is approximately

$$4p_b(4-t) = 4 \cdot \binom{4}{t} \cdot q^{t-4} \tag{7.28}$$

while for a correct pair it is $4 \cdot q^{t-4}$. Thus, each pair of plaintexts requires $\frac{p_b(4-t)^{-1}}{4}$ plaintext pairs using the yoyo game. Thus, the total data complexity is

$$2 \cdot (p_b(t)^{-1} \cdot (4 \cdot p_b(4-t))^{-1}) = \frac{p_b(t) \cdot p_b(4-t)^{-1}}{2}. \tag{7.29}$$

For $t = 2$, the data complexity is minimum at approximately $2^{25.8}$. The overall distinguisher is shown in Algorithm 7.4.

7.3.5 A Five Round Key Recovery Yoyo on AES

The aim of this attack is to find the first round key k_0 XORed in front of R^5. The MixColumns matrix M in AES is given by

$$M = \begin{pmatrix} \alpha & \alpha \oplus 1 & 1 & 1 \\ 1 & \alpha & \alpha \oplus 1 & 1 \\ 1 & 1 & \alpha & \alpha \oplus 1 \\ \alpha \oplus 1 & 1 & 1 & \alpha \end{pmatrix}. \tag{7.30}$$

The function $MC \circ SB$ works on each word independently, thus assume we pick two plaintexts p^0 and p^1 where the first words are given by $p_0^0 = (0, i, 0, 0)$ and $p_0^1 = (z, z \oplus i, 0, 0)$ where $z \in \mathbb{F}_q \setminus \{0\}$ and the three other words are equal. Let $k_0 = (k_{0,0}, k_{0,1}, k_{0,2}, k_{0,3})$ denote key bytes XORed with the first word of the plaintext. The difference between the first words after partial encryption of the two plaintexts $MC \circ SB \circ AK$ becomes

$$\alpha b_0 \oplus (\alpha \oplus 1) b_1 = y_0 \tag{7.31}$$

$$b_0 \oplus \alpha b_1 = y_1 \tag{7.32}$$

Algorithm 7.4 Distinguisher for Five Rounds of AES

Output: 1 for AES, -1 otherwise

$cnt1 \leftarrow 0$.
while $cnt1 < 2^{13.4}$ **do**
  $cnt1 \leftarrow cnt1 + 1$.
  $p^0, p^1 \leftarrow$ generate random pair with $hw(\nu(p^0 \oplus p^1)) = 3$.
  $cnt2 \leftarrow 0$, $WrongPair \leftarrow False$.
  while $cnt2 < 2^{11.4}$ & $WrongPair = False$ **do**
    $cnt2 \leftarrow cnt2 + 1$.
    $c^0 \leftarrow enc_k(p^0, 5)$, $c^1 \leftarrow enc_k(p^1, 5)$.
    $c'^0 \leftarrow \text{SIMPLESWAP}(c^0, c^1)$, $c'^1 \leftarrow \text{SIMPLESWAP}(c^1, c^0)$.
    $p'^0 \leftarrow dec_k(c'^0, 5)$, $p'^1 \leftarrow dec_k(c'^1, 5)$.
    for i from 0 to 3 **do**
      if $hw(\nu(p_i)) \geq 2$ **then**
        $WrongPair = True$
    $p'^0 \leftarrow \text{SIMPLESWAP}(p^0, p^1)$, $p'^1 \leftarrow \text{SIMPLESWAP}(p^1, p^0)$.
  if $WrongPair = False$ **then**
    return 1 ▷ *Did not find difference with two or more zeros.*
return -1

$$b_0 \oplus b_1 = y_2 \tag{7.33}$$
$$(\alpha \oplus 1)b_0 \oplus b_1 = y_3 \tag{7.34}$$

where $b_0 = s(k_{0,0}) \oplus s(z \oplus k_{0,0})$ and $b_1 = s(k_{0,1} \oplus i) \oplus s(k_{0,1} \oplus z \oplus i)$. In paritcular, (7.33) can be written as

$$s(k_{0,0}) \oplus s(k_{0,0} \oplus z) \oplus s(k_{0,1} \oplus i) \oplus s(k_{0,1} \oplus z \oplus i) = y_2. \tag{7.35}$$

As i runs over all $\mathbb{F}_q$, we see that $y_2 = 0$ for $i \in \{k_{0,0} \oplus k_{0,1}, k_{0,0} \oplus k_{0,1} \oplus z\}$. Hence, there will be at least two values of i for which $y_2 = 0$. Define $B = M \circ s^4$ to be the action of $MC \circ SB$ on one column, where s^4 is the concatenation of four S-boxes in parallel. We prepare a set $\mathcal{P}$ of plaintexts p^0 and p^1 where $p_0^0 = (0, i, 0, 0)$ and $p_0^1 = (z, z \oplus i, 0, 0)$. Let c^0, c^1 be the respective ciphertexts. Pick 5 new ciphertext pairs $(c'^0, c'^1) = (\rho^v(c^0, c^1), \rho^v(c^1, c^0))$ and let p'^0, p'^1 be the respective plaintexts. A correct pair will satisfy

$$B(p_0'^0 \oplus k_0) \oplus B(p_0'^1 \oplus k_0) = (z_0, z_1, 0, z_3). \tag{7.36}$$

The adversary can now test the remaining 2^{24} candidate keys and find whether the third byte of the first word is zero for all 5 pairs of plaintexts, where $k_{0,0} \oplus k_{0,1} \in \{i, i \oplus z\}$ for known i and z. This holds for all 5 pairs at random with probability $2^{-8 \cdot 5} = 2^{-40}$. Hence, a false positive might occur with probability 2^{-16} when testing 2^{24} keys. This probability can be reduced by testing with additional pairs when the test succeeds on the first five pairs, which is rare. Thus, the total data complexity (plaintexts and ciphertexts) is

$$D = 2 \cdot 2^8 \cdot 5 \approx 2^{11.32}. \tag{7.37}$$

For the computational complexity, we need to test 2^{24} keys for each set of plaintexts as we only need to set $k_{0,1} = k_{0,0} \oplus i$, since $k_{0,0}, i \in \mathbb{F}_q$. For each key, we will have $2 \cdot 4$ S-box lookups for 5 pairs to check (7.36), giving a total complexity of $2^{24} \cdot 2 \cdot 4 \cdot 5 \cdot 2^8 = 2^{37.3}$, which cooresponds to approximately2^{31} 5-rounds of AES (assuming 80 S-box lookups per round).

Since the adversary knows k_0, they can make a pair of words $a_0', b_0' \in \mathbb{F}_q^4$ that differ only in their first byte. The actual plaintext pair is obtained by performing $AK^{-1} \circ SB^{-1} \circ MC^{-1}$ on it to obtain a_0, b_0, which is

used to create plaintexts $p^0 = (a_0, 0, 0, 0)$ and $p^1 = (b_0, 0, 0, 0)$. However, this pair is useless in recovering the other subkeys since the last three words are equal. Instead, the yoyo can be used from this initial pair to generate pairs (p'^0, p'^1) that are with high probability different in the last three words and whose difference after $SR \circ MC \circ SB \circ AK$ is non-zero only in the first word. To attack k_1, notice that each of the m pairs returned by the yoyo satisfy

$$B(p_1'^0 \oplus k_1) \oplus B(p_1'^1 \oplus k_1) = (0, w, 0, 0) \tag{7.38}$$

for some $w \in \mathbb{F}_q$ and fixed k_1. This is because the i-th byte of the i-th word can be nonzero before SR. Notice that (7.38) can be written as

$$M^{-1}(0, w, 0, 0) = w \cdot M_2^{-1} = s^4(p_1'^0 \oplus k_1) \oplus s^4(p_1'^1 \oplus k_1). \tag{7.39}$$

where M_i^{-1} denotes the i-th column of M^{-1}. This can be used to solve for k_1 on fixing any byte in k_1. Hence, at most $4 \cdot 2^8$ guesses are spent on getting the correct key. Similarly, k_2 and k_3 can be found using analogous relationships with columns of M^{-1}.

A similar procedure can be followed for later rounds. To recover the remaining 3 round subkeys at once, the adversary should test the solutions against 4 plaintext pairs to ensure a comfortable margin against false positives. Since the initial pair is useless, 5 pairs are used to recover the full key. The key recovery algorithm is shown in Algorithm 7.5.

Algorithm 7.5 Key Recovery for Five Rounds of AES

Output: Secret key k_0
for i from 0 to $2^8 - 1$ **do**
  $p^0 \leftarrow 0$, $p^1 \leftarrow 0$
  $p_0^0 \leftarrow (0, i, 0, 0)$, $p_0^1 \leftarrow (1, 1 \oplus i, 0, 0)$
  $\mathcal{S} \leftarrow \{(p^0, p^1)\}$
  while $|\mathcal{S}| < 5$ **do**
    $c^0 \leftarrow enc_k(p^0, 5)$, $c^1 \leftarrow enc_k(p^1, 5)$
    $c'^0 \leftarrow \text{SIMPLESWAP}(c^0, c^1)$, $c'^1 \leftarrow \text{SIMPLESWAP}(c^1, c^0)$
    $p'^0 \leftarrow dec_k(c'^0, 5)$, $p'^1 \leftarrow dec_k(c'^1, 5)$
    $p^0 \leftarrow \text{SIMPLESWAP}(p'^0, p'^1)$, $p^1 \leftarrow \text{SIMPLESWAP}(p'^1, p'^0)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(p^0, p^1)\}$
  for all 2^{24} key candidates k_0 **do**
    for all $(p^0, p^1) \in \mathcal{S}$ **do**
      if $l_3(s^4(p_0^0 \oplus k_0) \oplus s^4(p_0^1 \oplus k_0)) \neq 0$ **then**
        Break and jump to next key
    return k_0