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