Yoyo Tricks with AES

Gautam Singh

Indian Institute of Technology Hyderabad

April 21, 2025

Yoyo Game

1. Introduced by Biham et. al in 1998 for cryptanalysis of SKIPJACK.

Yoyo Game

1. Introduced by Biham et. al in 1998 for cryptanalysis of SKIPJACK.
2. Main idea is to make new pairs of plaintexts and ciphertexts that preserve a property from the original plaintext.

Yoyo Game

1. Introduced by Biham et. al in 1998 for cryptanalysis of SKIPJACK.
2. Main idea is to make new pairs of plaintexts and ciphertexts that preserve a property from the original plaintext.
3. Partitions the plaintext and ciphertext spaces where each partition is closed under exchange operations.

Yoyo Game

1. Introduced by Biham et. al in 1998 for cryptanalysis of SKIPJACK.
2. Main idea is to make new pairs of plaintexts and ciphertexts that preserve a property from the original plaintext.
3. Partitions the plaintext and ciphertext spaces where each partition is closed under exchange operations.
4. Similar to the boomerang attack and works with both Feistel networks and 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.

Yoyo Game

1. Introduced by Biham et. al in 1998 for cryptanalysis of SKIPJACK.
2. Main idea is to make new pairs of plaintexts and ciphertexts that preserve a property from the original plaintext.
3. Partitions the plaintext and ciphertext spaces where each partition is closed under exchange operations.
4. Similar to the boomerang attack and works with both Feistel networks and 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.
5. For analysis, we consider permutations that iterate $L \circ S$, where L is a linear transformation.

Zero Difference Pattern

Suppose $q = 2^k$. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}_q^n$, where each $\alpha_i \in \mathbb{F}_q$ is called a *word*.

Zero Difference Pattern

Suppose $q = 2^k$. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}_q^n$, where each $\alpha_i \in \mathbb{F}_q$ is called a *word*.

Definition 1 (Zero Difference Pattern)

Let $\alpha \in \mathbb{F}_q^n$. Then, the *zero difference pattern* of α is given by

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

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

Zero Difference Pattern

Suppose $q = 2^k$. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}_q^n$, where each $\alpha_i \in \mathbb{F}_q$ is called a *word*.

Definition 1 (Zero Difference Pattern)

Let $\alpha \in \mathbb{F}_q^n$. Then, the *zero difference pattern* of α is given by

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

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

Observe that $\nu(\alpha) \in \mathbb{F}_2^n$. The complement of $\nu(\alpha)$ is called the *activity pattern*.

Properties of Zero Difference Pattern

Lemma 1

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

$$\nu(\alpha \oplus \beta) = \nu(S(\alpha) \oplus S(\beta)). \tag{2}$$

Properties of Zero Difference Pattern

Lemma 1

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

$$\nu(\alpha \oplus \beta) = \nu(S(\alpha) \oplus S(\beta)). \tag{2}$$

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. $\square$

Mixture of Pairs

Definition 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

$$\rho^v(\alpha, \beta)_i \triangleq \alpha_i v_i \oplus \beta_i (v_i \oplus 1) = \begin{cases} \alpha_i & v_i = 1 \\ \beta_i & v_i = 0 \end{cases}. \tag{3}$$

Mixture of Pairs

Definition 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

$$\rho^v(\alpha, \beta)_i \triangleq \alpha_i v_i \oplus \beta_i(v_i \oplus 1) = \begin{cases} \alpha_i & v_i = 1 \\ \beta_i & v_i = 0 \end{cases}. \tag{3}$$

From the definition it is evident that

$$\rho^v(\alpha, \beta) \oplus \rho^v(\beta, \alpha) = \alpha \oplus \beta. \tag{4}$$

Effect of a Permutation

Lemma 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,

$$\rho^v(S(\alpha), S(\beta)) = S(\rho^v(\alpha, \beta)) \tag{5}$$

and thus

$$S(\alpha) \oplus S(\beta) = S(\rho^v(\alpha, \beta)) \oplus S(\rho^v(\beta, \alpha)). \tag{6}$$

Effect of a Permutation

Lemma 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,

$$\rho^v(S(\alpha), S(\beta)) = S(\rho^v(\alpha, \beta)) \tag{5}$$

and thus

$$S(\alpha) \oplus S(\beta) = S(\rho^v(\alpha, \beta)) \oplus S(\rho^v(\beta, \alpha)). \tag{6}$$

Proof.

S operates on each word independently and the result follows immediately from Definition 2. □

Effect of a Linear Transformation

Lemma 3

For a linear transformation $L(x) = L(x_0, x_1, \ldots, x_{n-1})$ *and for any* $v \in \mathbb{F}_2^n$,

$$L(\alpha) \oplus L(\beta) = L(\rho^v(\alpha, \beta)) \oplus L(\rho^v(\beta, \alpha)) \tag{7}$$

Effect of a Linear Transformation

Lemma 3

For a linear transformation $L(x) = L(x_0, x_1, \ldots, x_{n-1})$ and for any $v \in \mathbb{F}_2^n$,

$$L(\alpha) \oplus L(\beta) = L(\rho^v(\alpha, \beta)) \oplus L(\rho^v(\beta, \alpha)) \tag{7}$$

Proof.

Using (4) and the linearity of L, we have

$$\begin{aligned} L(\alpha) \oplus L(\beta) = L(\alpha \oplus \beta) &= L(\rho^v(\alpha, \beta) \oplus \rho^v(\beta, \alpha)) && (8) \\ &= L(\rho^v(\alpha, \beta)) \oplus L(\rho^v(\beta, \alpha)) && (9) \end{aligned}$$

□

Combined Effect

1. Using Lemma 2 and Lemma 3, we have

$$L(S(\alpha)) \oplus L(S(\beta)) = L(S(\rho^v(\alpha, \beta))) \oplus L(S(\rho^v(\beta, \alpha))), \tag{10}$$

Combined Effect

1. Using Lemma 2 and Lemma 3, we have

$$L(S(\alpha)) \oplus L(S(\beta)) = L(S(\rho^v(\alpha, \beta))) \oplus L(S(\rho^v(\beta, \alpha))), \tag{10}$$

2. Switching S and L does not guarantee equality in (10).

Combined Effect

1. Using Lemma 2 and Lemma 3, we have

$$L(S(\alpha)) \oplus L(S(\beta)) = L(S(\rho^{v}(\alpha,\beta))) \oplus L(S(\rho^{v}(\beta,\alpha))), \tag{10}$$

2. Switching S and L does not guarantee equality in (10).
3. Zero difference pattern does not change when L or S is applied to any pair $\alpha' = \rho^{v}(\alpha,\beta)$ and $\beta' = \rho^{v}(\beta,\alpha)$. Thus,

$$\nu(S(L(\alpha)) \oplus S(L(\beta))) = \nu(S(L(\rho^{v}(\alpha,\beta))) \oplus S(L(\rho^{v}(\beta,\alpha)))). \tag{11}$$

Combined Effect

1. Using Lemma 2 and Lemma 3, we have

$$L(S(\alpha)) \oplus L(S(\beta)) = L(S(\rho^v(\alpha,\beta))) \oplus L(S(\rho^v(\beta,\alpha))), \quad (10)$$

2. Switching S and L does not guarantee equality in (10).
3. Zero difference pattern does not change when L or S is applied to any pair $\alpha' = \rho^v(\alpha,\beta)$ and $\beta' = \rho^v(\beta,\alpha)$. Thus,

$$\nu(S(L(\alpha)) \oplus S(L(\beta))) = \nu(S(L(\rho^v(\alpha,\beta))) \oplus S(L(\rho^v(\beta,\alpha)))). \quad (11)$$

4. Although equality may not hold, differences are zero in exactly the same positions when $S \circ L$ is applied.

Summary Theorem

Theorem 1

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

$$\nu(S \circ L \circ S(\alpha) \oplus S \circ L \circ S(\beta)) = \nu(S \circ L \circ S(\alpha') \oplus S \circ L \circ S(\beta')). \quad (12)$$

Summary Theorem

Theorem 1

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

$$\nu(S \circ L \circ S(\alpha) \oplus S \circ L \circ S(\beta)) = \nu(S \circ L \circ S(\alpha') \oplus S \circ L \circ S(\beta')). \quad (12)$$

Proof.

The proof follows from the following observations.

1. Lemma 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 1 gives (12).

□

Analysis of Two Generic SP-Rounds

1. Two generic SP rounds can be represented as $G_2' = L \circ S \circ L \circ S$.

Analysis of Two Generic SP-Rounds

1. Two generic SP rounds can be represented as $G_2' = L \circ S \circ L \circ S$.
2. The last linear layer can be removed to represent it as $G_2 = S \circ L \circ S$.

Analysis of Two Generic SP-Rounds

1. Two generic SP rounds can be represented as $G_2' = L \circ S \circ L \circ S$.
2. The last linear layer can be removed to represent it as $G_2 = S \circ L \circ S$.
3. Fix a pair of plaintexts p^0, p^1 with a paritcular $\nu(p^0 \oplus p^1)$.

Analysis of Two Generic SP-Rounds

1. Two generic SP rounds can be represented as $G_2' = L \circ S \circ L \circ S$.
2. The last linear layer can be removed to represent it as $G_2 = S \circ L \circ S$.
3. Fix a pair of plaintexts p^0, p^1 with a paritcular $\nu(p^0 \oplus p^1)$.
4. From the corresponding ciphertexts c^0, c^1, construct another pair of 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 1 and holds with probability 1.

Summary Theorem

Theorem 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* $v \in \mathbb{F}_2^n$, *let* $c'^0 = \rho^v(c^0, c^1)$ *and* $c'^1 = \rho^v(c^1, c^0)$. *Then,*

$$\nu(G_2^{-1}(c'^0) \oplus G_2^{-1}(c'^1)) = \nu(p'^0 \oplus p'^1) = \nu(p^0 \oplus p^1). \tag{13}$$

Summary Theorem

Theorem 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* $v \in \mathbb{F}_2^n$, *let* $c'^0 = \rho^v(c^0, c^1)$ *and* $c'^1 = \rho^v(c^1, c^0)$. *Then,*

$$\nu(G_2^{-1}(c'^0) \oplus G_2^{-1}(c'^1)) = \nu(p'^0 \oplus p'^1) = \nu(p^0 \oplus p^1). \tag{13}$$

Proof.

Since S^{-1} is also a permutation and L^{-1} is a linear transformation, we invoke Theorem 1 on $G_2^{-1} = S^{-1} \circ L^{-1} \circ S^{-1}$ to obtain (13). □

Distiguisher for Two SP-Rounds

1. Theorem 2 gives us a straightforward distinguisher for two generic SP-rounds requiring two plaintexts and two adaptively chosen ciphertexts.

Distiguisher for Two SP-Rounds

1. Theorem 2 gives us a straightforward distinguisher for two generic SP-rounds requiring two plaintexts and two adaptively chosen ciphertexts.
2. A random permutation would not give back a pair of decrypted plaintexts that still have the same zero difference pattern with very high probability.

Distiguisher for Two SP-Rounds

1. Theorem 2 gives us a straightforward distinguisher for two generic SP-rounds requiring two plaintexts and two adaptively chosen ciphertexts.
2. A random permutation would not give back a pair of decrypted plaintexts that still have the same zero difference pattern with very high probability.
3. One can also generate two ciphertexts and then observe the ciphertexts of the adaptively chosen plaintexts.

Analysis of Three Generic SP-Rounds

1. As before, three SP rounds can be modeled as $G_3 = S \circ L \circ S \circ L \circ S$.

Analysis of Three Generic SP-Rounds

1. As before, three SP rounds can be modeled as $G_3 = S \circ L \circ S \circ L \circ S$.
2. For two states α and β, using Theorem 2, it follows that

$$\nu(G_2^{-1}(\rho^v(G_2(\alpha), G_2(\beta))) \oplus G_2^{-1}(\rho^v(G_2(\beta), G_2(\alpha)))) = \nu(\alpha \oplus \beta). \tag{14}$$

Analysis of Three Generic SP-Rounds

1. As before, three SP rounds can be modeled as $G_3 = S \circ L \circ S \circ L \circ S$.
2. For two states α and β, using Theorem 2, it follows that

$$\nu(G_2^{-1}(\rho^v(G_2(\alpha), G_2(\beta))) \oplus G_2^{-1}(\rho^v(G_2(\beta), G_2(\alpha)))) = \nu(\alpha \oplus \beta). \tag{14}$$

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

$$\nu(G_2(\rho^v(G_2^{-1}(\alpha), G_2^{-1}(\beta))) \oplus G_2(\rho^v(G_2^{-1}(\beta), G_2^{-1}(\alpha)))) = \nu(\alpha \oplus \beta). \tag{15}$$

Analysis of Three Generic SP-Rounds

1. As before, three SP rounds can be modeled as $G_3 = S \circ L \circ S \circ L \circ S$.
2. For two states α and β, using Theorem 2, it follows that

$$\nu(G_2^{-1}(\rho^v(G_2(\alpha), G_2(\beta))) \oplus G_2^{-1}(\rho^v(G_2(\beta), G_2(\alpha)))) = \nu(\alpha \oplus \beta). \tag{14}$$

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

$$\nu(G_2(\rho^v(G_2^{-1}(\alpha), G_2^{-1}(\beta))) \oplus G_2(\rho^v(G_2^{-1}(\beta), G_2^{-1}(\alpha)))) = \nu(\alpha \oplus \beta). \tag{15}$$

4. Finally, from Lemma 2, zero difference patterns are preserved through an S-box layer.

Summary Theorem

Theorem 3 (Generic Yoyo Game for Three 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

$$\nu(G_2(\rho^{v_1}(p^0, p^1)) \oplus G_2(\rho^{v_1}(p^1, p^0))) = \nu(G_2^{-1}(\rho^{v_2}(c^0, c^1)) \oplus G_2^{-1}(\rho^{v_2}(c^1, c^0))) \quad (16)$$

for any $v_1, v_2 \in \mathbb{F}_2^n$.

Summary Theorem

Theorem 3 (Generic Yoyo Game for Three SP-Rounds)

Moreover, for any $z \in \mathbb{F}_2^n$*, define*

$$R_P(z) \triangleq \{(p^0, p^1) \mid \nu(G_2(p^0) \oplus G_2(p^1)) = z\} \tag{16}$$

$$R_C(z) \triangleq \{(c^0, c^1) \mid \nu(G_2^{-1}(c^0) \oplus G_2^{-1}(c^1)) = z\} \tag{17}$$

Then, for any $(p^0, p^1) \in R_P(z)$, $(G_3(\rho^v(p^0, p^1)), G_3(\rho^v(p^1, p^0))) \in R_C(z)$ *and for any* $(c^0, c^1) \in R_C(z)$,
$(G_3^{-1}(\rho^v(c^0, c^1)), G_3^{-1}(\rho^v(c^1, c^0))) \in R_C(z)$.

Distinguisher for Three Generic SP-Rounds

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

Distinguisher for Three Generic SP-Rounds

1. Given a pair in $R_P(z)$, we can get new pairs that belong to $R_P(z)$ and $R_C(z)$ with probability 1.
2. 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.

Distinguisher for Three Generic SP-Rounds

1. Given a pair in $R_P(z)$, we can get new pairs that belong to $R_P(z)$ and $R_C(z)$ with probability 1.
2. 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.
3. 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 $q = 2^k$.

Distinguisher for Three Generic SP-Rounds

1. Given a pair in $R_P(z)$, we can get new pairs that belong to $R_P(z)$ and $R_C(z)$ with probability 1.
2. 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.
3. 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 $q = 2^k$.
4. Thus, we need to test approximately the inverse of that number of pairs to find one correct pair.

Distinguisher for Three Generic SP-Rounds

1. 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.

Distinguisher for Three Generic SP-Rounds

1. 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.
2. 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.

Distinguisher for Three Generic SP-Rounds

1. 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.
2. 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.
3. 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 $U \oplus V = \mathbb{F}_q^n$.

Distinguisher for Three Generic SP-Rounds

1. 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.
2. 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.
3. 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 $U \oplus V = \mathbb{F}_q^n$.
4. We need to investigate whether $c^0 \oplus c^1 = S(x \oplus z) \oplus S(y \oplus z)$ has some distinguishing properties.

Round Function of AES

1. 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

$$R = AK \circ MC \circ SR \circ SB. \tag{18}$$

Round Function of AES

1. 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

$$R = AK \circ MC \circ SR \circ SB. \tag{18}$$

2. Since differences are used, strip AK operations. SR and SB commute.

Round Function of AES

1. 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

$$R = AK \circ MC \circ SR \circ SB. \tag{18}$$

2. Since differences are used, strip AK operations. SR and SB commute.
3. Two rounds of AES can be written as

$$R^{2\prime} = MC \circ SR \circ (SB \circ MC \circ SB) \circ SR \tag{19}$$

where $S = SB \circ MC \circ SB$ can be thought of as four parallel 32-bit super S-boxes.

Round Function of AES

1. 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

$$R = AK \circ MC \circ SR \circ SB. \tag{18}$$

2. Since differences are used, strip AK operations. SR and SB commute.
3. Two rounds of AES can be written as

$$R^{2\prime} = MC \circ SR \circ (SB \circ MC \circ SB) \circ SR \tag{19}$$

   where $S = SB \circ MC \circ SB$ can be thought of as four parallel 32-bit super S-boxes.
4. The initial SR has no effect, thus $R^2 = MC \circ SR \circ S$.

Representing AES as Generic SP-rounds

1. Considering $S = SB \circ MC \circ SB$ and $L = SR \circ MC \circ SR$, four rounds of AES can be represented using (19) as $R^{4'} = MC \circ SR \circ S \circ L \circ S \circ SR$ which ends up becoming $R^4 = S \circ L \circ S$.

Representing AES as Generic SP-rounds

1. Considering $S = SB \circ MC \circ SB$ and $L = SR \circ MC \circ SR$, four rounds of AES can be represented using (19) as $R^{4\prime} = MC \circ SR \circ S \circ L \circ S \circ SR$ which ends up becoming $R^4 = S \circ L \circ S$.
2. This also shows that a lower bound on the number of active S boxes over four rounds is 25.

Representing AES as Generic SP-rounds

1. Considering $S = SB \circ MC \circ SB$ and $L = SR \circ MC \circ SR$, four rounds of AES can be represented using (19) as $R^{4\prime} = MC \circ SR \circ S \circ L \circ S \circ SR$ which ends up becoming $R^4 = S \circ L \circ S$.
2. This also shows that a lower bound on the number of active S boxes over four rounds is 25.
   - 5 active super S-boxes due to the linear layer.
   - At least 5 active S boxes inside a super S-box due to MixColumns.

Representing AES as Generic SP-rounds

1. Considering $S = SB \circ MC \circ SB$ and $L = SR \circ MC \circ SR$, four rounds of AES can be represented using (19) as $R^{4\prime} = MC \circ SR \circ S \circ L \circ S \circ SR$ which ends up becoming $R^4 = S \circ L \circ S$.
2. This also shows that a lower bound on the number of active S boxes over four rounds is 25.
    - 5 active super S-boxes due to the linear layer.
    - At least 5 active S boxes inside a super S-box due to MixColumns.
3. Similarly, six rounds of AES can be written as

$$R^6 = S \circ L \circ S \circ L \circ S. \tag{20}$$

Definitions of Q, Q'

For convenience, we introduce the following definition.

Definition 3

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

Definitions of Q, Q'

For convenience, we introduce the following definition.

Definition 3

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

1. Since two rounds of AES correspond to one generic SPN round, we exploit the properties of one AES round to create distinguishers for an odd number of rounds.

Definitions of Q, Q'

For convenience, we introduce the following definition.

Definition 3

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

1. Since two rounds of AES correspond to one generic SPN round, we exploit the properties of one AES round to create distinguishers for an odd number of rounds.
2. Adding another round at the end of (19), three rounds of AES can be written as $Q \circ S$.

Definitions of Q, Q'

For convenience, we introduce the following definition.

Definition 3

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

1. Since two rounds of AES correspond to one generic SPN round, we exploit the properties of one AES round to create distinguishers for an odd number of rounds.
2. Adding another round at the end of (19), three rounds of AES can be written as $Q \circ S$.
3. Similarly, five rounds of AES can be written as $S \circ L \circ S \circ Q'$.

Properties of Q, Q'

1. 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 $z_i = 0$ and $x_i = 0$ otherwise.

Properties of Q, Q'

1. 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 $z_i = 0$ and $x_i = 0$ otherwise.
2. For any state $a = (a_0, a_1, a_2, a_3)$, let

$$T_{z,a} \triangleq \{Q(a \oplus x) \mid x \in V_z\}. \tag{21}$$

Properties of Q, Q'

1. 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 $z_i = 0$ and $x_i = 0$ otherwise.
2. For any state $a = (a_0, a_1, a_2, a_3)$, let

$$T_{z,a} \triangleq \{Q(a \oplus x) \mid x \in V_z\}. \tag{21}$$

3. Note that $T_{z,a}$ depends on keyed functions. Let H_i denote the image of the i-th word in $SR(a \oplus x)$ for $x \in V_z$. Notice that $|H_i| = q^{4-t}$.

Properties of Q, Q'

1. 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 $z_i = 0$ and $x_i = 0$ otherwise.
2. For any state $a = (a_0, a_1, a_2, a_3)$, let

$$T_{z,a} \triangleq \{Q(a \oplus x) \mid x \in V_z\}. \tag{21}$$

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

$$T_i^{z,a} \triangleq SB \circ MC(H_i). \tag{22}$$

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

Properties of Q, Q'

Lemma 4

The set $T_{z,a}$ satisfies

$$T_{z,a} = T_0^{z,a} \times T_1^{z,a} \times T_2^{z,a} \times T_3^{z,a} \tag{23}$$

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

Properties of Q, Q'

Lemma 4

The set $T_{z,a}$ satisfies

$$T_{z,a} = T_0^{z,a} \times T_1^{z,a} \times T_2^{z,a} \times T_3^{z,a} \tag{23}$$

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

Proof.

Each word of $Q(a \oplus x)$ contributes one byte to each word after SR. 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.

The SimpleSWAP Algorithm

Algorithm 1 is a primitive used to perform the yoyo itself.

Algorithm 1 Swaps the first word where texts are different and returns one word.

---

function $\textsc{SimpleSWAP}(x^0, x^1)$ $\triangleright$ $x^0 \neq x^1$
  $x'^0 \leftarrow x'^1$
  for i from 0 to 3 **do**
    if $x_i^0 \neq x_i^1$ **then**
      $x_i'^0 \leftarrow x_i'^1$
      return x'^0

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.
2. Using Lemma 1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$.

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.
2. Using Lemma 1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$.
3. From Lemma 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 $T_i^{z,a}$.

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.
2. Using Lemma 1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$.
3. From Lemma 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 $T_i^{z,a}$.
4. In paritcular,

$$T'_{z,a} = \{c_0^0, c_0^1\} \times \{c_1^0, c_1^1\} \times \{c_2^0, c_2^1\} \times \{c_3^0, c_3^1\} \subset T_{z,a}. \tag{24}$$

where the size of $T'_{z,a}$ is at most 2^4 and $\{c_i^0, c_i^1\} \subset T_i^{z,a}$.

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.
2. Using Lemma 1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$.
3. From Lemma 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 $T_i^{z,a}$.
4. In paritcular,

$$T'_{z,a} = \{c_0^0, c_0^1\} \times \{c_1^0, c_1^1\} \times \{c_2^0, c_2^1\} \times \{c_3^0, c_3^1\} \subset T_{z,a}. \tag{24}$$

   where the size of $T'_{z,a}$ is at most 2^4 and $\{c_i^0, c_i^1\} \subset T_i^{z,a}$.
5. Any other $c' \neq c^0, c^1 \in T'_{z,a}$ satisfies $\nu(Q^{-1}(c') \oplus S(p^0)) = \nu(Q^{-1}(c') \oplus S(p^1)) = \nu(S(p^0) \oplus S(p^1))$.

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.
2. Using Lemma 1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$.
3. From Lemma 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 $T_i^{z,a}$.
4. In paritcular,

$$T'_{z,a} = \{c_0^0, c_0^1\} \times \{c_1^0, c_1^1\} \times \{c_2^0, c_2^1\} \times \{c_3^0, c_3^1\} \subset T_{z,a}. \tag{24}$$

   where the size of $T'_{z,a}$ is at most 2^4 and $\{c_i^0, c_i^1\} \subset T_i^{z,a}$.
5. Any other $c' \neq c^0, c^1 \in T'_{z,a}$ satisfies $\nu(Q^{-1}(c') \oplus S(p^0)) = \nu(Q^{-1}(c') \oplus S(p^1)) = \nu(S(p^0) \oplus S(p^1))$.
6. In particular, $\nu(R^{-3}(c') \oplus p^0) = \nu(R^{-3}(c') \oplus p^1) = \nu(p^0 \oplus p^1)$.

Distinguisher for Three Rounds of AES

1. Consider plaintexts p^0, p^1 such that $z = \nu(p^0 \oplus p^1)$ and $t = hw(z)$.
2. Using Lemma 1, we see that $\nu(S(p^0) \oplus S(p^1)) = \nu(p^0 \oplus p^1)$.
3. From Lemma 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 $T_i^{z,a}$.
4. In paritcular,

$$T'_{z,a} = \{c_0^0, c_0^1\} \times \{c_1^0, c_1^1\} \times \{c_2^0, c_2^1\} \times \{c_3^0, c_3^1\} \subset T_{z,a}. \tag{24}$$

where the size of $T'_{z,a}$ is at most 2^4 and $\{c_i^0, c_i^1\} \subset T_i^{z,a}$.

5. Any other $c' \neq c^0, c^1 \in T'_{z,a}$ satisfies $\nu(Q^{-1}(c') \oplus S(p^0)) = \nu(Q^{-1}(c') \oplus S(p^1)) = \nu(S(p^0) \oplus S(p^1))$.
6. In particular, $\nu(R^{-3}(c') \oplus p^0) = \nu(R^{-3}(c') \oplus p^1) = \nu(p^0 \oplus p^1)$.
7. With a random permutation, the chosen ciphertext c' would satisfy this condition with probability 2^{-96}.

Distinguisher for Three Rounds of AES

Algorithm 2 Distinguisher for Three Rounds of AES

Require: Plaintexts p^0, p^1 with $hw(\nu(p^0 \oplus p^1)) = 3$
Ensure: 1 for AES, -1 otherwise

1: $c^0 \leftarrow enc_k(p^0, 3)$, $c^1 \leftarrow enc_k(p^1, 3)$
2: $c' \leftarrow \textsc{SimpleSWAP}(c^0, c^1)$
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

Data complexity: two plaintexts and one adaptively chosen ciphertext.

Distinguisher for Four Rounds of AES

1. Four rounds of AES can be represented as $R^4 = S \circ L \circ S$ after simplification.

Distinguisher for Four Rounds of AES

1. Four rounds of AES can be represented as $R^4 = S \circ L \circ S$ after simplification.
2. Theorem 2 is invoked to create the distinguisher.

Distinguisher for Four Rounds of AES

1. Four rounds of AES can be represented as $R^4 = S \circ L \circ S$ after simplification.
2. Theorem 2 is invoked to create the distinguisher.
3. Again, the new ciphertexts are created by simply exchanging words between the two obtined ciphertexts, as shown in Algorithm 3.

Distinguisher for Four Rounds of AES

Algorithm 3 Distinguisher for Four Rounds of AES

Require: Plaintexts p^0, p^1 with $hw(\nu(p^0 \oplus p^1)) = 3$
Ensure: 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

Data complexity: two plaintexts and two adaptively chosen ciphertexts.

Distiguisher for Five Rounds of AES

1. If the difference between two plaintexts after Q' is zero in t words, we can apply the yoyo game and get new pairs that are zero in exactly the same words after Q' and reside in the same sets by Lemma 4.

Distiguisher for Five Rounds of AES

1. If the difference between two plaintexts after Q' is zero in t words, we can apply the yoyo game and get new pairs that are zero in exactly the same words after Q' and reside in the same sets by Lemma 4.
2. 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 $\left|T_i^{z,a}\right| = q^{4-3} = q$ by Lemma 4.

Distiguisher for Five Rounds of AES

1. If the difference between two plaintexts after Q' is zero in t words, we can apply the yoyo game and get new pairs that are zero in exactly the same words after Q' and reside in the same sets by Lemma 4.
2. 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 $\left|T_i^{z,a}\right| = q^{4-3} = q$ by Lemma 4.
3. A property of the MixColumns matrix can be exploited to get a tighter bound, which is stated below.

Distiguisher for Five Rounds of AES

1. If the difference between two plaintexts after Q' is zero in t words, we can apply the yoyo game and get new pairs that are zero in exactly the same words after Q' and reside in the same sets by Lemma 4.
2. 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 $\left|T_i^{z,a}\right| = q^{4-3} = q$ by Lemma 4.
3. A property of the MixColumns matrix can be exploited to get a tighter bound, which is stated below.

Lemma 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.

Summary Theorem

Theorem 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.

Summary Theorem

Theorem 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 4, words in same positions are drawn from $T_i^{z,a}$ with size q^{4-t}, thus they are equal with probability q^{t-4}. Since t words are zero in $Q'(a) \oplus Q'(b)$, each word of $SR^{-1}(Q'(a)) \oplus SR^{-1}(Q'(b))$ has t zero bytes. From Lemma 5, $4 - t$ bytes cannot be zero in each word after MC^{-1}. This is preserved through SB^{-1} and XOR with the round key. □

Building the Distiguisher

1. To build the distinguisher, we need to create enough plaintext pairs so that there will be exactly t zeros after the application of Q'.

Building the Distiguisher

1. To build the distinguisher, we need to create enough plaintext pairs so that there will be exactly t zeros after the application of Q'.
2. Notice that two equal columns remain equal on applying $MC \circ SB$.

Building the Distiguisher

1. To build the distinguisher, we need to create enough plaintext pairs so that there will be exactly t zeros after the application of Q'.
2. Notice that two equal columns remain equal on applying $MC \circ SB$.
3. 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.

Building the Distiguisher

1. To build the distinguisher, we need to create enough plaintext pairs so that there will be exactly t zeros after the application of Q'.
2. Notice that two equal columns remain equal on applying $MC \circ SB$.
3. 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.
4. This would imply that $Q'(p^0) \oplus Q'(p^1)$ has t zero words.

Building the Distiguisher

1. To build the distinguisher, we need to create enough plaintext pairs so that there will be exactly t zeros after the application of Q'.
2. Notice that two equal columns remain equal on applying $MC \circ SB$.
3. 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.
4. This would imply that $Q'(p^0) \oplus Q'(p^1)$ has t zero words.
5. 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 4.

Attack Analysis

1. 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 (where $q = 2^8$)

$$p_b(t) = \binom{4}{t} q^{-t}. \tag{25}$$

Attack Analysis

1. 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 (where $q = 2^8$)

$$p_b(t) = \binom{4}{t} q^{-t}. \tag{25}$$

2. We require $p_b(t)^{-1}$ pairs to get one such pair. To distinguish it, 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

$$4p_b(4-t) = 4 \cdot \binom{4}{t} \cdot q^{t-4} \tag{26}$$

while for a correct pair it is $4 \cdot q^{t-4}$.

Data Complexity

1. Each pair of plaintexts requires $\frac{p_b(4-t)^{-1}}{4}$ plaintext pairs using the yoyo game.

Data Complexity

1. Each pair of plaintexts requires $\frac{p_b(4-t)^{-1}}{4}$ plaintext pairs using the yoyo game.
2. The total data complexity is

$$2 \cdot (p_b(t)^{-1} \cdot (4 \cdot p_b(4-t))^{-1}) = \frac{p_b(t) \cdot p_b(4-t)^{-1}}{2}. \tag{27}$$

Data Complexity

1. Each pair of plaintexts requires $\frac{p_b(4-t)^{-1}}{4}$ plaintext pairs using the yoyo game.
2. The total data complexity is

$$2 \cdot \left(p_b(t)^{-1} \cdot (4 \cdot p_b(4-t))^{-1}\right) = \frac{p_b(t) \cdot p_b(4-t)^{-1}}{2}. \tag{27}$$

3. For $t = 2$, the data complexity is minimum at approximately $2^{25.8}$. The overall distinguisher is shown in Algorithm 4.

Distinguisher for Five Rounds of AES

Algorithm 4 Distinguisher for Five Rounds of AES

Ensure: 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

Five Round Key Recovery Yoyo on AES

1. Want to find the first round key k_0 XORed in front of R^5.

Five Round Key Recovery Yoyo on AES

1. Want to find the first round key k_0 XORed in front of R^5.
2. The MixColumns matrix M in AES is given by (for some constant $\alpha \in \mathbb{F}_{2^8}$)

$$M = \begin{pmatrix} \alpha & \alpha \oplus 1 & 1 & 1 \\ 1 & \alpha & \alpha \oplus 1 & 1 \\ 1 & 1 & \alpha & \alpha \oplus 1 \\ \alpha \oplus 1 & 1 & 1 & \alpha \end{pmatrix}. \tag{28}$$

Five Round Key Recovery Yoyo on AES

1. Want to find the first round key k_0 XORed in front of R^5.
2. The MixColumns matrix M in AES is given by (for some constant $\alpha \in \mathbb{F}_{2^8}$)

$$M = \begin{pmatrix} \alpha & \alpha \oplus 1 & 1 & 1 \\ 1 & \alpha & \alpha \oplus 1 & 1 \\ 1 & 1 & \alpha & \alpha \oplus 1 \\ \alpha \oplus 1 & 1 & 1 & \alpha \end{pmatrix}. \tag{28}$$

3. 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.

Five Round Key Recovery Yoyo on AES

1. The difference between the first words after partial encryption of the two plaintexts $MC \circ SB \circ AK$ becomes

$$\alpha b_0 \oplus (\alpha \oplus 1) b_1 = y_0 \tag{29}$$

$$b_0 \oplus \alpha b_1 = y_1 \tag{30}$$

$$b_0 \oplus b_1 = y_2 \tag{31}$$

$$(\alpha \oplus 1) b_0 \oplus b_1 = y_3 \tag{32}$$

where $b_0 = s(k_{0,0}) \oplus s(z \oplus k_{0,0})$ and $b_1 = s(k_{0,1} \oplus i) \oplus s(k_{0,1} \oplus z \oplus i)$.

Five Round Key Recovery Yoyo on AES

1. The difference between the first words after partial encryption of the two plaintexts $MC \circ SB \circ AK$ becomes

$$\alpha b_0 \oplus (\alpha \oplus 1) b_1 = y_0 \tag{29}$$

$$b_0 \oplus \alpha b_1 = y_1 \tag{30}$$

$$b_0 \oplus b_1 = y_2 \tag{31}$$

$$(\alpha \oplus 1) b_0 \oplus b_1 = y_3 \tag{32}$$

where $b_0 = s(k_{0,0}) \oplus s(z \oplus k_{0,0})$ and $b_1 = s(k_{0,1} \oplus i) \oplus s(k_{0,1} \oplus z \oplus i)$.

2. (31) can be written as

$$s(k_{0,0}) \oplus s(k_{0,0} \oplus z) \oplus s(k_{0,1} \oplus i) \oplus s(k_{0,1} \oplus z \oplus i) = y_2. \tag{33}$$

Five Round Key Recovery Yoyo on AES

1. The difference between the first words after partial encryption of the two plaintexts $MC \circ SB \circ AK$ becomes

$$\alpha b_0 \oplus (\alpha \oplus 1) b_1 = y_0 \tag{29}$$

$$b_0 \oplus \alpha b_1 = y_1 \tag{30}$$

$$b_0 \oplus b_1 = y_2 \tag{31}$$

$$(\alpha \oplus 1) b_0 \oplus b_1 = y_3 \tag{32}$$

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

$$s(k_{0,0}) \oplus s(k_{0,0} \oplus z) \oplus s(k_{0,1} \oplus i) \oplus s(k_{0,1} \oplus z \oplus i) = y_2. \tag{33}$$

3. Note 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 $i \in \mathbb{F}_q$ for which $y_2 = 0$.

Five Round Key Recovery Yoyo on AES

5. 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.

Five Round Key Recovery Yoyo on AES

5. 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.
6. 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.

Five Round Key Recovery Yoyo on AES

5. 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.
6. Prepare a set $\mathcal{P}$ of plaintexts p^0 and p^1 where $p^0_0 = (0, i, 0, 0)$ and $p^1_0 = (z, z \oplus i, 0, 0)$. Let c^0, c^1 be the respective ciphertexts.
7. 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.

Five Round Key Recovery Yoyo on AES

5. 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.
6. 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.
7. 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.
8. A correct pair will satisfy

$$B(p_0'^0 \oplus k_0) \oplus B(p_0'^1 \oplus k_0) = (z_0, z_1, 0, z_3). \tag{34}$$

Five Round Key Recovery Yoyo on AES

9. 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.

Five Round Key Recovery Yoyo on AES

9. 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.
10. This holds for all 5 pairs at random with probability $2^{-8 \cdot 5} = 2^{-40}$.

Five Round Key Recovery Yoyo on AES

9. 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.
10. This holds for all 5 pairs at random with probability $2^{-8 \cdot 5} = 2^{-40}$.
11. 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.

Attack Analysis

1. The total data complexity (plaintexts and ciphertexts) is

$$D = 2 \cdot 2^8 \cdot 5 \approx 2^{11.32}. \tag{35}$$

Attack Analysis

1. The total data complexity (plaintexts and ciphertexts) is

$$D = 2 \cdot 2^8 \cdot 5 \approx 2^{11.32}. \tag{35}$$

2. 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 (34), giving a total complexity of $2^{24} \cdot 2 \cdot 4 \cdot 5 \cdot 2^8 = 2^{37.3}$.

Attack Analysis

1. The total data complexity (plaintexts and ciphertexts) is

$$D = 2 \cdot 2^8 \cdot 5 \approx 2^{11.32}. \tag{35}$$

2. 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 (34), giving a total complexity of $2^{24} \cdot 2 \cdot 4 \cdot 5 \cdot 2^8 = 2^{37.3}$.
3. This cooresponds to approximately2^{31} 5-rounds of AES (assuming 80 S-box lookups per encryption).

Extracting the Full Subkey

1. 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.

Extracting the Full Subkey

1. 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.
2. 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 $p^1 = (b_0, 0, 0, 0)$.

Extracting the Full Subkey

1. 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.
2. 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 $p^1 = (b_0, 0, 0, 0)$.
3. However, this pair is useless in recovering the other subkeys since the last three words are equal.

Extracting the Full Subkey

1. 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.
2. 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 $p^1 = (b_0, 0, 0, 0)$.
3. However, this pair is useless in recovering the other subkeys since the last three words are equal.
4. 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.

Extracting the Full Subkey

6. To attack k_1, notice that each of the m pairs returned by the yoyo satisfy

$$B(p_1'^0 \oplus k_1) \oplus B(p_1'^1 \oplus k_1) = (0, w, 0, 0) \tag{36}$$

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.

Extracting the Full Subkey

6. To attack k_1, notice that each of the m pairs returned by the yoyo satisfy

$$B(p_1'^0 \oplus k_1) \oplus B(p_1'^1 \oplus k_1) = (0, w, 0, 0) \tag{36}$$

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.

7. Notice that (36) can be written as

$$M^{-1}(0, w, 0, 0) = w \cdot M_2^{-1} = s^4(p_1'^0 \oplus k_1) \oplus s^4(p_1'^1 \oplus k_1). \tag{37}$$

where M_i^{-1} denotes the i-th column of M^{-1}.

Extracting the Full Subkey

6. To attack k_1, notice that each of the m pairs returned by the yoyo satisfy
$$B(p_1'^0 \oplus k_1) \oplus B(p_1'^1 \oplus k_1) = (0, w, 0, 0) \tag{36}$$
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.
7. Notice that (36) can be written as
$$M^{-1}(0, w, 0, 0) = w \cdot M_2^{-1} = s^4(p_1'^0 \oplus k_1) \oplus s^4(p_1'^1 \oplus k_1). \tag{37}$$
where M_i^{-1} denotes the i-th column of M^{-1}.
8. This can be used to solve for k_1 on fixing any byte in k_1. At most $4 \cdot 2^8$ guesses are spent on getting the correct key.

Extracting the Full Subkey

6. To attack k_1, notice that each of the m pairs returned by the yoyo satisfy

$$B(p_1'^0 \oplus k_1) \oplus B(p_1'^1 \oplus k_1) = (0, w, 0, 0) \tag{36}$$

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.

7. Notice that (36) can be written as

$$M^{-1}(0, w, 0, 0) = w \cdot M_2^{-1} = s^4(p_1'^0 \oplus k_1) \oplus s^4(p_1'^1 \oplus k_1). \tag{37}$$

where M_i^{-1} denotes the i-th column of M^{-1}.

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

Recovering all Round Subkeys

1. 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.

Recovering all Round Subkeys

1. 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.
2. Since the initial pair is useless, 5 pairs are used to recover the full key.

Key Recovery Algorithm for Five Rounds of AES

Algorithm 5 Key Recovery for Five Rounds of AES

Ensure: Secret key k_0

for i from 0 to $2^8 - 1$ **do**
  $p^0 \leftarrow 0,\ p^1 \leftarrow 0$
  $p^0_0 \leftarrow (0, i, 0, 0),\ p^1_0 \leftarrow (1, 1 \oplus i, 0, 0)$
  $\mathcal{S} \leftarrow \{(p^0, p^1)\}$
  while $|\mathcal{S}| < 5$ **do**
    $c^0 \leftarrow enc_k(p^0, 5),\ c^1 \leftarrow enc_k(p^1, 5)$
    $c'^0 \leftarrow \textsc{SimpleSWAP}(c^0, c^1),\ c'^1 \leftarrow \textsc{SimpleSWAP}(c^1, c^0)$
    $p'^0 \leftarrow dec_k(c'^0, 5),\ p'^1 \leftarrow dec_k(c'^1, 5)$
    $p^0 \leftarrow \textsc{SimpleSWAP}(p'^0, p'^1),\ p^1 \leftarrow \textsc{SimpleSWAP}(p'^1, p'^0)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(p^0, p^1)\}$
  for all 2^{24} key candidates k_0 **do**
    for all $(p^0, p^1) \in \mathcal{S}$ **do**
      if $l_3(s^4(p^0_0 \oplus k_0) \oplus s^4(p^1_0 \oplus k_0)) \neq 0$ **then**
        Break and jump to next key
    return k_0