The Retracing Boomerang Attack

Gautam Singh

Indian Institute of Technology Hyderabad

April 28, 2025

Introduction

1. Broke the record for 5-round AES when it was published.

Introduction

1. Broke the record for 5-round AES when it was published.
2. Brings the attack complexity down to $2^{16.5}$ encryptions.

Introduction

1. Broke the record for 5-round AES when it was published.
2. Brings the attack complexity down to $2^{16.5}$ encryptions.
3. Uncovers a hidden relationship between boomerang attacks and two other cryptanalysis techniques: yoyo game and mixture differentials.

The Boomerang Attack

1. Typically split the encryption function as $E = E_1 \circ E_0$, with differential trails for each sub-cipher.

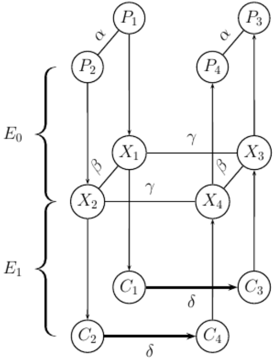

Figure 1: The boomerang attack.

The Boomerang Attack

1. Typically split the encryption function as $E = E_1 \circ E_0$, with differential trails for each sub-cipher.
2. We can build a distinguisher that can distinguish E from a truly random permutation in $\mathcal{O}((pq)^{-2})$ plaintext pairs.

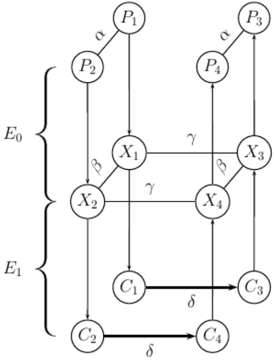

Figure 1: The boomerang attack.

The Boomerang Distinguisher

Algorithm 1 The Boomerang Attack Distinguisher

```
Initialize a counter ctr ← 0.
Generate (pq)^-2 plaintext pairs (P1, P2) such that P1 ⊕ P2 = α.
for all pairs (P1, P2) do
    Ask for the encryption of (P1, P2) to (C1, C2).
    Compute C3 = C1 ⊕ δ and C4 = C2 ⊕ δ.        ▷ δ-shift
    Ask for the decryption of (C3, C4) to (P3, P4).
    if P3 ⊕ P4 = α then
        Increment ctr
if ctr > 0 then
    return This is the cipher E
else
    return This is a random permutation
```

Boomerang Switches

1. Gain 1-2 middle rounds for free by choosing differentials carefully. Here, we discuss the *S-box switch*.

Boomerang Switches

1. Gain 1-2 middle rounds for free by choosing differentials carefully. Here, we discuss the *S-box switch*.
2. Suppose the last operation in E_0 is a layer of S-boxes where $S(\rho_1 \| \rho_2 \| \dots \| \rho_t) = (f_1(\rho_1) \| f_2(\rho_2) \| \dots \| f_t(\rho_t))$ for t independent keyed functions f_i. Suppose the difference for both β and γ corresponding to the output of some f_j is equal to Δ.

Boomerang Switches

1. Gain 1-2 middle rounds for free by choosing differentials carefully. Here, we discuss the *S-box switch*.
2. Suppose the last operation in E_0 is a layer of S-boxes where $S(\rho_1 \| \rho_2 \| \dots \| \rho_t) = (f_1(\rho_1) \| f_2(\rho_2) \| \dots \| f_t(\rho_t))$ for t independent keyed functions f_i. Suppose the difference for both β and γ corresponding to the output of some f_j is equal to Δ.
3. Denoting this part of the intermediate state by X_j,

$$(X_1)_j \oplus (X_2)_j = (X_1)_j \oplus (X_3)_j = (X_2)_j \oplus (X_4)_j = \Delta \tag{1}$$

which shows $(X_1)_j = (X_4)_j$ and $(X_2)_j = (X_3)_j$.

Boomerang Switches

1. Gain 1-2 middle rounds for free by choosing differentials carefully. Here, we discuss the *S-box switch*.
2. Suppose the last operation in E_0 is a layer of S-boxes where $S(\rho_1\|\rho_2\|\ldots\|\rho_t) = (f_1(\rho_1)\|f_2(\rho_2)\|\ldots\|f_t(\rho_t))$ for t independent keyed functions f_i. Suppose the difference for both β and γ corresponding to the output of some f_j is equal to Δ.
3. Denoting this part of the intermediate state by X_j,

$$(X_1)_j \oplus (X_2)_j = (X_1)_j \oplus (X_3)_j = (X_2)_j \oplus (X_4)_j = \Delta \tag{1}$$

   which shows $(X_1)_j = (X_4)_j$ and $(X_2)_j = (X_3)_j$.
4. If the differential characteristic in f_j^{-1} holds for (X_1, X_2), then it will hold for (X_3, X_4). *We pay for probability in one direction.*

Boomerang Switches

1. Gain 1-2 middle rounds for free by choosing differentials carefully. Here, we discuss the *S-box switch*.
2. Suppose the last operation in E_0 is a layer of S-boxes where $S(\rho_1 \| \rho_2 \| \dots \| \rho_t) = (f_1(\rho_1) \| f_2(\rho_2) \| \dots \| f_t(\rho_t))$ for t independent keyed functions f_i. Suppose the difference for both β and γ corresponding to the output of some f_j is equal to Δ.
3. Denoting this part of the intermediate state by X_j,

$$(X_1)_j \oplus (X_2)_j = (X_1)_j \oplus (X_3)_j = (X_2)_j \oplus (X_4)_j = \Delta \tag{1}$$

   which shows $(X_1)_j = (X_4)_j$ and $(X_2)_j = (X_3)_j$.
4. If the differential characteristic in f_j^{-1} holds for (X_1, X_2), then it will hold for (X_3, X_4). *We pay for probability in one direction.*
5. Distinguisher probability increases by a factor of $(q')^{-1}$, where q' is the probability of the differential characteristic in f_j.

The Yoyo Game

1. Similar to boomerang, starts by encrypting (P_1, P_2) to (C_1, C_2), then modifying them to (C_3, C_4) and decrypting them.

The Yoyo Game

1. Similar to boomerang, starts by encrypting (P_1, P_2) to (C_1, C_2), then modifying them to (C_3, C_4) and decrypting them.
2. *Unlike* the boomerang attack, this process continues in the yoyo game.

The Yoyo Game

1. Similar to boomerang, starts by encrypting (P_1, P_2) to (C_1, C_2), then modifying them to (C_3, C_4) and decrypting them.
2. *Unlike* the boomerang attack, this process continues in the yoyo game.
3. *All* pairs of intermediate values (X_{2l+1}, X_{2l+2}) satisfy some property (such as zero difference in some part).

The Yoyo Game

1. Similar to boomerang, starts by encrypting (P_1, P_2) to (C_1, C_2), then modifying them to (C_3, C_4) and decrypting them.
2. *Unlike* the boomerang attack, this process continues in the yoyo game.
3. *All* pairs of intermediate values (X_{2l+1}, X_{2l+2}) satisfy some property (such as zero difference in some part).
4. Probabilities are low with large l. Still, the yoyo technique has been used to attack AES reduced to 5 rounds.

Mixture

Definition 1 (Mixture)

Suppose $P_i \triangleq (\rho_1^i, \rho_2^i, \ldots, \rho_t^i)$. Given a plaintext pair (P_1, P_2), we say (P_3, P_4) is a *mixture counterpart* of (P_1, P_2) if for each $1 \leq j \leq t$, the quartet $(\rho_j^1, \rho_j^2, \rho_j^3, \rho_j^4)$ consists of two pairs of equal values or of four equal values. The quartet (P_1, P_2, P_3, P_4) is called a *mixture*.

Mixture

Definition 1 (Mixture)

Suppose $P_i \triangleq (\rho_1^i, \rho_2^i, \ldots, \rho_t^i)$. Given a plaintext pair (P_1, P_2), we say (P_3, P_4) is a *mixture counterpart* of (P_1, P_2) if for each $1 \leq j \leq t$, the quartet $(\rho_j^1, \rho_j^2, \rho_j^3, \rho_j^4)$ consists of two pairs of equal values or of four equal values. The quartet (P_1, P_2, P_3, P_4) is called a *mixture*.

1. If (P_1, P_2, P_3, P_4) is a mixture, then XOR of the intermediate values (X_1, X_2, X_3, X_4) is zero.

Mixture

Definition 1 (Mixture)

Suppose $P_i \triangleq (\rho_1^i, \rho_2^i, \ldots, \rho_t^i)$. Given a plaintext pair (P_1, P_2), we say (P_3, P_4) is a *mixture counterpart* of (P_1, P_2) if for each $1 \leq j \leq t$, the quartet $(\rho_j^1, \rho_j^2, \rho_j^3, \rho_j^4)$ consists of two pairs of equal values or of four equal values. The quartet (P_1, P_2, P_3, P_4) is called a *mixture*.

1. If (P_1, P_2, P_3, P_4) is a mixture, then XOR of the intermediate values (X_1, X_2, X_3, X_4) is zero.
2. $X_1 \oplus X_3 = \gamma \implies X_2 \oplus X_4 = \gamma$. Hence, for $\gamma \xrightarrow{q} \delta$ in E_1, $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$ with probability q^2.

Mixture

Definition 1 (Mixture)

Suppose $P_i \triangleq (\rho_1^i, \rho_2^i, \ldots, \rho_t^i)$. Given a plaintext pair (P_1, P_2), we say (P_3, P_4) is a *mixture counterpart* of (P_1, P_2) if for each $1 \leq j \leq t$, the quartet $(\rho_j^1, \rho_j^2, \rho_j^3, \rho_j^4)$ consists of two pairs of equal values or of four equal values. The quartet (P_1, P_2, P_3, P_4) is called a *mixture*.

1. If (P_1, P_2, P_3, P_4) is a mixture, then XOR of the intermediate values (X_1, X_2, X_3, X_4) is zero.
2. $X_1 \oplus X_3 = \gamma \implies X_2 \oplus X_4 = \gamma$. Hence, for $\gamma \xrightarrow{q} \delta$ in E_1, $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$ with probability q^2.
3. Has been applied to AES reduced up to 6 rounds. E_0 is taken to be the first 1.5 rounds of AES, which can be treated as four parallel super S-boxes.

The Retracing Boomerang Framework

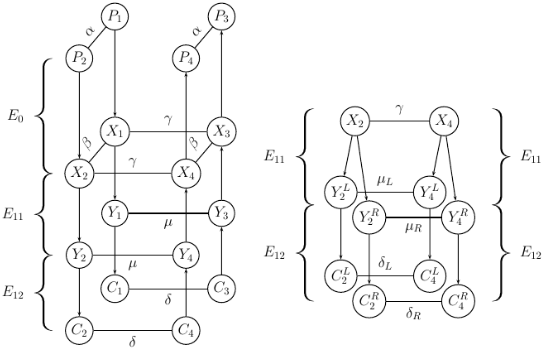

Figure 2: The retracing boomerang attack.

The Retracing Boomerang Attack

1. The *retracing boomerang* framework consists of a *shifting* type and a *mixing* type.

The Retracing Boomerang Attack

1. The *retracing boomerang* framework consists of a *shifting* type and a *mixing* type.
2. Both attacks use the setup shown in Figure 2.

The Retracing Boomerang Attack

1. The *retracing boomerang* framework consists of a *shifting* type and a *mixing* type.
2. Both attacks use the setup shown in Figure 2.
3. Although the additional split looks restrictive, it applies for a wide class of block ciphers such as SASAS constructions.

The Retracing Boomerang Attack

1. The *retracing boomerang* framework consists of a *shifting* type and a *mixing* type.
2. Both attacks use the setup shown in Figure 2.
3. Although the additional split looks restrictive, it applies for a wide class of block ciphers such as SASAS constructions.
4. Further, we assume that E_{12} can be split into two parts of size b and $n - b$ bits, call these functions E_{12}^L and E_{12}^R, with characteristic probabilities q_2^L and q_2^R respectively.

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.
2. Check if $C_1^L \oplus C_2^L = 0 \text{ or } \delta_L$. *Discard all such pairs that do not satisfy this relation.*

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.
2. Check if $C_1^L \oplus C_2^L = 0 \text{ or } \delta_L$. *Discard all such pairs that do not satisfy this relation*.
3. A δ-shift is performed on the filtered ciphertext pairs to get (C_3, C_4).

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.
2. Check if $C_1^L \oplus C_2^L = 0 \text{ or } \delta_L$. *Discard all such pairs that do not satisfy this relation*.
3. A δ-shift is performed on the filtered ciphertext pairs to get (C_3, C_4).
4. Filtering ensures that the two unordered pairs (C_1, C_3) and (C_2, C_4) are *equal*.

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.
2. Check if $C_1^L \oplus C_2^L = 0 \text{ or } \delta_L$. *Discard all such pairs that do not satisfy this relation*.
3. A δ-shift is performed on the filtered ciphertext pairs to get (C_3, C_4).
4. Filtering ensures that the two unordered pairs (C_1, C_3) and (C_2, C_4) are *equal*.
5. If one of these pairs satisfies the differential characteristic $\delta_L \xrightarrow{q_2^L} \mu_L$, *the other pair will too!*.

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.
2. Check if $C_1^L \oplus C_2^L = 0 \text{ or } \delta_L$. *Discard all such pairs that do not satisfy this relation*.
3. A δ-shift is performed on the filtered ciphertext pairs to get (C_3, C_4).
4. Filtering ensures that the two unordered pairs (C_1, C_3) and (C_2, C_4) are *equal*.
5. If one of these pairs satisfies the differential characteristic $\delta_L \xrightarrow{q_2^L} \mu_L$, *the other pair will too!*.
6. Increases the probability of the boomerang distinguisher by $(q_2^L)^{-1}$.

The Shifting Retracing Boomerang Attack

1. Adds a $(b-1)$-bit filtering in the middle of the attack procedure.
2. Check if $C_1^L \oplus C_2^L = 0 \text{ or } \delta_L$. *Discard all such pairs that do not satisfy this relation*.
3. A δ-shift is performed on the filtered ciphertext pairs to get (C_3, C_4).
4. Filtering ensures that the two unordered pairs (C_1, C_3) and (C_2, C_4) are *equal*.
5. If one of these pairs satisfies the differential characteristic $\delta_L \xrightarrow{q_2^L} \mu_L$, *the other pair will too!*.
6. Increases the probability of the boomerang distinguisher by $(q_2^L)^{-1}$.
7. Any possible characteristic of (E_{12}^L) has probability at least 2^{-b+1}, thus the overall probability increases by a factor of at most 2^{b-1}. On the other hand, filtering only leaves 2^{-b+1} of the pairs, so there is no apparent gain.

The Shifting Retracing Boomerang Attack

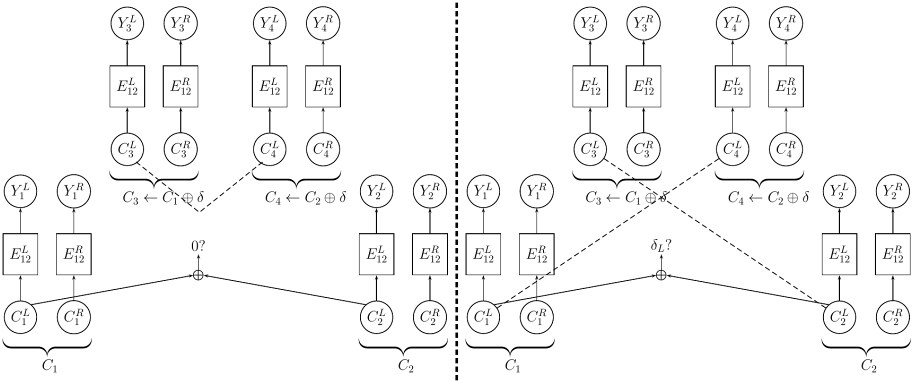

Figure 3: A shifted quartet (dashed lines indicate equality).

Advantages of Filtering

1. *Improving the signal to noise ratio.* Improving the probability by a factor of $(q_2^L)^{-1}$ improves the SNR which ensures a higher fraction of the filtered pairs on average satisfy $P_3 \oplus P_4 = \alpha$. The characteristic $\beta \xrightarrow{p} \alpha$ in the backward direction for the pair (X_3, X_4) can be replaced by a truncated differential characteristic $\beta \xrightarrow{p'} \alpha'$ of higher probability.

Advantages of Filtering

1. *Improving the signal to noise ratio.* Improving the probability by a factor of $(q_2^L)^{-1}$ improves the SNR which ensures a higher fraction of the filtered pairs on average satisfy $P_3 \oplus P_4 = \alpha$. The characteristic $\beta \xrightarrow{p} \alpha$ in the backward direction for the pair (X_3, X_4) can be replaced by a truncated differential characteristic $\beta \xrightarrow{p'} \alpha'$ of higher probability.
2. *Reducing the data complexity.* Due to the filtering, the attack leaves fewer ciphertexts. This improves the complexity in cases where more decryption queries are made.

Advantages of Filtering

1. *Improving the signal to noise ratio.* Improving the probability by a factor of $(q_2^L)^{-1}$ improves the SNR which ensures a higher fraction of the filtered pairs on average satisfy $P_3 \oplus P_4 = \alpha$. The characteristic $\beta \xrightarrow{p} \alpha$ in the backward direction for the pair (X_3, X_4) can be replaced by a truncated differential characteristic $\beta \xrightarrow{p'} \alpha'$ of higher probability.
2. *Reducing the data complexity.* Due to the filtering, the attack leaves fewer ciphertexts. This improves the complexity in cases where more decryption queries are made.
3. *Reducing the time complexity.* The filtering can also reduce the time complexity if it is dominated by the analysis of the plaintext pairs (P_3, P_4).

The Mixing Retracing Boomerang Attack

1. In the shifting attack, the attacker forces equality between the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) using a δ-shift.

The Mixing Retracing Boomerang Attack

1. In the shifting attack, the attacker forces equality between the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) using a δ-shift.
2. In this type of attack, each ciphertext pair can be shifted by $(C_1^L \oplus C_2^L, 0)$. The resulting ciphertexts are

$$C_3 = (C_3^L, C_3^R) = (C_1^L \oplus (C_1^L \oplus C_2^L), C_1^R) = (C_2^L, C_1^R), \quad (2)$$

$$C_4 = (C_4^L, C_4^R) = (C_2^L \oplus (C_1^L \oplus C_2^L), C_2^R) = (C_1^L, C_2^R). \quad (3)$$

The Mixing Retracing Boomerang Attack

1. In the shifting attack, the attacker forces equality between the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) using a δ-shift.
2. In this type of attack, each ciphertext pair can be shifted by $(C_1^L \oplus C_2^L, 0)$. The resulting ciphertexts are

$$C_3 = (C_3^L, C_3^R) = (C_1^L \oplus (C_1^L \oplus C_2^L), C_1^R) = (C_2^L, C_1^R), \tag{2}$$
$$C_4 = (C_4^L, C_4^R) = (C_2^L \oplus (C_1^L \oplus C_2^L), C_2^R) = (C_1^L, C_2^R). \tag{3}$$

3. Again, the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) are equal.

The Mixing Retracing Boomerang Attack

1. In the shifting attack, the attacker forces equality between the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) using a δ-shift.
2. In this type of attack, each ciphertext pair can be shifted by $(C_1^L \oplus C_2^L, 0)$. The resulting ciphertexts are

$$C_3 = (C_3^L, C_3^R) = (C_1^L \oplus (C_1^L \oplus C_2^L), C_1^R) = (C_2^L, C_1^R), \tag{2}$$

$$C_4 = (C_4^L, C_4^R) = (C_2^L \oplus (C_1^L \oplus C_2^L), C_2^R) = (C_1^L, C_2^R). \tag{3}$$

3. Again, the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) are equal.
4. Further, $C_1^R = C_3^R$ and $C_2^R = C_4^R$, thus we gain an *additional* factor of $(q_2^R)^{-2}$ for a total probability of $(pq_1)^2 q_2^L$, *better than shifting*!

The Mixing Retracing Boomerang Attack

1. In the shifting attack, the attacker forces equality between the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) using a δ-shift.
2. In this type of attack, each ciphertext pair can be shifted by $(C_1^L \oplus C_2^L, 0)$. The resulting ciphertexts are

$$C_3 = (C_3^L, C_3^R) = (C_1^L \oplus (C_1^L \oplus C_2^L), C_1^R) = (C_2^L, C_1^R), \tag{2}$$

$$C_4 = (C_4^L, C_4^R) = (C_2^L \oplus (C_1^L \oplus C_2^L), C_2^R) = (C_1^L, C_2^R). \tag{3}$$

3. Again, the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) are equal.
4. Further, $C_1^R = C_3^R$ and $C_2^R = C_4^R$, thus we gain an *additional* factor of $(q_2^R)^{-2}$ for a total probability of $(pq_1)^2 q_2^L$, *better than shifting*!
5. Similar to the core step used in the yoyo attack on AES.

The Mixing Retracing Boomerang Attack

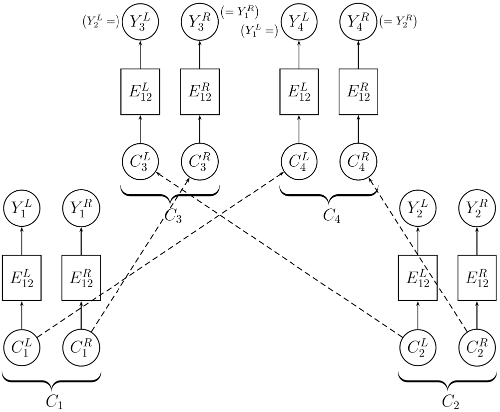

Figure 4: A mixture quartet of ciphertexts (dashed lines indicate equality).

Advantages of Shifting Retracing Attack

1. **Using structures**

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.
2. **Combination with E_{11}**

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.
2. **Combination with E_{11}**
   - In mixing, the output difference of E_{12}^{L} is arbitrary.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.
2. **Combination with E_{11}**
   - In mixing, the output difference of E_{12}^L is arbitrary.
   - Usually no good combination between characteristics of $(E_{12}^L)^{-1}$ and $(E_{11})^{-1}$. For instance, in the yoyo attack, E_{11} is empty.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.
2. **Combination with E_{11}**
   - In mixing, the output difference of E_{12}^L is arbitrary.
   - Usually no good combination between characteristics of $(E_{12}^L)^{-1}$ and $(E_{11})^{-1}$. For instance, in the yoyo attack, E_{11} is empty.
3. **Construction of 'friend pairs'**

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.
2. **Combination with E_{11}**
   - In mixing, the output difference of E_{12}^L is arbitrary.
   - Usually no good combination between characteristics of $(E_{12}^L)^{-1}$ and $(E_{11})^{-1}$. For instance, in the yoyo attack, E_{11} is empty.
3. **Construction of 'friend pairs'**
   - 'Friend pairs' are pairs which satisfy a common property.

Advantages of Shifting Retracing Attack

1. **Using structures**
   - Shifting applies the same δ-shift to all pairs of ciphertexts.
   - Filtering is applied first to reduce the data complexity.
   - Not possible in mixing: shift is based on ciphertexts, no filtering.
   - Basic boomerang attacks add a round at the top or bottom of the distinguisher. With shifting, one can obtain all ciphertexts, shift them by δ and then decrypt, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.
2. **Combination with E_{11}**
   - In mixing, the output difference of E_{12}^L is arbitrary.
   - Usually no good combination between characteristics of $(E_{12}^L)^{-1}$ and $(E_{11})^{-1}$. For instance, in the yoyo attack, E_{11} is empty.
3. **Construction of 'friend pairs'**
   - 'Friend pairs' are pairs which satisfy a common property.
   - More 'friend pairs' can be constructed in the shifting variant.

Description of AES, Notation

1. Byte ordering shown after *SB* in Figure 5 (column major).

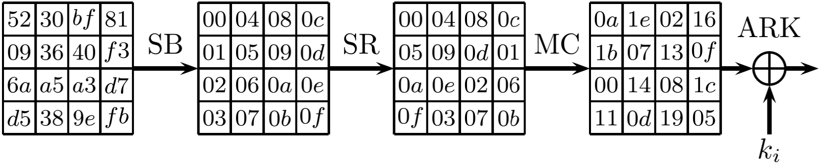

Figure 5: An AES round.

Description of AES, Notation

1. Byte ordering shown after *SB* in Figure 5 (column major).
2. j-th byte of a state X_i is denoted as $X_{i,j}$ or $(X_i)_j$.

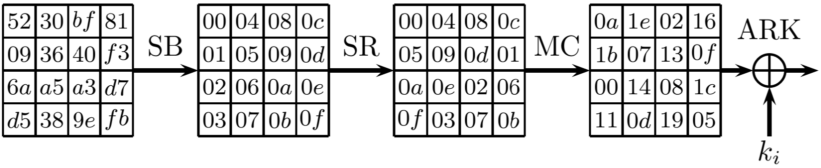

Figure 5: An AES round.

Description of AES, Notation

1. Byte ordering shown after *SB* in Figure 5 (column major).
2. j-th byte of a state X_i is denoted as $X_{i,j}$ or $(X_i)_j$.
3. Denote by W, Z and X the states before *MC* in round 0, at the input to round 1 and before *MC* in round 2 respectively.

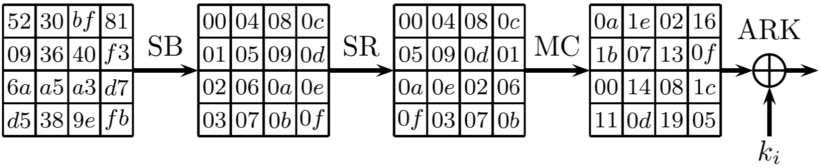

Figure 5: An AES round.

Description of AES, Notation

1. Byte ordering shown after SB in Figure 5 (column major).
2. j-th byte of a state X_i is denoted as $X_{i,j}$ or $(X_i)_j$.
3. Denote by W, Z and X the states before MC in round 0, at the input to round 1 and before MC in round 2 respectively.
4. The l-th shifted column (resp. l-th inverse shifted column) refers to application of SR (resp. SR^{-1}) to the l-th column.

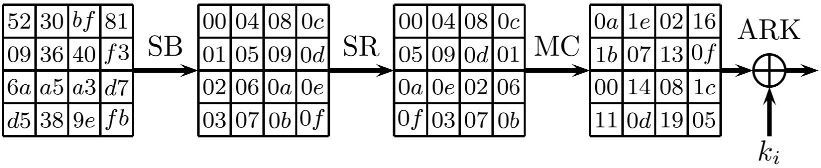

Figure 5: An AES round.

Description of AES, Notation

1. Byte ordering shown after SB in Figure 5 (column major).
2. j-th byte of a state X_i is denoted as $X_{i,j}$ or $(X_i)_j$.
3. Denote by W, Z and X the states before MC in round 0, at the input to round 1 and before MC in round 2 respectively.
4. The l-th shifted column (resp. l-th inverse shifted column) refers to application of SR (resp. SR^{-1}) to the l-th column.
5. Round subkeys are $k_{-1}, k_0, \ldots$.

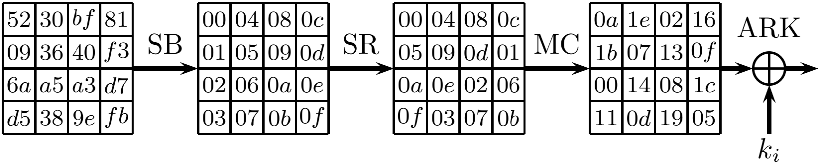

Figure 5: An AES round.

Summary of Yoyo Attack on Five Round AES

1. Decomposes AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 is the first 2.5 rounds, E_{11} is the MC of round 2 and E_{12} is the last 2 rounds.

Summary of Yoyo Attack on Five Round AES

1. Decomposes AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 is the first 2.5 rounds, E_{11} is the MC of round 2 and E_{12} is the last 2 rounds.
2. Truncated differential characteristic for E_0: zero input difference in three inverse shifted columns and zero output difference in a single shifted column with probability $4 \cdot 2^{-8} = 2^{-6}$. (*why?*)

Summary of Yoyo Attack on Five Round AES

1. Decomposes AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 is the first 2.5 rounds, E_{11} is the MC of round 2 and E_{12} is the last 2 rounds.
2. Truncated differential characteristic for E_0: zero input difference in three inverse shifted columns and zero output difference in a single shifted column with probability $4 \cdot 2^{-8} = 2^{-6}$. (*why?*)
3. For E_{12}, 1.5 rounds of AES can be taken as four 32-bit super S-boxes.

Summary of Yoyo Attack on Five Round AES

1. Decomposes AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 is the first 2.5 rounds, E_{11} is the MC of round 2 and E_{12} is the last 2 rounds.
2. Truncated differential characteristic for E_0: zero input difference in three inverse shifted columns and zero output difference in a single shifted column with probability $4 \cdot 2^{-8} = 2^{-6}$. (*why?*)
3. For E_{12}, 1.5 rounds of AES can be taken as four 32-bit super S-boxes.
4. Ciphertext pair (C_1, C_2) modified into its mixture (C_3, C_4) w.r.t. super S-boxes and decrypted. The four inputs to the S-boxes have zero XOR, thus $X_1 \oplus X_2 \oplus X_3 \oplus X_4 = 0$ since MC is linear.

Summary of Yoyo Attack on Five Round AES

1. Decomposes AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 is the first 2.5 rounds, E_{11} is the MC of round 2 and E_{12} is the last 2 rounds.
2. Truncated differential characteristic for E_0: zero input difference in three inverse shifted columns and zero output difference in a single shifted column with probability $4 \cdot 2^{-8} = 2^{-6}$. (*why?*)
3. For E_{12}, 1.5 rounds of AES can be taken as four 32-bit super S-boxes.
4. Ciphertext pair (C_1, C_2) modified into its mixture (C_3, C_4) w.r.t. super S-boxes and decrypted. The four inputs to the S-boxes have zero XOR, thus $X_1 \oplus X_2 \oplus X_3 \oplus X_4 = 0$ since MC is linear.
5. $X_3 \oplus X_4 = 0$ in a shifted column and $Z_3 \oplus Z_4 = 0$ in an inverse shifted column with probability 2^{-6}. This corresponds to one of the four quartets $(0, 5, 10, 15)$, $(1, 4, 11, 14)$, $(2, 5, 8, 13)$, $(3, 6, 9, 12)$.

Summary of Yoyo Attack on Five Round AES

1. Decomposes AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 is the first 2.5 rounds, E_{11} is the MC of round 2 and E_{12} is the last 2 rounds.
2. Truncated differential characteristic for E_0: zero input difference in three inverse shifted columns and zero output difference in a single shifted column with probability $4 \cdot 2^{-8} = 2^{-6}$. (*why?*)
3. For E_{12}, 1.5 rounds of AES can be taken as four 32-bit super S-boxes.
4. Ciphertext pair (C_1, C_2) modified into its mixture (C_3, C_4) w.r.t. super S-boxes and decrypted. The four inputs to the S-boxes have zero XOR, thus $X_1 \oplus X_2 \oplus X_3 \oplus X_4 = 0$ since MC is linear.
5. $X_3 \oplus X_4 = 0$ in a shifted column and $Z_3 \oplus Z_4 = 0$ in an inverse shifted column with probability 2^{-6}. This corresponds to one of the four quartets $(0, 5, 10, 15)$, $(1, 4, 11, 14)$, $(2, 5, 8, 13)$, $(3, 6, 9, 12)$.
6. Attack quartets of k_{-1}. Friend pairs of (Z_3, Z_4) used to get more information.

Algortihm of Yoyo Attack

Algorithm 2 Yoyo Attack on Five Round AES

Ask for the encryption of 2^6 pairs (P_1, P_2) of chosen plaintexts with non-zero difference only in bytes $0, 5, 10, 15$.

for all corresponding ciphertext pairs (C_1, C_2) **do**

  Let (C_3^j, C_4^j), $j = 1, 2, 3, 4$ be the mixture counterparts of the pair (C_1, C_2).

  Ask for the decryption of the ciphertext pairs and consider the pairs (Z_3^j, Z_4^j).

  for all $l \in \{0, 1, 2, 3\}$ **do**

    Assume all four pairs (Z_3^j, Z_4^j) and the pair (Z_1, Z_2) have zero difference in byte l.

    Use the assumption to extract bytes $0, 5, 10, 15$ of k_{-1}.

    if a contradiction is reached **then**

      Increment l

      if $l > 3$ **then** Discard the pair

    else

      Using $Z_3^j \oplus Z_4^j = 0$ in the entire l-th inverse shifted column, attack the three remaining columns of round 0 (sequentially) and decude the rest of k_{-1}.

Meet in the Middle Improvement on Yoyo Attack

The yoyo attack has data complexity about 2^9 and overall time complexity is 2^{40}. A careful analysis of round 0 can reduce the complexity down to 2^{31} encryptions. However, there is a better improvement that can be made using a meet in the middle (MITM) attack on bytes 0, 5, 10 and 15 of k_{-1}. Denote the intermediate value of byte m before the MC operation of round 0 during encryption as W_m, and consider WLOG $l = 0$. Then, the input to round 1 satisfies

$$Z_0 = 02_x \cdot W_0 \oplus 03_x \cdot W_1 \oplus 01_x \cdot W_2 \oplus 01_x \cdot W_3. \tag{4}$$

In the MITM attack, the adversary guesses bytes 0, 5 of k_{-1} by computing the values

$$02_x \cdot ((W_3^j)_0 \oplus (W_4^j)_0) \oplus 03_x \cdot ((W_3^j)_1 \oplus (W_4^j)_1) \tag{5}$$

for $j = 1, 2, 3$, concatenating these values and storing them in a table for each guess. Similarly, the adversary guesses the values for bytes 10, 15 of

k_{-1} and computes

$$01_x \cdot ((W_3^j)_2 \oplus (W_4^j)_2) \oplus 01_x \cdot ((W_3^j)_3 \oplus (W_4^j)_3) \tag{6}$$

for $j = 1, 2, 3$ and checks for a match in the table, which is equivalent to the condition $(Z_3^j)_0 = (Z_4^j)_0$ for $j = 1, 2, 3$. This 24-bit filtering leaves 2^8 candidates for bytes 0, 5, 10, 15 of k_{-1}. These can be checked by using the conditions $(Z_3^4)_0 = (Z_4^4)_0$ and $(Z_1)_0 = (Z_2)_0$.
Although the data complexity looks like 2^{16}, the *dissection technique* can be used to maintain the memory at 2^9. The time complexity is now reduced to $2^6 \cdot 4 \cdot 2^{16} = 2^{24}$ operations, which is roughly equivalent to less than 2^{23} encryptions.