CS5760: Topics in Cryptanalysis **26 March 2025**

Lecture 6: The Retracing Boomerang Attack

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

6.1 Introduction

The retracting boomerang attack broke the record for 5-round AES when it was published, bringing the attack complexity down to $2^{16.5}$ encryption/decryption operations. It uncovers a hidden relationship between boomerang attacks and two other cryptanalysis techniques, namely the yoyo game and mixture differentials.

6.2 Preliminaries

6.2.1 Boomerang Attacks

The working of a boomerang attack is shown in Figure 6.1.

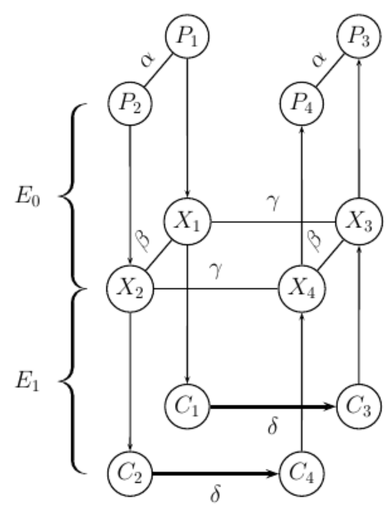

Figure 6.1: The boomerang attack.

Boomerang attacks typically split the encryption function as $E = E_1 \circ E_0$, with differential trails for each sub-cipher. Consider $\alpha \to \beta$ to be the differential characteristic in E_0 with probability p and $\gamma \to \delta$ to be the differential characteristic in E_1 with probability q. Then, one can build a distinguisher that can distinguish the full cipher E from a truly random permutation in $\mathcal{O}((pq)^{-2})$ plaintext pairs. This is shown in Algorithm 1.

Algorithm 1 The Boomerang Attack Distinguisher

Initialize a counter $ctr \leftarrow 0$.
Generate $(pq)^{-2}$ plaintext pairs (P_1, P_2) such that $P_1 \oplus P_2 = \alpha$.
for all pairs (P_1, P_2) **do**
    Ask for the encryption of (P_1, P_2) to (C_1, C_2).
    Compute $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$. $\triangleright$ *δ-shift*
    Ask for the decryption of (C_3, C_4) to (P_3, P_4).
    if $P_3 \oplus P_4 = \alpha$ **then**
        Increment ctr
if $ctr > 0$ **then**
    return This is the cipher E
else
    return This is a random permutation

6.2.2 The S-box Switch

Boomerang switches aim to gain 1-2 middle-rounds for free by choosing these differentials carefully. Here, we discuss the S-box switch.

Suppose the last operation in E_0 is a layer of S-boxes applied in parallel, and this layer S transforms $\rho = (\rho_1, \rho_2, \ldots, \rho_t)$ to $S(\rho) = (f_1(\rho_1) \| f_2(\rho_2) \| \ldots \| f_t(\rho_t))$ 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. Denoting this part of the intermediate state by X_j, using the notation of Figure 6.1 gives

$$(X_1)_j \oplus (X_2)_j = (X_1)_j \oplus (X_3)_j = (X_2)_j \oplus (X_4)_j = \Delta \tag{6.1}$$

which shows $(X_1)_j = (X_4)_j$ and $(X_2)_j = (X_3)_j$. This *S-box switch* shows that if the differential characteristic in f_j^{-1} holds for the pair (X_1, X_2), then it will hold for the pair (X_3, X_4). Thus, we pay for probability in one direction, since the equality is guaranteed to hold in the other direction. In particular, the overall probability of the distinguisher is increased by a factor of $(q')^{-1}$, where q' is the probability of the differential characteristic in f_j.

6.2.3 The Yoyo Game

Like the boomerang attack, the yoyo game starts off by encrypting a pair of plaintexts (P_1, P_2) to (C_1, C_2), then modifying them to (C_3, C_4) and decrypting them. However, unlike the boomerang attack, this process continues in the yoyo game. This process satisfies the property that *all* pairs of intermediate values (X_{2l+1}, X_{2l+2}) satisfy some property (such as zero difference in some part). However, the probability that such a property is satisfied by such a sequence is extermely low and impractical. Still, the yoyo technique has been used to attack AES reduced to 5 rounds.

6.2.4 Mixture Differentials

We begin by defining a mixture.

Definition 6.1 (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*.

From Definition 6.1, we observe that if (P_1, P_2, P_3, P_4) is a mixture, then the XOR of the intermediate values (X_1, X_2, X_3, X_4) is zero. Thus, if $X_1 \oplus X_3 = \gamma$, then $X_2 \oplus X_4 = \gamma$. Hence, for a characteristic $\gamma \xrightarrow{q} \delta$ in E_1, we see that $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$ with probability q^2.

The technique of mixture differentials has been applied to AES reduced up to 6 rounds. Usually E_0 is taken to be the first 1.5 rounds of AES, which can be treated as four parallel super S-boxes.

6.3 The Retracing Boomerang Attack

The *retracing boomerang* framework contains two attack types - a *shifting* type and a *mixing* type, both of which are explored below. Both attacks make use of the setup shown in Figure 6.2. Although the additional split $E_1 = E_{12} \circ E_{11}$ looks restrictive, it applies for a wide class of block ciphers such as SASAS constructions. 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.

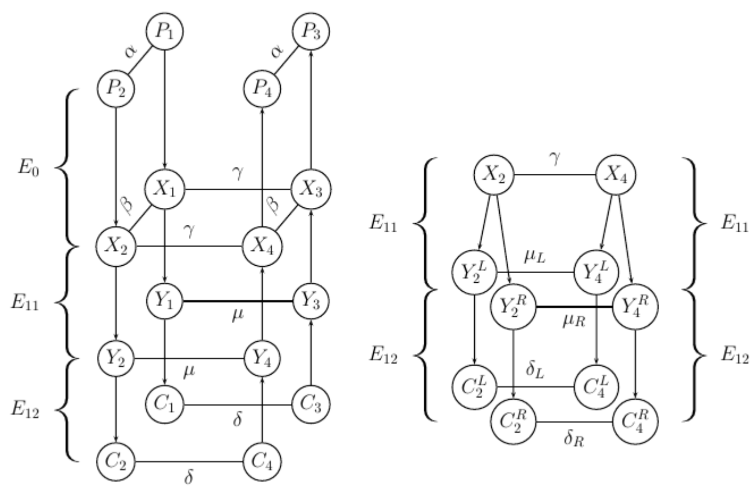

Figure 6.2: The retracing boomerang attack.

6.3.1 The Shifting Retracing Attack

Assuming that $pq_1 q_2^L q_2^R \gg 2^{-n/2}$, we can use the standard boomerang attack to build a distinguisher similar to Algorithm 1. The main idea of the shifting retracing attack is to add a $(b-1)$-bit filtering in the middle of the attack procedure, which is to check if $C_1^L \oplus C_2^L = 0$ or δ_L. We discard all such pairs that do not satisfy this relation. A δ-shift is performed on the filtered ciphertext pairs to get (C_3, C_4).

The key idea of performing this filtering is that the two unordered pairs (C_1, C_3) and (C_2, C_4) are *equal*. Thus, if one of these pairs satisfies the differential characteristic $\delta_L \xrightarrow{q_2^L} \mu_L$, the other pair will too, which increases the probability of the boomerang distinguisher by $(q_2^L)^{-1}$.

Notice that any possible characteristic of (E_{12}^L) has probability at least 2^{-b+1}, thus the overall probability of the distinguisher increases by a factor of at most 2^{b-1}. On the other hand, the filtering only leaves 2^{-b+1} of the pairs, so there is no apparent gain. However, this approach has the following advantages.

1. *Improving the signal to noise ratio.* Improving the probability by a factor of $(q_2^L)^{-1}$ improves the

signal to noise ratio which ensures a higher fraction of the filtered pairs on average satisfy $P_3 \oplus P_4 = \alpha$. Furthermore, the characteristic $\beta \xrightarrow{p} \alpha$ 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).

6.3.2 The Mixing Retracing Attack

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. Instead, in this type of attack, each ciphertext pair can be shifted by $(C_1^L \oplus C_2^L, 0)$. The resulting ciphertexts are

$$C_3 = (C_3^L, C_3^R) = (C_1^L \oplus (C_1^L \oplus C_2^L), C_1^R) = (C_2^L, C_1^R), \tag{6.2}$$
$$C_4 = (C_4^L, C_4^R) = (C_2^L \oplus (C_1^L \oplus C_2^L), C_2^R) = (C_1^L, C_2^R). \tag{6.3}$$

Again, the unordered pairs (C_1^L, C_2^L) and (C_3^L, C_4^L) are equal. Further, we have $C_1^R = C_3^R$ and $C_2^R = C_4^R$, thus we gain an additional factor of $(q_2^R)^{-2}$ for a total probability of $(pq_1)^2 q_2^L$. This mixing is also similar to the core step used in the yoyo attack on AES.

6.3.3 Comparison Between the Two Types of Retracing Attacks

Although the mixing attack has a higher probability, the shifting attack is better in various scenarios.

1. *Using structures.* In the shifting attack, the same δ-shift is applied to all pairs of ciphertexts and the filtering is applied first to reduce the data complexity. This is not possible in the mixing attack since the shift is based on the ciphertext pair and nothing is discarded.

   Typically, the basic boomerang attack is extended by adding a round at the top or bottom of the distinguisher. In such cases, the shifting attack can be used to obtain all ciphertexts, shift all of them by δ and then decrypt all of them, simulatneously checking for the filter and condition between P_3 and P_4 using a hash table.

2. *Combination with E_{11}.* In the mixing variant, the output difference of E_{12}^L is arbitrary and changes with each ciphertext pair. In most cases, there is no good combination between differential characteristics of $(E_{12}^L)^{-1}$ and $(E_{11})^{-1}$ that can be used. For instance, in the yoyo attack, E_{11} is empty.

3. *Construction of 'friend pairs'.* 'Friend pairs' are pairs that are attached to other pairs which satisfy a common property. There are many more 'friend pairs' that can be constructed in the shifting variant, making it advantageous.

6.4 Retracing Boomerang Attack on Five Round AES

The retracing boomerang attack is based on the yoyo attack, which is described first.

6.4.1 The Yoyo Attack on Five Round AES

The yoyo attack decomposes 5-round AES as $E = E_{12} \circ E_{11} \circ E_0$ where E_0 consists of the first 2.5 rounds, E_{11} is the first MC operation of round 2 and E_{12} consists of rounds 3 and 4.

For E_0 in the forward direction, the adversary uses a trucated differential characteristic whose input difference is zero in three inverse shifted columns and whose output difference is zero in a single shifted column. The probability of this characteristic is $4 \cdot 2^{-8} = 2^{-6}$, since it holds iff the output difference of the active column in round 0 is zero in at least one byte.

For E_{12} in the backward direction, notice that 1.5 rounds of AES can be taken as four 32-bit super S-boxes. For each ciphertext pair (C_1, C_2), the adversary modifies it into its mixture (C_3, C_4) with respect to the super S-boxes and asks for their decryption. Due to the mixture construction, the four inputs to the S-boxes have an XOR of zero, therefore $X_1 \oplus X_2 \oplus X_3 \oplus X_4 = 0$ as well since MC is linear. Therefore, with probability 2^{-6}, we have $X_3 \oplus X_4 = 0$ in a shifted column. Subsequently, $Z_3 \oplus Z_4 = 0$ in an inverse shifted column, which corresponds to one of the four quartets $(0, 5, 10, 15), (1, 4, 11, 14), (2, 5, 8, 13), (3, 6, 9, 12)$. This can be used to set up an attack on bytes $(0, 5, 10, 15)$ of the subkey k_{-1}. To get more information about k_{-1}, friend pairs of (Z_3, Z_4) are used.

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

$$Z_0 = 02_x \cdot W_0 \oplus 03_x \cdot W_1 \oplus 01_x \cdot W_2 \oplus 01_x \cdot W_3. \tag{6.4}$$

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

$$02_x \cdot ((W_3^j)_0 \oplus (W_4^j)_0) \oplus 03_x \cdot ((W_3^j)_1 \oplus (W_4^j)_1) \tag{6.5}$$

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

$$01_x \cdot ((W_3^j)_2 \oplus (W_4^j)_2) \oplus 01_x \cdot ((W_3^j)_3 \oplus (W_4^j)_3) \tag{6.6}$$

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.

6.4.2 Improved Attack on Five Round AES

The MITM attack can be improved to a time complexity of $2^{16.5}$ at the expense of incereasing the data complexity to 2^{15}. The main idea is to reduce the number of possible key values to 2^8 instead of 2^{16} as described earlier. This is done in multiple steps.

6.4.2.1 Specific Choice of Plaintexts

To reduce the number of possible values of $k_{-1,\{0,5\}}$, we choose plaintexts with non-zero difference only in bytes 0 and 5. For such pairs, the condition $(Z_1)_0 = (Z_2)_0$ becomes

$$02_x \cdot ((W_1)_0 \oplus (W_2)_0) \oplus 03_x \cdot ((W_1)_1 \oplus (W_2)_1). \tag{6.7}$$

This equation is now an 8-bit filtering and leaves only 2^8 candidates for $k_{-1,\{0,5\}}$.

To detect the right key bytes efficiently, an even more specific choice of plaintexts is used. We choose plaintexts (P_1, P_2) satisfying $(P_1)_5 \oplus (P_2)_5 = 01_x$. Additionally, the row of the difference distribution table (DDT) of the AES S-box corresponding to the input difference 01_x is computed and stored it in memory, where each output difference is stored along with the value(s) that lead to it. Thus, for each pair (P_1, P_2) and for each guess of $k_{-1,0}$, we can use (6.7) to compute the output difference of the SB operation in byte 5. Since the input difference in this byte is 01_x, a lookup can be used to find the inputs that can lead to this difference and retrieve possible values of $k_{-1,5}$ that correspond to the guessed $k_{-1,0}$. This gives us 2^8 values of $k_{-1,\{0,5\}}$ in about 2^8 simple operations per pair.

6.4.2.2 Eliminating Key Bytes Using Friend Pairs

To reduce the number of possible values of $k_{-1,\{10,15\}}$, the boomerang process is used to return multiple friend pairs (P_3^j, P_4^j). In particular, we choose one such pair for which

$$(P_3^j)_{10} \oplus (P_4^j)_{10} = 0 \quad \text{or} \quad (P_3^j)_{15} \oplus (P_4^j)_{15} = 0. \tag{6.8}$$

Assume WLOG that equality holds in byte 10. Then, (6.6) depends only on $k_{-1,15}$ and thus has only 2^8 possible values. This procedure requires 2^9 simple operations and leaves 2^8 suggestions for $k_{-1,\{0,5,15\}}$ since this is an additional 8-bit filtering. Finally, another friend pair can be taken and a similar MITM procedure followed to obtain the unique value of $k_{-1,\{0,5,10,15\}}$ by isolating the contribution of $k_{-1,10}$.

Therefore, the time complexity of this MITM attack is reduced to about 2^8 operations for each pair (P_1, P_2) and for each value of l. This totals to about 2^{16} operations. For each pair, we require 2^7 friend pairs to find one that satisfies (6.8) with high probability. Hence, the total data complexity is increased to about 2^{15}.

6.4.2.3 Attack Algortihm

The improved attack algorithm is described below.

1. **Precomputation:** Compute the row of the DDT of the AES S-box that corresponds to input difference 01_x, along with the actual values that lead to this difference.

2. **Online Phase:** Take 64 pairs (P_1, P_2) of plaintexts such that in each pair, $(P_1)_5 = 00_x$ and $(P_2)_5 = 01_x$, $(P_1)_0 \neq (P_2)_0$ and all other corresponding bytes are equal.

3. For each plaintext pair, create 2^7 'friend pairs' (P_1^j, P_2^j) such that for each j we have $P_1^j \oplus P_2^j = P_1 \oplus P_2$ and $(P_1^j)_{\{0,5,10,15\}} = (P_1)_{\{0,5,10,15\}}$.

4. For each plaintext pair (P_1, P_2) and for each $l \in \{0, 1, 2, 3\}$, do the following. (Here, $l = 0$ is taken for convenience.)

(a) For each guess of $k_{-1,0}$, partially encrypt (P_1, P_2) through SB in byte 0 of round 0 to find the output difference. Then, assuming (P_1, P_2) satisfies E_0, that is, $(Z_1)_0 = (Z_2)_0$, use (6.7) to find the output difference of the SB operation in byte 5 of round 0. Use the precomputed DDT to deduce inputs to the SB operation in byte 5 of round 0. Consequently, deduce the value of $k_{-1,5}$ for this guess of $k_{-1,0}$. Store all 2^8 possible values of $k_{-1,\{0,5\}}$ in a table.

(b) Ask for the encryption of (P_1, P_2) and of its friend pairs (P_1^j, P_2^j). For each ciphertext pair (C_1, C_2) or (C_1^j, C_2^j), replace it by its mixture counterparts to obtain (C_3, C_4) or (C_3^j, C_4^j). Ask for the decryption of these pairs, and let them be (P_3, P_4) and (P_3^j, P_4^j).

(c) Find a j for which (6.8) is satisfied.

(d) Perform an MITM attack on column 0 of round 0 using the corresponding pair (P_3^j, P_4^j) to obtain 2^8 possible values of $k_{-1,\{0,5,15\}}$.

(e) Perform another MITM attack on column 0 of round 0 using two plaintext pairs $(P_3^{j'}, P_4^{j'})$. For each guess of $k_{-1,10}$, compute the contribution of byte 2 to (6.7) and check for a collision. This gives a possible value for $k_{-1,\{0,5,10,15\}}$. If a contradiction is reached, move to the next value of l. If a contradiction is reached for all values of l, discard the pair (P_1, P_2) and move on to the next pair.

5. Using a pair (P_1, P_2) for which no contradiction occurred along with its 'friend pairs', perform MITM attacks on columns 1, 2 and 3 of round 0 using the fact that $Z_3 \oplus Z_4$ equals 0 in the l-th inverse shifted column to recover the entire subkey k_{-1}.

6.4.2.4 Attack Analysis

The attack succeeds if the data contains a pair that satisfies the truncated differential characteristic of E_0 and for one of the 'friend pairs' of that pair, the corresponding plaintext pair (P_3^j, P_4^j) has zero difference in either byte 10 or 15. With 64 plaintext pairs and 128 'friend pairs' per plaintext pair, each of these events occur with probability $1 - e^{-1} \approx 0.63$ giving us a probability of success of $0.63^2 = 0.4$. Increasing the number of initial pairs and friend pairs per initial pair will boost the success probability.

Another way of boosting the succees probability is to find other ways to cancel terms in (6.6). For instance, if there exist j, j' such that $\{(P_3^j)_{10}, (P_4^j)_{10}\} = \{(P_3^{j'})_{10}, (P_4^{j'})_{10}\}$, we can take the XOR of (6.6) to cancel the effect of $k_{-1,10}$, thus increasing the success probability even when there is no pair that satisfies (6.8).

The data complexity of the above attack is $2 \cdot 2^6 \cdot 2^7 = 2^{14}$ chosen plaintexts and 2^{14} adaptively chosen ciphertexts. One can use structures to reduce the data complexity to slightly above 2^{14} adaptively chosen ciphertexts and plaintexts, but this also slightly reduces the success probability due to additional dependencies between analyzed pairs.

The memory complexity of the attack remains at 2^9 128-bit memory cells, like the yoyo attack.

The time complexity is dominated by several MITM attacks that take 2^{16} operations each. Considering one AES operation to be equivalent to 80 S-box lookups and adding it to the number of queries gives us a total of $2^{16.5}$ encryptions.

6.5 Improved Attack on Five Round AES with a Secret S-box

The retracing boomerang technique can be used to devise an attack on five round AES with a secret S-box. This attack recovers the secret key without fully recovering the secret S-box (the S-box is recovered upto an affine transformation in $GF(2^8)$). The idea is to start with exploiting the fact that with probability 2^{-6}, the pair (Z_3, Z_4) has zero difference in an inverse shifted column. This observation does not depend on the specific structure of MC and SB operations, hence it can be applied to key-dependent variants as well.

6.5.1 Partial Recovery of the S-box

Assume WLOG that the retracing boomerang produces zero difference in byte 0 of state Z, that is, $(Z_3)_0 \oplus (Z_4)_0 = 0$. (6.4) can be rewritten as

$$0 = (Z_3)_0 \oplus (Z_4)_0 \tag{6.9}$$

$$= 02_x \cdot ((W_3)_0 \oplus (W_4)_0) \oplus 03_x \cdot ((W_3)_1 \oplus (W_4)_1) \oplus 01_x \cdot ((W_3)_2 \oplus (W_4)_2) \oplus 01_x \cdot ((W_3)_3 \oplus (W_4)_3). \tag{6.10}$$

Note that $(W_3)_j = SB(P_3 \oplus k_{-1,j'})$ for $j = 0, 1, 2, 3$ where $j' = SR^{-1}(j)$. Therefore, if we define $4 \cdot 256 = 1024$ variables $x_{m,j} = SB(m \oplus k_{-1,j'})$ for $m \in \mathbb{F}_q$ and $j = 0, 1, 2, 3$, then each plaintext pair P_1, P_2 which satisfies (6.10) provides a linear equation in the variables $x_{m,j}$. To obtain many pairs, we attach about 2^{10} 'friend pairs' to each of the 2^6 original pairs (P_1, P_2). For each original pair along with its 'friend pairs', we perform the mixing retracing boomerang process to obtain a linear equation in the variables $x_{m,j}$. A few more friend pairs are taken for extra filtering of the original pairs.

Since we are working with differences in (6.10), we can recover the S-box with an invertible linear transformation over $GF(2^8)$. In other words, we can only obtain functions S_0, S_1, S_2, S_3 such that

$$S_j(x) = L_0(SB(x \oplus k_{-1,j'})), \tag{6.11}$$

for some unknown linear transformation L_0. Similar linear transformations L_t will be obtained for column t.

6.5.2 Recovering the Secret Key

The secret key k_{-1} can be recovered despite not knowing the S-box in two steps.

First, for each j', we can recover $\bar{k_{j'}} = k_{-1,0} \oplus k_{-1,j'}$ as $\bar{k_{j'}}$ is the unique value of c such that $S_j(x) = S_0(x \oplus c)$ for all x. Similarly, we can recover each inverse shifted column of k_{-1} up to 2^8 possible values. This reduces the total number of candidates for k_{-1} to 2^{32}.

Second, the differences $k_{-1,0} \oplus k_{-1,j}$ for $j = 1, 2, 3$ can be found by taking several quartets of values (x_0, x_1, x_2, x_3) such that $\bigoplus_{i=0}^{3} S_0(x_i) = 0$. These quartets eliminate the effect of the difference between the linear transformations L_0 and L_j by finding the unique value of c_j such that $\bigoplus_{i=0}^{3} S_j(c_j \oplus x) = 0$. Thus, in about 2^{12} operations, we can determine the entire secret key k_{-1} upto the value of $k_{-1,0}$. These 2^8 possibilities can be exhaustively searched.

6.5.3 Attack Analysis

The data complexity of this attack is $2 \cdot 2^6 \cdot 2^{10} = 2^{17}$ chosen plaintexts and 2^{17} adaptively chosen ciphertexts. Using structures, the amount of chosen plaintexts can be reduced to 2^{14}, thus the overall data complexity is less than $2^{17.5}$ chosen plaintexts and adaptively chosen ciphertexts.

The time complexity is dominated by solving a system of 1034 equations in 1024 variables for each of the 2^6 pairs (P_1, P_2). Using an efficient algorithm such as the Method of the Four Russians, each solution takes about 2^{27} simple operations or approximately 2^{21} encryptions. Thus, the overall time complexity is 2^{29}.

The memory complexity is dominated by the memory required for solving the equations, which is about 2^{17} 128-bit blocks.

6.5.4 Improvement Using a Distinguisher Before the Attack

The equation solving step has to be applied 2^8 times since we do not know if a pair satisfies the boomerang property. To obtain this information in advance, we can use the five-round yoyo distinguisher. In this variant, the time complexity is dominated by the complexity of the yoyo distinguisher, which is $2^{25.8}$. The memory complexity is still 2^{17}.

6.6 The Retracing Rectange Attack and Mixture Differentials

A drawback of the retracing boomerang attack is that it uses the stronger adaptively chosen plaintext and ciphertext model. However, the amplified boomerang attack uses only the chosen plaintext model of attack. We first explore this attack before introducing the retracing variant.

6.6.1 The Amplified Boomerang Attack

In this version of the boomerang attack, the adversary considers *pairs of pairs* of plaintexts $((P_1, P_2), (P_3, P_4))$ such that $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$. For each such pair, the adversary then checks whether the corresponding quartet of ciphertexts $((C_1, C_2), (C_3, C_4))$ satisfy $C_1 \oplus C_2 = C_3 \oplus C_4 = \delta$.

The analysis behind this distinguisher is that the overall probability of the boomerang is p^2q^2, thus if $pq \gg 2^{-n/2}$, then a distinguisher can be created using $4 \cdot 2^{n/2}(pq)^{-1}$ chosen plaintexts. The time complexity of this attack is $\mathcal{O}(2^{n/2}(pq)^{-1})$ using hash tables.

6.6.2 The Retracing Rectangle Attack

Translating a retracing boomerang attack to a retracing rectangle attack follows the same idea as translating a (classical) boomerang attack to a rectangle attack. We start with decomposing $E = E_1 \circ E_{02} \circ E_{01}$, where E_{01} divides the state into two parts of b and $n - b$ bits. Further, suppose that there exists differential characteristics $\alpha_L \xrightarrow{p_1^L} \mu_L$ for E_{01}^L, $\alpha_R \xrightarrow{p_1^R} \mu_R$ for E_{01}^R, $\mu \xrightarrow{p_2} \beta$ for E_{02} and $\gamma \xrightarrow{q} \delta$ for E_1. This is illustrated in Figure 6.3.

A distinguisher can be built assuming $p_1^L p_1^R p_2 q \gg 2^{-n/2}$ as before. However, in the retracing rectangle attack, we consider plaintext quartets that satisfy

$$(P_1 \oplus P_2 = \alpha) \wedge (P_3 \oplus P_4 = \alpha) \wedge (P_1^L \oplus P_3^L = 0 \text{ or } \alpha_L). \tag{6.12}$$

As a result of (6.12), $\{P_1^L, P_2^L\} = \{P_3^L, P_4^L\}$. If one of them satisfies the differential characteristic of E_{10}^L, then so does the other. This improves the probability of the distinguisher by a factor of $(p_1^L)^{-1}$.

Unlike the shifting retracing boomerang attack, there is no need to filter data to obtain an improvement. Additionally, the signal to noise ratio is improved. Usually, the adversary starts with a structure $\mathcal{S}$ of plaintext pairs with input difference α. Instead of checking all $\binom{|\mathcal{S}|}{2}$ pairs, a hash table is used to check all quartets in $\mathcal{O}(|\mathcal{S}|)$ time.

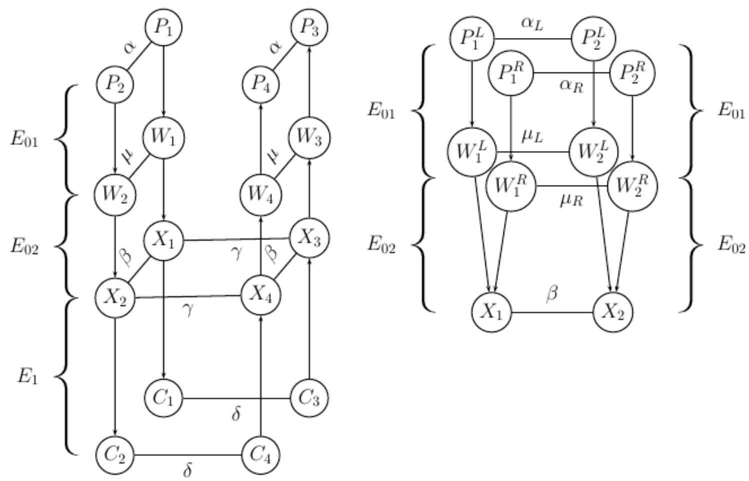

Figure 6.3: The retracing rectangle attack.

6.6.3 Mixing Variant and Relation to Mixture Differentials

Similar to the mixing retracing boomerang attack, the adversary can force, $\{P_1^L, P_2^L\} = \{P_3^L, P_4^L\}$ by choosing $P_3 = (P_2^L, P_1^R)$ and $P_4 = (P_1^L, P_2^R)$. As this choice forces $\{P_1^R, P_2^R\} = \{P_3^R, P_4^R\}$, the probability of the rectangle distinguisher is increased by a factor of $(p_1^L p_1^R)^{-1}$.