The ISC Int'l Journal of Information Security

July 2024, Volume 16, Number 2 (pp. 1–13)

http://www.isecure-journal.org

Boomerang Attacks on Reduced-Round Midori64

Mehmet Emin Gönen [1], Muhammed Said Gündoğan [1], and Kamil Otal [1,*]

[1] *TÜBİTAK BİLGEM National Research Institute of Electronics and Cryptology, Gebze, Turkey*

ARTICLE INFO.

Article history:
Received:
Revised:
Accepted:
Published Online:

Keywords:
Boomerang attack, Substitution-Permutation Network (SPN), Block Cipher, Midori, Lightweight Cryptography

Type: Research Article

doi:

dor:

ABSTRACT

Midori64 is a lightweight SPN block cipher introduced by Banik *et al.* at ASIACRYPT 2015 and it operates on 64-bit states through 16 rounds using a 128-bit key. In the last decade, Midori64 has been exposed to several attacks. In this paper, to the best of our knowledge, we provide the first boomerang attack on Midori64 in the literature. For this purpose, firstly, we present a practical single key 7-round boomerang attack on Midori64, improving the mixture idea of Biryukov by a new technique which we call "mixture pool", and then extend our attack up to 9 rounds with time complexity $2^{122.3}$, and memory and data complexity 2^{36}. (The authors of Midori stated that they expect much smaller rounds than eight rounds of Midori64 are secure against boomerang-type attacks.) We also emphasize that the mixture pool idea provides a kind of data-memory tradeoff and presents more usefulness for boomerang-type attacks.

1 Introduction

Midori is a symmetric lightweight block cipher designed by Banik *et al.* considering the optimization concerning the low energy consumption [1]. Midori has two versions: Midori64 operates on 64-bit states through 16 rounds using a 128-bit key, whereas Midori128 operates on 128-bit states through 20 rounds using a 128-bit key.

1.1 Related Works

Both Midori64 and Midori128 attracted much attention and have been examined by various attacks in the last decade. In particular, the most destructive attacks on Midori64 were given by Guo *et al.* in [2] and Todo *et al.* in [3] by showing that approximately 2^{-96} and 2^{-64} of all keys have a sort of weakness in terms of cryptanalysis. Additionally, there are also some different types of attacks on round-reduced versions of Midori64 such as meet-in-the-middle attacks up to 12 rounds given in [4], impossible differential attacks up to 11 and 12 rounds as in [5, 6], differential attacks using MILP techniques up to 11 rounds given by [7], and integral attacks using some novel invariants up to 10 rounds by [8]. Table 1 and 2 summarize all attacks together with their complexities.

1.2 Motivation

We would like to emphasize that there have been no boomerang attacks on Midori in the literature. The designers of Midori stated their expectation that much smaller rounds than 8 rounds are secure against boomerang-type attacks for Midori in [1]. In general, the single-key boomerang-type attacks on non-tweakable SPN ciphers in the literature are not longer than 7 rounds [9–11]. Therefore, we wanted to examine the potency of boomerang attacks on Midori.

* Corresponding author.
Email addresses: memin.gonen@gmail.com, muhammedsaidgundogan@gmail.com, kamil.otal@gmail.com

ISSN: 2008-2045

Attack Type	Rounds	Data	Time	Memo
Meet-in-the-middle [4]	12(16)	$2^{55.5}$	$2^{125.5}$	2^{106}
Impossible differential [5]	11(16)	2^{60}	$2^{116.59}$	$2^{92,76}$
*Impossible differential [6]	12(16)	$2^{61.87}$	$2^{90.51}$	2^{41}
Differential [7]	11(16)	$2^{61.2}$	$2^{100.3}$	-
Boomerang (Section 4)	9(16)	2^{36}	$2^{122.3}$	2^{106}

Table 1. Table of best single-key attacks on Midori64 (* means that this attack considers Midori64 without pre- and post-whitening keys)

Attack Type	WK	Rounds	Data	Time	Memo
Invariant subspace [2]	2^{32}	16(16)	2^{1}	2^{16}	-
Non-linear invariant [3]	2^{64}	16(16)	2^{1}	2^{16}	-
Non-linear invariant [8]	2^{96}	10(16)	$2^{21.32}$	2^{56}	-

Table 2. Table of best weak-key attacks on Midori64

1.3 Our Contributions

In this paper, to the best of our knowledge, we propose the first boomerang-type attack on Midori. For this purpose, we first give a 7-round attack with time complexity $2^{28.5}$ and data complexity 2^{57}. We then extend our attack to 9 rounds. The method of these attacks is based on the technique of Dunkelmann *et al.* [10] applied to the 6-round AES. Note that the technique of Dunkelmann *et al.* [10] improves Biryukov's boomerang attack [11]. Additionally, we utilize a new idea, which we call the "mixture pool". We express that the mixture pool idea provides a kind of data-memory tradeoff and hence presents more usefulness for boomerang-type attacks.

1.4 Organization of the Paper

In Section 2, we firstly briefly explain the cipher Midori64, then give the structure of boomerang attacks and mixture idea. We then present our 7-round attack in Section 3. Later, we extend this attack to 9 rounds in Section 4. Lastly, we summarize the paper in Section 5.

2 Preliminaries

In this section, we briefly introduce Midori64 and the boomerang attacks to make the paper self-contained and fix our notation at the beginning.

2.1 Midori

The states of Midori64 are denoted by 4×4 matrices whose entries are nibbles. We enumerate the nibbles of states from 0 to 15 as shown in Figure 1.

The round functions used in Midori64 are defined below.

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

Figure 1. Notation of the Nibble Enumeration

- **SubCell:** Apply the same 4-bit to 4-bit nonlinear S-box to each nibble of the state.
- **ShuffleCell:** The nibbles of the state are permuted as follows.

$$\begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix} \mapsto \begin{pmatrix} s_0 & s_{14} & s_9 & s_7 \\ s_{10} & s_4 & s_3 & s_{13} \\ s_5 & s_{11} & s_{12} & s_2 \\ s_{15} & s_1 & s_6 & s_8 \end{pmatrix}.$$

- **MixColumn:** The state is multiplied from the left by the following near MDS matrix defined over GF(16).

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

- **KeyAdd:** The state is XORed by the i^{th} round key.

The key schedule can be summarized as follows: The master key K is separated into two as $K = K_0||K_1$ in which K_i's are of 64-bit length whereas K has length 128-bit. The round keys RK_i and the whitening key WK are determined by

$$\begin{aligned} WK &= K_0 \oplus K_1, \\ RK_{2i} &= K_0 \oplus \alpha_{2i} \quad \text{for } 0 \le i \le 7, \\ RK_{2i+1} &= K_1 \oplus \alpha_{2i+1} \quad \text{for } 0 \le i \le 6, \\ RK_{15} &= K_0 \oplus K_1, \end{aligned}$$

where α_i's denote round constants.

In Midori64, we firstly apply AddKey to the plaintext using the whitening key WK, then apply all the round operations in the given order for the first 15 rounds using round keys from RK_0 to RK_{14} respectively, and finally apply only SubCell and AddKey with WK in the last round.

In this paper, for all reduced round versions of Midori64, we assume that the last round contains only SubCell and AddKey operations.

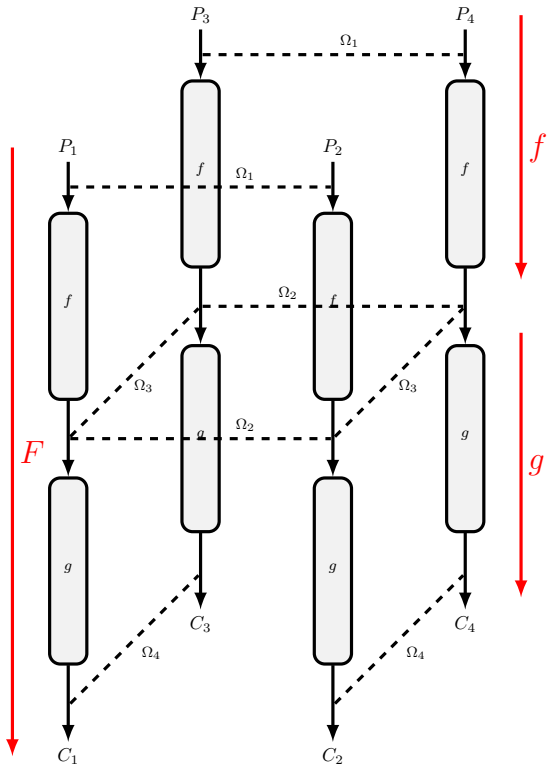

Figure 2. General idea of boomerang attacks (diagram credit [14])

2.2 Boomerang Attacks

Differential cryptanalysis, introduced by Biham and Shamir [12], uses high probability differential characteristics to attack a cipher. By a differential characteristic of a cipher, we mean a difference Ω_I of a pair of states which results a difference Ω_O after several applications of the round function of the cipher with probability p. For a block cipher with n-bit states, if there exists a characteristic $\Omega_I \to \Omega_O$ with probability $p > 2^{-n}$, then this characteristic distinguishes the cipher or several rounds of the cipher from a random permutation. By using such characteristics, key recovery attacks can be established to (the reduced-round versions of) the cipher. This type of attacks needs long differential trails because two differential characteristics cannot be concatenated unless the output of one of them equals to the input of the other one. Therefore, it has been believed that the non-existence of long differential characteristics provides security against differential attacks until 1999 when Wagner introduced the boomerang attack [13]. By using the boomerang attack technique, one can combine two arbitrary characteristics to attack a cipher.

As shown in Figure 2, let the cipher be $F = g \circ f$. Assume that we have differential characteristics $\Omega_1 \xrightarrow{f} \Omega_2$ with probability p and $\Omega_3 \xrightarrow{g} \Omega_4$ with probability q. Then, we take a pair of plaintexts (P_1, P_2) with differences $P_1 \oplus P_2 = \Omega_1$ and ask for their encryption, $C_1 = F(P_1)$ and $C_2 = F(P_2)$. Define $C_3 = C_1 \oplus \Omega_4$ and $C_4 = C_2 \oplus \Omega_4$, we ask for decryptions of C_3 and C_4, $P_3 = F^{-1}(C_3)$ and $P_4 = F^{-1}(C_4)$. We expect that

$$\begin{aligned} f(P_1) \oplus f(P_2) &= \Omega_2 \quad \text{with probability } p, \\ g^{-1}(C_1) \oplus g^{-1}(C_3) &= \Omega_3 \quad \text{with probability } q, \\ g^{-1}(C_2) \oplus g^{-1}(C_4) &= \Omega_3 \quad \text{with probability } q. \end{aligned}$$

So, with probability pq^2, we have $f(P_3) \oplus f(P_4) = \Omega_2$. Then, with probability p^2q^2, we have $P_3 \oplus P_4 = \Omega_1$. Hence, we can distinguish the cipher F from a random permutation if $p^2q^2 > 2^{-n}$ where n is the block size of the cipher.

2.3 Mixture Idea

Biryukov applied a boomerang attack on AES in [11]. The idea of "mixture differentials" has been introduced by Grassi in [15]. Later on, Dunkelman *et al.* [10] considered Biryukov's boomerang attack and improved his attacks on 5-round and 6-round AES using an idea similar to mixture differentials. In this paper, we apply the idea of the 6-round AES attack in [10] to the 7-round Midori64. Then, we improve our attack for 9 rounds using pools of mixtures to manage the data complexity.

We enumerate the columns from 0 to 3 and use terms *shuffled column* and *inverse shuffled column* to name the output of ShuffleCell and the inverse ShuffleCell operations, respectively. Table 3 shows the corresponding sets of nibbles for these terms. For example, by 0^{th} column we mean nibbles $\{0, 1, 2, 3\}$. By 0^{th} shuffled column, we mean that the new addresses of the nibbles $\{0, 1, 2, 3\}$ after ShuffleCell operations. So, 0^{th} shuffled column is the set of nibbles $\{0, 7, 9, 14\}$ because the new addresses of the nibbles 0, 1, 2 and 3 are 0, 7, 14 and 9 respectively. Similarly, by 0^{th} inverse shuffled column, we mean that the new addresses of the nibbles $\{0, 1, 2, 3\}$ after inverse ShuffleCell operations. So, 0^{th} inverse shuffled column is the set of nibbles $\{0, 5, 10, 15\}$.

Definition 1. [10, 15] Let k denote a fixed integer between 1 and 16. Assume that we have 4 plaintexts P_1, P_2, P_3, P_4 and their corresponding states D_1, D_2, D_3, D_4 before the k-th MixColumn operation. Suppose that for any $0 \le i \le 3$, the i-th columns of D_1, D_2, D_3, D_4 consist of equal values, or the i-th columns of D_1 and D_2 consist of equal values and the i-th columns of D_3 and D_4 consist of equal values [1]. Then we call D_1, D_2, D_3, D_4 as a **mixture**, or P_1, P_2, P_3, P_4 as a **mixture** at the k-th round, or (P_3, P_4) as a **mixture** of (P_1, P_2) at round k.

[1] Here, we mean 16-bit equalities by column equalities.

j	j-th column	j-th shuffled column	j-th inverse shuffled column
0	{0, 1, 2, 3}	{0, 7, 9, 14}	{0, 5, 10, 15}
1	{4, 5, 6, 7}	{2, 5, 11, 12}	{1, 4, 11, 14}
2	{8, 9, 10, 11}	{1, 6, 8, 15}	{3, 6, 9, 12}
3	{12, 13, 14, 15}	{3, 4, 10, 13}	{2, 7, 8, 13}

Table 3. Columns, images and preimages of columns under the ShuffleCell operation

Theorem 1. *Let k be a fixed integer and*

- *P_1, P_2, P_3, P_4 be four plaintexts,*
- *A_1, A_2, A_3, A_4 be their corresponding states after the $(k-1)^{th}$ MixColumn operation,*
- *D_1, D_2, D_3, D_4 be their corresponding states before the k^{th} MixColumn operation,*
- *E_1, E_2, E_3, E_4 be their corresponding states after the k^{th} MixColumn operation,*
- *Y_1, Y_2, Y_3, Y_4 be their corresponding states before the $(k+1)^{th}$ MixColumn operation.*

Then the following statements are equivalent:

i) For $0 \leq j \leq 3$, the j-th inverse shuffled columns of the states A_1, A_2, A_3, A_4 consists of four equal values or of two pairs of equal values.
ii) The states D_1, D_2, D_3, D_4 constitute a mixture.
iii) The states E_1, E_2, E_3, E_4 constitute a mixture.
iv) For $0 \leq j \leq 3$, the j-th shuffled columns of the states Y_1, Y_2, Y_3, Y_4 consist of four equal values or of two pairs of equal values.

Proof. For $i \Leftrightarrow ii$, the j^{th} inverse shuffled column of state A uniquely determines the j^{th} column of the state D. Similarly, for $iii \Leftrightarrow iv$, the j^{th} column of E uniquely determines the j^{th} shuffled column of the state Y.

For $ii \Leftrightarrow iii$, by the MixColumn operation, the j^{th} column of D uniquely determines the j^{th} column of E. □

See Figure 3 as a depiction of Theorem 1.

Corollary 1. *With the same notations of Theorem 1, let X_i denote the state before the $(k-1)^{th}$ MixColumn operation of P_i. If (P_3, P_4) is a mixture of (P_1, P_2) at round k, then $X_3 \oplus X_4 = X_1 \oplus X_2$.*

Proof. By Theorem 1, we have $A_1 \oplus A_2 \oplus A_3 \oplus A_4 = 0$. Since MixColumn is linear, this equality is still valid along the MixColumn operation. Hence, $X_1 \oplus X_2 \oplus X_3 \oplus X_4 = 0$. □

3 A 7-Round Boomerang Attack on Midori64

In this section, we give a 7-round boomerang attack in which we use the following ideas:

- The mixture idea in [11, 16],
- Extension of the mixture idea applied on 6-round AES in [11],
- The meet-in-the-middle idea in [10] to reduce the time complexities in [11],
- The mixture pool idea to reduce the data complexity.

Our attack decomposes the 7-round Midori64 as $f = f_3 \circ f_2 \circ f_1$. Here, f_1 is the first 3.5 round (up to the 4^{th} ShuffleCell) of the cipher, f_2 is the operations from the 4^{th} MixColumn operation to the 6^{th} ShuffleCell operation and f_3 is the operations from the 6^{th} MixColumn operation to the end of the cipher.

3.1 Characteristics

We use the following two conditions for the chosen-plaintexts in our characteristic.

Condition 1. *After the first MixColumn operation, we have zero differences at all the nibbles of the state except nibbles $\{4, 8\}$.*

Condition 2. *After the 6^{th} ShuffleCell operation (the state is shown by Y in Figure 4), we have zero differences at nibbles $\{0, 7, 9, 14\}$ of the pair.*

For a plaintext pair having differences only in nibbles $\{1, 3, 4, 6, 11, 12\}$, Condition 1 has probability 2^{-16} and Condition 2 has probability 2^{-16}. In total, our conditions hold with probability 2^{-32}. To expect a right pair, the attack requires 2^{32} chosen-plaintext pairs having differences only in nibbles $\{1, 3, 4, 6, 11, 12\}$.

As shown in Figure 4, Condition 1 leads to a differential state having zero differences in nibbles $\{0, 7, 9, 14\}$ at the state X which is before the fourth MixColumn operation.

3.2 Generating Mixtures

We assume we know nibbles 1, 2, 3 of the "output" whitening key. Then, given a plaintext P_1 with ciphertext C_1, we can calculate nibbles $1, 2, 3$ of the state at the beginning of round 7.

For $1 \leq j \leq 15$, we define C_3^j so that $Y_1 \oplus Y_3$ has a difference δ_j where Y_1 and Y_3 are the states before

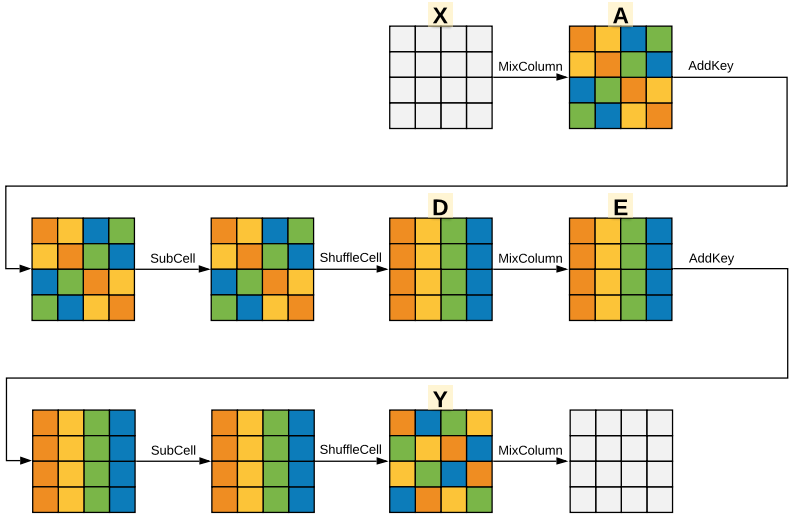

Figure 3. The mixture idea for Midori64

the last MixColumn operation of C_1 and C_3, and δ_j is a 64-bit (16 nibbles) value such that it has j in the 0^{th} nibble and 0 in the other nibbles.

From the equation $Y_1 \oplus Y_3 = \delta_j$, we can calculate their difference at the beginning of the seventh round, then we can calculate C_3^j for all $1 \leq j \leq 15$, by using nibbles $\{1, 2, 3\}$ of the whitening key.

Now, we describe the role of these adaptively chosen ciphertexts. Let (P_1, P_2) be a pair of chosen-plaintexts that satisfy Condition 2 and assume that for $1 \leq j \leq 15$ the ciphertexts C_3^j and C_4^j generated by P_1 and P_2 as in the way described above. Now, we fix a $1 \leq j \leq 15$ and consider the quadruple (Y_1, Y_2, Y_3^j, Y_4^j). Since $Y_1 \oplus Y_3^j = Y_2 \oplus Y_4^j = \delta_j$ and δ_j has a difference only in nibble 0, the first, second and third shuffled columns of the Y_1 and Y_3^j are equal. Similarly, the first, second and third shuffled columns of the Y_2 and Y_4^j are equal. Also, by Condition 2 the 0^{th} shuffled columns of Y_1 and Y_2 are equal. Since $Y_1 \oplus Y_3^j = Y_2 \oplus Y_4^j = \delta_j$, we get that the 0^{th} shuffled columns of Y_3^j and Y_4^j are equal. Hence, by Theorem 1, (P_3^j, P_4^j) is a mixture of (P_1, P_2) at round 5. Then, by Corollary 1, we have $X_3^j \oplus X_4^j = X_1 \oplus X_2$. The backward journey where the mixtures recover the key is explained in Section 3.4.

3.3 Mixture Pool

The attack needs $2^{16.5}$ chosen-plaintexts and some adaptively chosen ciphertexts. For each guess on 28-bits of the key and for each remaining pairs in Step 3 of the attack given in Section 3.5, we need 15 pairs of mixtures. In this way, it seems that we need a data complexity of $2^{28} \cdot 2^{16.5} \cdot 15 \approx 2^{48.5}$. To reduce the data complexity, we use the mixture pool technique which is described as follows:

(1) We ask for encryption of $2^{16.5}$ chosen-plaintexts.
(2) For any chosen-plaintext P_1 with ciphertext C_1, apply the following steps.
   (a) Calculate all the ciphertexts that differ from C_1 only in nibbles $\{1, 2, 3\}$ and ask for their decryption.
   (b) Save all the data.

Here, in Step 2, we calculate $2^{12} - 1$ ciphertexts differ than C_1. In Step 2.a, we ask for decryptions of $2^{12} - 1$ ciphertexts per chosen-plaintexts. So, in the mixture pool, we have $2^{16.5} \cdot (2^{12} - 1)$ adaptively chosen ciphertexts and plaintexts. In total, our data complexity is $2^{16.5} + 2^{16.5} \cdot (2^{12} - 1) = 2^{28.5}$ chosen-plaintexts and adaptively chosen ciphertexts. So, we reduce the data complexity from $2^{48.5}$ to $2^{28.5}$. However, the memory complexities increased by $2^{28.5}$ plaintexts and ciphertexts to save the mixture pool.

3.4 Retracing Mixtures

In this section, we describe how to recover some key-bits with a right pair of chosen-plaintexts and its mixtures. Assume that we know nibbles $\{1, 2, 3, 6, 11\}$ of the whitening key and the chosen-plaintext pair (P_1, P_2) satisfies both Condition 1 and Condition 2. So, we generate adaptively chosen ciphertexts called mixtures as described in Section 3.2. Then, $X_3^j \oplus X_4^j$ has zero difference at the 0^{th} shuffled column for any $1 \leq j \leq 15$. Thus, we have zero difference at the 0^{th} inverse shuffled column of the Z state (see Figure 5). Let a_j, b_j and c_j be the first, second and third nibbles of the W state. Since $a_j \oplus b_j \oplus c_j$ is the 0^{th} nibble of the Z state, we have $a_j \oplus b_j = c_j$ for all $1 \leq j \leq 15$. Since we know nibbles $\{1, 2, 3, 6, 11\}$ of the whitening key, for the value $a_j \oplus b_j \oplus c_j$, we

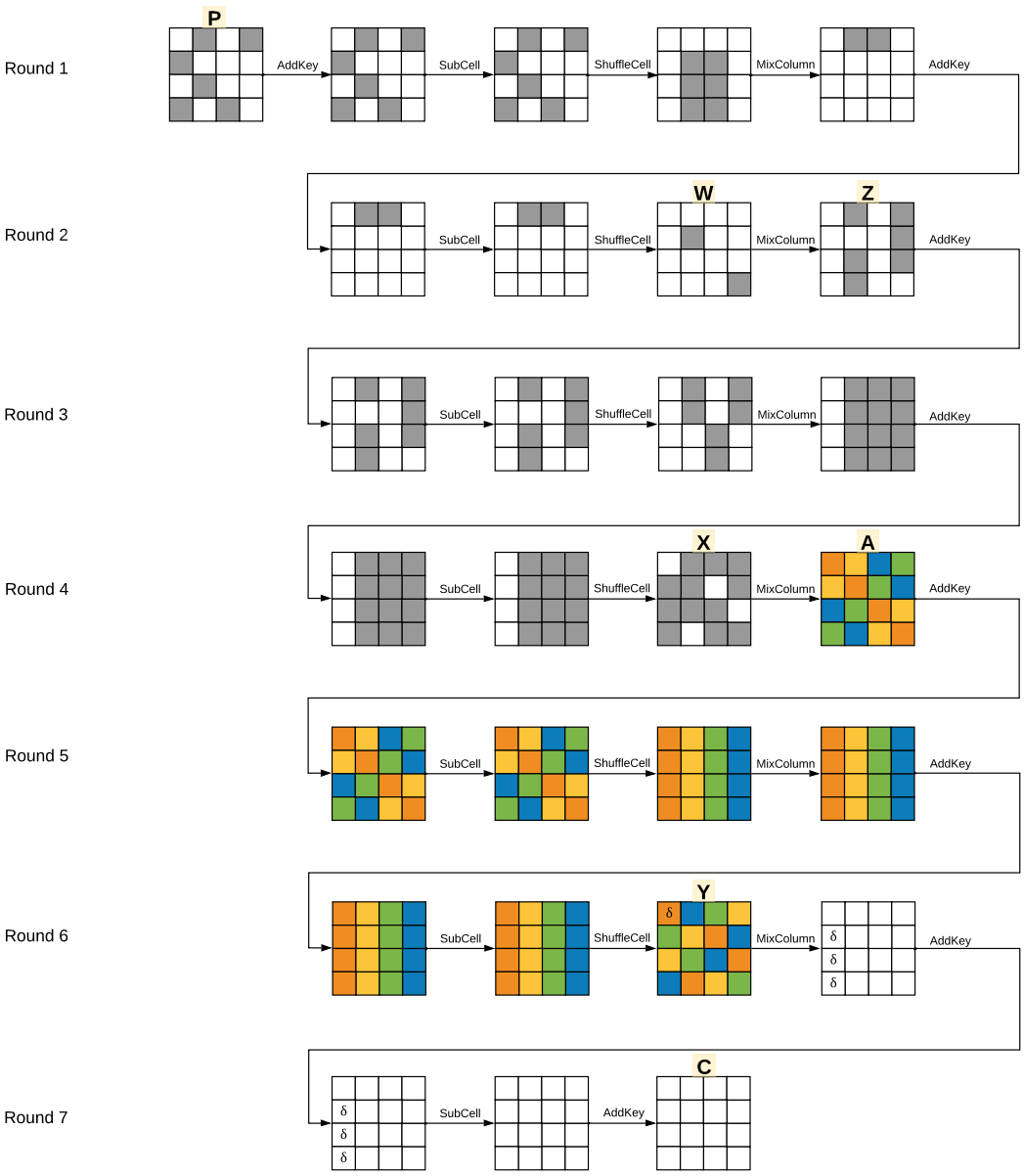

Figure 4. A 7-Round boomerang attack

need 7 more key nibbles ($\{7, 9, 13, 14\}$ of the whitening key and $\{5, 10, 15\}$ of K_1). If we exhaustively search all these 7 nibbles, it leads to a complexity of about 2^{28}. To reduce the time complexity, we use the meet-in-the-middle (MITM) procedure as described in [10]: The value c_j can be calculated by guessing nibbles $\{7, 13\}$ of the whitening key and the nibble 15 of K_0. For any guess of these 12-bits, we calculate $c_1||c_2||\cdots||c_{15}$. So we have a list of 60-bit elements with length 2^{12}. Similarly, for any guess of nibbles $\{9, 14\}$ of the whitening key and nibbles $\{5, 10\}$ of K_0, we calculate $a_1 \oplus b_1||a_2 \oplus b_2||\cdots||a_{15} \oplus b_{15}$. Here, we have a list of 60-bit elements of length 2^{16}. We sort these two lists and search them for a collision. If there is a collision, we say that the pair (P_1, P_2) suggests a 28-bit key (In fact, we will say it suggests a 56-bit key considering the 28-bits guessed in the 3rd step of the attack).

To create the first list, for each of the guesses on 12-bits, for each of the 15 pairs, we need to calculate one nibble at the end of round 1 and we operate this nibble along with one KeyAdd and one SubCell operation (which is less than a 1/4 round of Midori). By considering 1/28 encryptions instead of 1/4 rounds, the time complexity can be estimated as $2^{12} \cdot 15 \cdot 2 \cdot \frac{1}{28} \approx 2^{12}$. Similarly, to create the second list, the time complexity can be estimated as

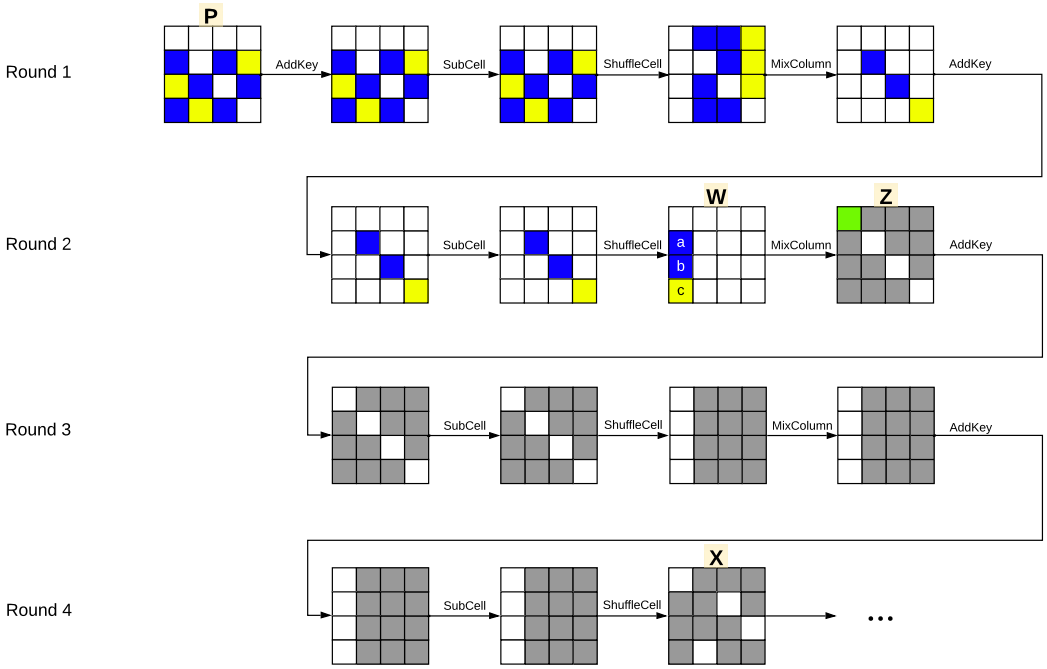

Figure 5. The retracing process of the 7-round attack

$$2^{16} \cdot 15 \cdot 2 \cdot \frac{1}{14} \approx 2^{17}$$

encryptions. So, we reduce the time complexity from around 2^{28} to 2^{17}.

In this process, we check 60-bits of collision for 28 bits of possible guess. Then, a random pair with a random guess on 28-bits of the whitening key passes this process and suggests additional 28-bit keys with probability $2^{28}/2^{60} = 2^{-32}$.

3.5 Attack Steps

We apply the following steps in our attack:

(1) We define a *structure* S to be the set of states whose nibbles $\{1, 2, 3, 4, 6, 11, 12\}$ are active and the other nibbles are passive (constant). We uniformly generate a subset of the structure S of plaintexts of size $2^{16.5}$ and ask for encryption of all these plaintexts. We have $\binom{2^{16.5}}{2} \approx 2^{32}$ pairs of plaintexts and expect that one of them satisfies both Condition 1 and Condition 2. We query for encryption of all the $2^{16.5}$ chosen-plaintexts.

(2) From the ciphertexts of the chosen-plaintexts generated in the previous step, we create the ciphertexts of the mixture pool as described in Section 3.3. We ask for decryptions of all the ciphertexts that we generated for the mixture pool.

(3) We guess the whitening key nibbles $\{1, 2, 3, 4, 6, 11, 12\}$. For all the 28-bit values, we follow the steps:

(a) We eliminate some pairs out of 2^{32} pairs using the guessed 28-bits of the whitening key:

(i) We check Condition 1, hence eliminate to 2^{-16} of 2^{32} pairs.

(ii) We calculate the differences of the 0^{th} nibbles of the state Y and eliminate the pairs having nonzero differences. This eliminates to 2^{-4} of them, we are left with 2^{12} pairs out of 2^{16} pairs.

(b) To apply the retracing process, we need a pair of plaintexts (P_1, P_2) with its pairs of mixtures (P_3^j, P_4^j) for $1 \leq j \leq 15$. As explained in Section 3.2, we calculate the ciphertexts of these mixtures and we look up the mixture pool to find their plaintexts. We assume that our guess on 28-bits of the whitening key is true and the plaintext pair (P_1, P_2) satisfies Condition 1 and Condition 2, and apply the process described in Section 3.4. Since a wrong pair passes this process and gives 28 suggested key bits with probability 2^{-32}, we left with $2^{12} \cdot 2^{-32} = 2^{-20}$ wrong pairs with suggested 28-bits of the key. Completing the loop over the guess on 28-bits of the key, we left with $2^{28} \cdot 2^{-20} = 2^8$ pairs so

that each pair suggests $28 + 28 = 56$-bits of the key.

(4) For the remaining pairs, we use the fact that the 0^{th} inverse shuffled columns of the state Z of a right pair is zero. We use the zeros at nibbles $\{5, 10, 15\}$ to apply the retracing the mixtures process (recall that we used zero difference at nibble of state Z in Step 3). Each zero gives 15 equations of 4-bits. In other words, a retracing process yields 60-bit information about the master key for each of nibbles $\{5, 10, 15\}$ of state Z. So a wrong pair passes this step with probability $(2^{-60})^3 = 2^{-180}$. At this step, we expect that one right pair left with a true suggestion on some bits of the master key. The 0^{th}, 5^{th}, 10^{th} and 15^{th} nibbles of state Z is determined by all the nibbles of the whitening key and nibbles $\{1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15\}$ of K_0. Then the four retracing mixture processes applied on the right pair give 112 bits of the master key.

(5) Since we know 112 bits of the master key, we exhaustively search the rest 16-bits of the master key.

3.6 Complexities

Data complexity is $2^{28.5}$, since the attack requires $2^{16.5} \cdot 2^{12} = 2^{28.5}$ chosen-plaintexts and adaptively chosen ciphertexts.

Time complexity is 2^{57}: The time complexity of the first two steps does not have any notable time complexity. In Step 3.a.i, we need to check the equality of the nibbles $\{5, 6, 7, 9, 10, 11\}$ at the end of round 1 where we need 24-bit key guess and $2 \cdot 6/16$ rounds encryption per pair. So, the time complexity of this step is $2^{24} \cdot 2 \cdot 2^{32} \cdot 1/7 \cdot 6/16 \approx 2^{53}$. In Step 3.a.ii, we need to calculate the difference of the nibble 0 at state Y where we need $2 \cdot 3/16$ rounds encryption per pair for all 28-bit guesses. So, the time complexity of this step is $2^{28} \cdot 2 \cdot 2^{16} \cdot 1/7 \cdot 3/16 \approx 2^{40}$. The time complexity of retracing mixtures is about 2^{17} for each pair, for each guess on 28 bits of the key. So, the time complexity of Step 3.b of the attack is about $2^{28} \cdot 2^{12} \cdot 2^{17} = 2^{57}$ encryptions. For Step 4, we apply the retracing mixtures for 2^4 pairs at most three times. This step requires at most $2^4 \cdot 3 \cdot 2^{21} = 3 \cdot 2^{25}$ encryptions which is negligible compared to 2^{57} encryptions. Step 5 has a time complexity of 2^{16} encryptions which is also negligible.

Memory Complexity is $2^{28.5}$: The mixture pool needs a memory of $2^{28.5}$ pairs of plaintexts and ciphertexts which is equal to $2^{28.5} \cdot (64 + 64) = 2^{35.5}$ bits of memory. Fix a guess on 28 bits of the whitening key. We use a memory for 2^{12} pairs after the elimination process. For each pair, we apply the retracing mixtures process for which we need $2^{16} + 2^{12}$ 60-bit blocks. After the retracing process, we delete these lists and keep the pair with its suggested key bits if the pair passes the process. For the lists, we need $(2^{16} + 2^{12}) \cdot 60 < 2^{22}$ bits of memory which is negligible compared to the memory allocated for the mixture pool. In Step 4, we expect 2^8 pairs with the suggested 56 bits of the key. This requires a memory of $2^8 \cdot (64 + 64 + 56)$ bits, which is also negligible.

4 A 9-Round Boomerang Attack on Midori64

In this section, we give a 9-round boomerang attack using similar ideas as in Section 3 but exploiting a different characteristic.

Our attack decomposes 9-round Midori64 as $f = f_3 \circ f_2 \circ f_1$. Here, f_1 is the first 4.5 round of the cipher, f_2 is the operations from the 5^{th} MixColumn operation to the 7^{th} ShuffleCell operation and f_3 is the operations from the 7^{th} MixColumn operation to the end of the cipher.

We note that we extend the mixture attack forward for two rounds. Recall that Biryukov's idea in [11] extends the mixture attack forward for one round.

4.1 Characteristics

We use the following two conditions for the chosen-ciphertexts in our characteristic.

Condition 3. *After the second MixColumn operation, we have zero difference at all the nibbles of the state except nibbles* $\{3, 7\}$.

Condition 4. *After the ShuffleCell operation in round 7 (the state is called Y as shown in Figure 6), we have zero difference at nibbles* $\{0, 7, 9, 14\}$.

For a ciphertext pair, Condition 3 has probability of 2^{-56} and Condition 4 has probability of 2^{-16}. In total, our conditions hold with probability 2^{-72}.

Since a pair of ciphertexts is a right pair with probability 2^{-72}, if we have 2^{36} data, we have a have a right pair with probability

$$1 - (1 - 2^{-72})^{2^{71}} \approx 0.393.$$

As shown in Figure 6, Condition 3 leads to a differential state having zero difference only in nibbles $\{1, 6, 8, 15\}$ at state X in round 5.

4.2 Mixture Pool

For the mixture pool, we need all the 2^{36} possible data for every ciphertext (see Figure 6). In the 7-round attack, we needed a data of size 2^{12} times of the data of chosen-plaintexts (see Figure 4). To reduce the data complexity, unlike the 7-round attack, we

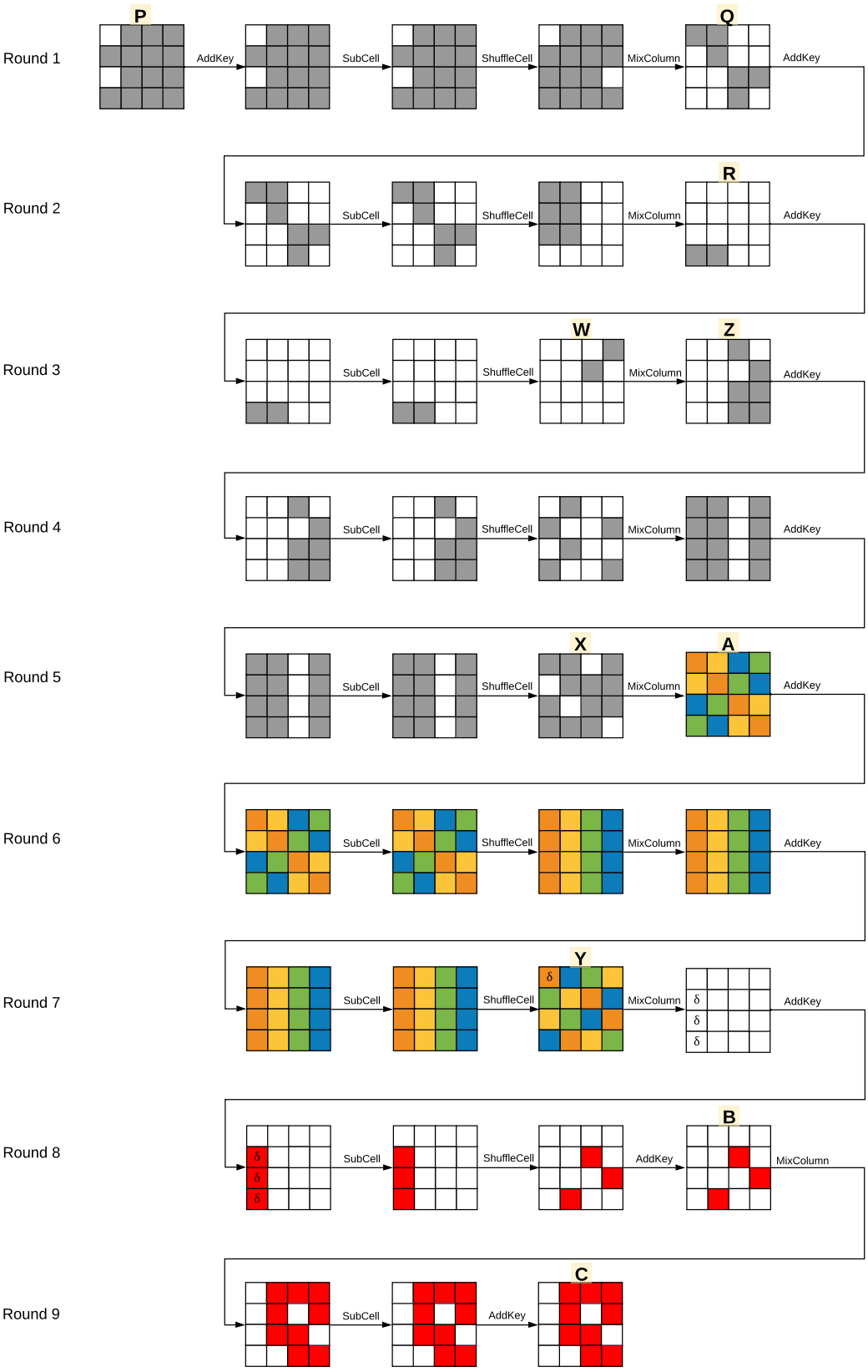

Figure 6. A 9-round boomerang attack

construct here a chosen ciphertext attack. So, we choose one structure of chosen ciphertexts having constant values on nibbles $\{0, 1, 2, 3, 7, 9, 14\}$ and all possible values on nibbles $\{4, 5, 6, 8, 10, 11, 12, 13, 15\}$. So the structure has 2^{36} ciphertexts. In this attack, the mixture pool is the same as the chosen data.

4.3 Generating Mixtures

First, we remark that we can change the order of MixColumn and AddKey operations by using the key $MC(K)$ instead of K. Assume that we know the whitening key and nibbles $\{7, 9, 14\}$ of $MC(K_1)$. Then, given a plaintext P_1 with ciphertext C_1, we can calculate nibbles $\{1, 2, 3\}$ of the state at the beginning of round 8 as shown in Figure 6.

For $1 \leq j \leq 15$, we define C_3^j so that $Y_1 \oplus Y_3^j$ has a difference δ_j where Y_1 and Y_3^j are the states before the 7^{th} MixColumn operation of C_1 and C_3^j, and δ_j is a 64-bit (16 nibbles) value such that it has j in the 0^{th} nibble and 0 in the other nibbles.

We can calculate their difference at the beginning of the seventh round from the equation $Y_1 \oplus Y_3^j = \delta_j$. Then, we can calculate C_3^j for all $1 \leq j \leq 15$ by using nibbles $\{1, 2, 3\}$ of the whitening key.

Similar to the 7-round attack, by assuming Condition 4, we use Theorem 1 and Corollary 1 to conclude that $X_3^j \oplus X_4^j = X_1 \oplus X_2$. The backward journey continues in the following subsection of the attack.

4.4 Retracing Mixtures

In this subsection, we describe how to recover some key bits with a right pair of chosen-plaintext and its mixtures. Assume that a chosen-ciphertext pair (P_1, P_2) satisfies both Condition 3 and Condition 4. So, we generate adaptively chosen ciphertexts called mixtures as in Section 4.3. Then, for any $1 \leq j \leq 15$, the 2^{nd} shuffled column of the difference $X_3^j \oplus X_4^j$ is zero. Hence, we have zero difference at the 2^{nd} inverse shuffled column at state Z as shown in Figure 7. Let a_j, b_j and c_j be the 4^{th}, 5^{th} and 7^{th} nibbles of state W, respectively, as shown in Figure 7. We have $a_j = b_j \oplus c_j$ for all $1 \leq j \leq 15$, since $a_j \oplus b_j \oplus c_j$ is the 6^{th} nibble of state Z as shown in Figure 7. We know the whitening key and nibbles $\{0, 4, 5, 6, 8, 10, 11, 14\}$ of K_0, so there are 4 key nibbles remained to calculate the values $a_j \oplus b_j \oplus c_j$. To reduce the time complexity, we use the technique described in [10] as we used in the 7-round attack in Section 3. As shown in Figure 7, 4 out of 8 key nibbles (colored blue in Figure 7) are needed to calculate a_i and the other 4 of them (colored yellow in Figure 7) are needed to calculate $b_i \oplus c_i$. We create a list of 60-bit elements of length 2^8 in which each element represents all the 4-bit differences at a_i for 15 mixtures for any guess on 4 key nibbles (determining blue nibbles). Similarly, we have a second list for $b_i \oplus c_i$ of the same size.

To create the first list, for each of the guesses on 8-bits, for each of the 15 pairs, we need to calculate one nibble at the end of round 2, and we follow this nibble along with one KeyAdd and one SubCell operation (which is less than a 3/2 round of Midori). By considering 1/6 encryptions instead of 3/2 rounds, the time complexity can be estimated as $2^8 \cdot 15 \cdot 2 \cdot \frac{1}{6} \approx 2^{10.3}$. Similarly, to create the second list, the time complexity can be estimated as $2^{10.3}$ encryptions. In total, this process has time complexity $2^{11.3}$ for each chosen-plaintext pair and for each 100-bits of guess.

In this process, we check a 60-bit collision for 16-bits of possible guess. Then a random pair with a wrong guess on 100 bits of the key passes this process and suggests 16-bit keys with probability $2^{16}/2^{60} = 2^{-44}$.

4.5 Attack Steps

The attack steps are as follows:

(1) We generate 2^{36} ciphertexts having constant values on nibbles $\{0, 1, 2, 3, 7, 9, 14\}$. Request to decrypt all the data. Now we have $\binom{2^{36}}{2} \approx 2^{71}$ pairs of chosen ciphertexts. For any guess on 64-bit WK do the following steps.
(2) Since we guessed WK, we encrypt all the plaintexts for one round and decrypt all the ciphertexts for one round. Now we get states Q and B of all the data (see Figure 6).
(3) We eliminate the pairs to pairs having zero differences on nibbles $\{1, 2, 3, 6, 7, 8, 9, 12, 13, 15\}$. So we left with $2^{71} \cdot 2^{-40} = 2^{31}$ pairs. (Note that, we do not need to work on 2^{71} pairs, we can sort the list of all 2^{36} Q-states according to the mentioned nibbles. Then, we can get the required pairs.)
(4) We guess nibbles $\{0, 4, 5, 10, 11, 14\}$ of K_0. For any 24-bit guess, we do the following steps.
(5) For all the remaining pairs, we calculate the differences at state R and eliminate pairs to pairs having zero differences at all nibbles except $\{3, 7\}$ at state R. So we left with $2^{31} \cdot 2^{-16} = 2^{15}$ pairs.
(6) We guess nibbles $\{6, 8\}$ of K_0 and nibble 14 of $MC(K_0)$. This guess also gives nibbles 7, 9 of $MC(K_0)$. For any 12-bit guess, we do the following steps.
(7) We calculate the difference of nibble 0 of state Y and eliminate pairs to pairs have zero differences at nibble 0 of state Y. We left with $2^{15} \cdot 2^{-4} = 2^{11}$ pairs.

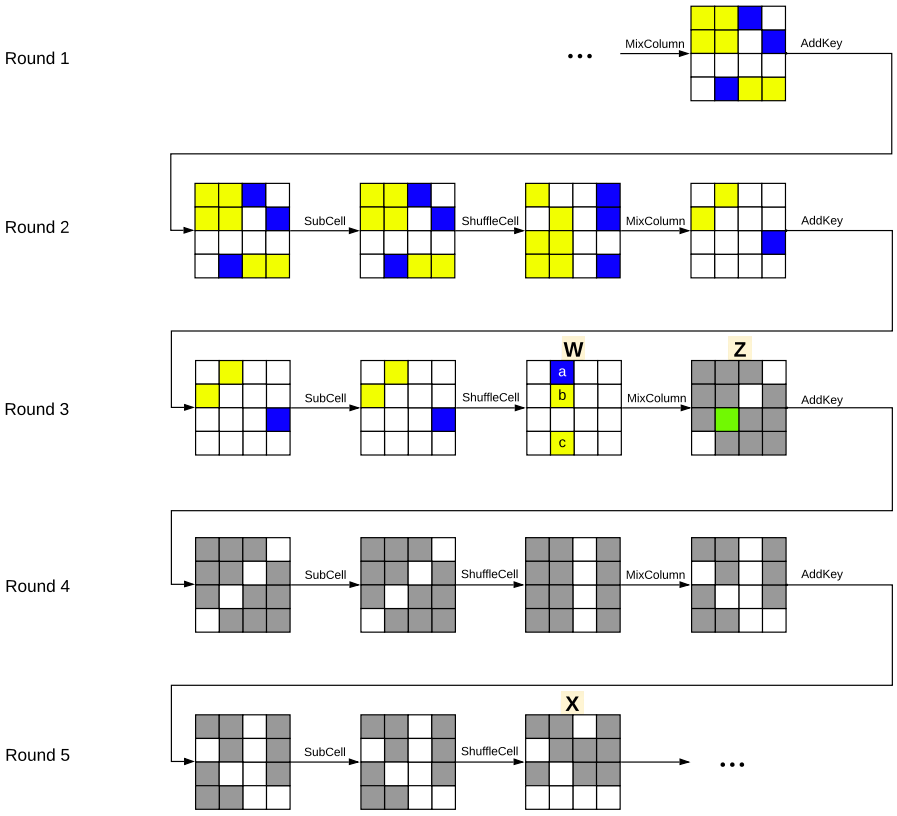

Figure 7. The retracing process of the 9-round attack

(8) For each of the remaining pairs, we apply the retracing mixtures described in Section 4.4. If a pair passes the process and suggests 116-bit key, we exhaustively search for the remaining 12-bit key. Otherwise, continue to the loops.

4.6 Complexities

Data complexity is 2^{36} since the attack requires 2^{36} chosen-ciphertexts.

Time complexity is $2^{122.3}$: Starting from Step 2, all the complexities will be multiplied by 2^{64} because we guess 64-bit WK in Step 1. In Step 2, we have 2 rounds of encryption for all the 2^{36} data. In Step 3, we sort 2^{36} data. While calculating the complexity of this sorting in terms of Midori encryptions is not straightforward, it is unnecessary to do so because the complexity of this step is negligible compared to the total complexity. In Step 4, we guess an additional 24-bit key. Then, starting from Step 5, all the complexities will also be multiplied by 2^{24}. In Step 5, we do $6/16 \cdot 2$ rounds encryption for each pair. In Step 6, we guess an additional 12-bit key then, starting from Step 7, all the complexities will also be multiplied by 2^{12}. In Step 7, we do $3/16 \cdot 2$ rounds encryption for each remaining pair. The complexity of Step 8 is $2^{11.3}$ encryptions for each pair as mentioned in Section 4.4. Hence, the total time complexity of the attack is $2^{122.3}$ as shown in Equation 1.

Memory complexity is 2^{36}: The mixture pool needs a memory of 2^{36} pairs of plaintexts and ciphertexts. The lists generated for the retracing processes use negligible memory. The pairs and eliminated pairs also use negligible memory.

5 Conclusion

In this paper, we mainly give a 9-round boomerang attack on Midori64 with data complexity 2^{36}, time complexity $2^{122.3}$ and memory complexity 2^{36}. We also express that the designers of Midori64 expected that much smaller rounds than 8 are secure against boomerang-type attacks.

In our attacks, we use the ideas in papers [10, 11] together with our new mixture pool technique (see Section 3.3). We believe that the mixture pool technique can be useful for further boomerang-type attacks for various block ciphers.

We also express that we have not utilized any automated tools such as MILP or SAT in order to drive

our characteristics. Therefore, we think that it maybe an interesting problem whether our characteristics are the best or not by using such automated tools, as a future work.

$$2^{64} \cdot \left(\frac{2}{9} \cdot 2^{36} + 2^{24}\left(\frac{6}{16} \cdot \frac{1}{9} \cdot 2^{31} \cdot 2 + 2^{12}\left(\frac{3}{16} \cdot \frac{1}{9} \cdot 2^{15} \cdot 2 + 2^{11} \cdot 2^{11.3}\right)\right)\right) \approx 2^{122.3} \tag{1}$$

Acknowledgment

The authors would like to thank the editor and anonymous reviewers for their valuable comments.

References

[1] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015.

[2] Jian Guo, Jérémy Jean, Ivica Nikolic, Kexin Qiao, Yu Sasaki, and Siang Meng Sim. Invariant subspace attack against Midori64 and the resistance criteria for S-box designs. *IACR Trans. Symmetric Cryptol.*, 2016(1):33–56, 2016.

[3] Yosuke Todo, Gregor Leander, and Yu Sasaki. Nonlinear invariant attack - practical attack on full scream, iscream, and Midori64. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 3–33, 2016.

[4] Li Lin and Wenling Wu. Meet-in-the-middle attacks on reduced-round Midori64. *IACR Trans. Symmetric Cryptol.*, 2017(1):215–239, 2017.

[5] Yong Liu, Zejun Xiang, Siwei Chen, Shasha Zhang, and Xiangyong Zeng. A novel automatic technique based on MILP to search for impossible differentials. In Mehdi Tibouchi and Xiaofeng Wang, editors, *Applied Cryptography and Network Security - 21st International Conference, ACNS 2023, Kyoto, Japan, June 19-22, 2023, Proceedings, Part I*, volume 13905 of *Lecture Notes in Computer Science*, pages 119–148. Springer, 2023.

[6] Aein Rezaei Shahmirzadi, Seyyed Arash Azimi, Mahmoud Salmasizadeh, Javad Mohajeri, and Mohammad Reza Aref. Impossible differential cryptanalysis of reduced-round Midori64 block cipher. In *14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology, ISCISC 2017, Shiraz, Iran, September 6-7, 2017*, pages 99–104. IEEE, 2017.

[7] Hongluan Zhao, Guoyong Han, Letian Wang, and Wen Wang. MILP-based differential cryptanalysis on round-reduced Midori64. *IEEE Access*, 8:95888–95896, 2020.

[8] Tim Beyne. Block cipher invariants as eigenvectors of correlation matrices. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 3–31. Springer, 2018.

[9] Ewan Fleischmann, Christian Forler, Michael Gorski, and Stefan Lucks. New boomerang attacks on ARIA. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings*, volume 6498 of *Lecture Notes in Computer Science*, pages 163–175. Springer, 2010.

[10] Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. The retracing boomerang attack. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 280–309. Springer, 2020.

[11] Alex Biryukov. The boomerang attack on 5 and 6-round reduced AES. In Hans Dobbertin, Vincent Rijmen, and Aleksandra Sowa, editors, *Advanced Encryption Standard - AES, 4th International Conference, AES 2004, Bonn, Germany, May 10-12, 2004, Revised Selected and Invited Papers*, volume 3373 of *Lecture Notes in Computer Science*, pages 11–15. Springer, 2004.

[12] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.

[13] David A. Wagner. The boomerang attack. In

Lars R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.

[14] Jérémy Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/, 2016.

[15] Lorenzo Grassi. Mixture differential cryptanalysis: a new approach to distinguishers and attacks on round-reduced AES. *IACR Trans. Symmetric Cryptol.*, 2018(2):133–160, 2018.

[16] Alex Biryukov, Christophe De Cannière, and Gustaf Dellkrantz. Cryptanalysis of SAFER++. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2003.

Mehmet Emin Gönen is currently a chief researcher at TÜBİTAK BİLGEM National Research Institute of Electronics and Cryptology. He received his B.S., M.S., and Ph.D. degrees in Mathematics from Boğaziçi University, Bahçeşehir University, and Gebze Technical University, respectively.

Muhammed Said Gündoğan is currently an experienced senior researcher at TÜBİTAK BİLGEM National Research Institute of Electronics and Cryptology. He received his B.S. and Ph.D. degrees in Mathematics both from Bilkent University.

Kamil Otal is currently an experienced senior researcher at TÜBİTAK BİLGEM National Research Institute of Electronics and Cryptology. He received his B.S. and M.S. degrees in mathematics both from TOBB University of Economics and Technology, and Ph.D. degree in Applied Mathematics from Middle East Technical University.