Practical-time Attack on the Full MMB Block Cipher*

Keting Jia[1,2], Jiazhe Chen[1,2], Meiqin Wang[1,2], and Xiaoyun Wang[1,2,3]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China
{ktjia, jiazhechen}@mail.sdu.edu.cn, mqwang@sdu.edu.cn
[2] School of Mathematics, Shandong University, Jinan 250100, China
[3] Institute for Advanced Study, Tsinghua University, Beijing 100084, China
xiaoyunwang@tsinghua.edu.cn

Abstract. Modular Multiplication based Block Cipher (MMB) is a block cipher designed by Daemen *et al.* as an alternative to the IDEA block cipher. In this paper, we give a practical-time attack on the full MMB with adaptive chosen plaintexts and ciphertexts. By the constructive sandwich distinguisher for 5 of the 6 rounds of MMB with amazingly high probability 1, we give the key recovery attack on the full MMB with data complexity 2^{40} and time complexity $2^{13.4}$ MMB encryptions. Then a rectangle-like sandwich attack on the full MMB is presented, with $2^{66.5}$ chosen plaintexts, 2^{64} MMB encryptions and $2^{70.5}$ memory bytes. By the way, we show an improved differential attack on the full MMB with data complexity of 2^{96} chosen plaintexts and ciphertexts, time complexity 2^{64} encryptions and 2^{66} bytes of memory.

Key words: MMB block cipher, sandwich distinguisher, practical attack, differential attack.

1 Introduction

Modular Multiplication based Block Cipher (MMB) [5] was designed by Daemen, Govaerts and Vandewalle in 1993. It uses the cyclic multiplication in $\mathbb{Z}_{2^{32}-1}$ and it was proposed as an alternative to the IDEA block cipher [7]. The number of rounds is 6. The block size and key size of MMB are both 128 bits. The key schedule of the original MMB, say MMB version 1.0, is successive rotated the last subkey by 32 bits to the left. A tweaked key-schedule against the related-key attack was designed by xoring a constant value to the subkey after each rotation, and the new cipher is called MMB version 2.0 [6]. The only cryptanalysis on MMB version 2.0 was proposed by Wang *et al.* [10], they proposed a differential attack on the full 6-round MMB with 2^{118} chosen plaintexts, $2^{95.61}$ encryptions

* Supported by 973 Project (No.2007CB807902), the National Natural Science Foundation of China (Grant No.60910118) and Tsinghua University Initiative Scientific Research Program (2009THZ01002).

Table 1. Summary of the Attacks on MMB

#Rounds	Type	Time	Data	Memory	Source
3	LC	2^{126} EN	$2^{114.56}$ KP	-	[10]
4	SQ	$2^{126.32}$ EN	2^{34} CP	2^{64}	[10]
6	DC	$2^{95.91}$ EN	2^{118} CP	2^{64}	[10]
6	SW	$2^{13.4}$EN	2^{40} ACP,ACC	2^{18}	this paper
6	SR	2^{64} EN	$2^{66.5}$ CP	$2^{70.5}$	this paper
6	SR	$2^{13.4}$ EN	$2^{66.5}$ CP,CC	$2^{70.5}$	this paper
6	DC	2^{64} EN	2^{96} CP	2^{66}	this paper
6	DC	2^{44} EN	2^{96} CP,CC	2^{66}	this paper

LC: Linear Cryptanalysis; DC: Differential Cryptanalysis;
SQ: Square Attack; SW: Sandwich Attack; SR: Rectangle-like Sandwich Attack;
EN: MMB Encryption; CP: Chosen Plaintexts;
ACP,ACC: Adaptive Chosen Plaintexts and Ciphertexts;
CP,CC: Chosen Plaintexts and Ciphertexts; KP: Known Plaintexts.

and 2^{64} memory blocks. They also presented linear and square attacks on the reduced-round MMB in [10].

In this paper, we construct a wonderful sandwich distinguisher of 5-round MMB with probability 1. Using the distinguisher, we present a practical-time attack on the full MMB. The data complexity of our attack is 2^{40} adaptively chosen plaintexts and ciphertexts, and the time complexity is $2^{13.4}$ MMB encryptions. Then we give a rectangle-like sandwich attack on the full MMB, with $2^{66.5}$ chosen plaintexts and 2^{64} encryptions. The memory complexity is $2^{70.5}$ bytes. Meanwhile, we introduce an improved differential attack on the full MMB with data complexity of 2^{96} chosen plaintexts, time complexity of 2^{64} MMB encryptions and 2^{66} bytes of memory. Both our attack and the attacks in [10] are independent of the key schedule algorithm. We summarize the existing results on MMB version 2.0 in Table 1.

2 Description of the Block Cipher MMB

MMB is a block cipher with 128-bit text block and 128-bit key. It has a Substitution-Permutation Network (SPN) structure and iterates 6 rounds. We use the notions in [10] to give a description of MMB. Let the 128-bit key of MMB as $K = (k_0, k_1, k_2, k_3)$, then the key schedule of MMB version 2.0 can be described as: $k_i^j = k_{(i+j) \bmod 4} \oplus (B \lll j)$, where k^j is the j-round subkey, $k^j = (k_0^j, k_1^j, k_2^j, k_3^j)$, $k_i^j (i = 0, \ldots, 3)$ are 32-bit words, and $j = 0, \ldots, 6$. The round transformation of MMB is denoted as ρ, then the j-th round can be described as:

$$\rho[k^j](X) = \theta \circ \eta \circ \gamma \circ \sigma[k^j](X),$$

where X is the 128-bit input value to the j-th round. The full MMB encryption is described as

$$E(P) = \sigma[k^6] \circ \rho[k^5] \circ \rho[k^4] \circ \rho[k^3] \circ \rho[k^2] \circ \rho[k^1] \circ \rho[k^0](P),$$

where P is the plaintext and k^6 is a post-whitening key. The 4 transformations $\sigma, \gamma, \eta, \theta$ of the round function are described as follows:

1. $\sigma[k^j]$: XOR the intermediate value with the round key,

$$\sigma[k^j](a_0, a_1, a_2, a_3) = (a_0 \oplus k_0^j, a_1 \oplus k_1^j, a_2 \oplus k_2^j, a_3 \oplus k_3^j),$$

   where $\oplus$ denotes bitwise exclusive-or, $a_i (i = 0, 1, 2, 3)$ are 32-bit words, and (a_0, a_1, a_2, a_3) is the 128-bit intermediate value. The operation $\sigma[k^j]$ is a involution, namely $\sigma[k^j]^{-1}(x) = \sigma[k^j](x)$.
2. γ: modular multiplication of each word of the intermediate value with fixed 32-bit constants G_i,

$$\gamma(a_0, a_1, a_2, a_3) = (a_0 \otimes G_0, a_1 \otimes G_1, a_2 \otimes G_2, a_3 \otimes G_3).$$

   For $0 < y < 2^{32} - 1$, let

$$x \otimes y = \begin{cases} x \times y \mod 2^{32} - 1 & \text{if } x < 2^{32} - 1, \\ 2^{32} - 1 & \text{if } x = 2^{32} - 1. \end{cases}$$

   $G_0 = 0x025f1cdb$, $G_1 = 2 \otimes G_0 = 0x04be39b6$, $G_2 = 2^3 \otimes G_0 = 0x12f8e6d8$, $G_3 = 2^7 \otimes G_0 = 0x2f8e6d81$, and the inverse values are $G_0^{-1} = 0x0dad4694$, $G_1^{-1} = 0x06d6a34a$, $G_2^{-1} = 0x81b5a8d2$, $G_3^{-1} = 0x281b5a8d$. There are two fixed points for any G_i: $0 \otimes G_i = 0$, and $(2^{32}-1) \otimes G_i = 2^{32}-1$. γ is invertible but not an involution.
3. η: operating on two of the four input words,

$$\eta(a_0, a_1, a_2, a_3) = (a_0 \oplus (lsb(a_0) \times \delta), a_1, a_2, a_3 \oplus ((1 \oplus lsb(a_3)) \times \delta)),$$

   where 'lsb' means the least significant bit, and $\delta = 0x2aaaaaaa$. η is an involution and a non-linear operation. But we can know that if there is a difference in the lsb of a_0, then the difference of a_0 after the transformation η will change, otherwise it will remain the same. It is the same to a_3.
4. θ: the only diffusion operation in MMB.

$$\theta(a_0, a_1, a_2, a_3) = (a_3 \oplus a_0 \oplus a_1, a_0 \oplus a_1 \oplus a_2, a_1 \oplus a_2 \oplus a_3, a_2 \oplus a_3 \oplus a_0).$$

   θ is an involution.

3 Boomerang Attack and Sandwich Attack

3.1 Boomerang Attack

The boomerang attack was first introduced by Wagner [11]. It is an adaptive chosen plaintext and ciphertext attack. And it was further developed by Kelsey

et al. [4] into a chosen plaintext attack called the amplified boomerang attack (Biham *et al.* independently introduced as the rectangle attack [2]) .

The boomerang attack bases on the differential attack [1], which joins two short differential characteristics with high probabilities in a quartet instead of a long differential to get a distinguisher with more rounds and higher probability. Let E be a block cipher with block size n, that is considered as a cascade of two sub-ciphers: $E = E_1 \circ E_0$. For the sub-cipher E_0 there is a differential characteristic $\alpha \rightarrow \beta$ with probability p, and for E_1 there is differential path $\gamma \rightarrow \zeta$ with probability q. $E^{-1}, {E_0}^{-1}$ and ${E_1}^{-1}$ stand for the inverse of E, E_0, E_1 respectively. The boomerang distinguisher can be constructed as follows:

- Randomly choose a pair of plaintexts (P, P') such that $P' \oplus P = \alpha$.
- Encrypt P, P' to get $C = E(P)$, $C' = E(P')$.
- Compute $\widetilde{C} = C \oplus \zeta$, $\widetilde{C'} = C' \oplus \zeta$. Decrypt $\widetilde{C}, \widetilde{C'}$ to get $\widetilde{P} = E^{-1}(\widetilde{C})$, $\widetilde{P'} = E^{-1}(\widetilde{C'})$.
- Check whether $\widetilde{P'} \oplus \widetilde{P} = \alpha$.

The quartet $(P, P', \widetilde{P}, \widetilde{P'})$, whose corresponding ciphertexts (C, C', $\widetilde{C}$, $\widetilde{C'}$), is a right quartet when it passes the boomerang distinguisher. That is to say, it satisfies the following conditions besides $P' \oplus P = \alpha$ and $\widetilde{P'} \oplus \widetilde{P} = \alpha$,

$$E_0(P') \oplus E_0(P) = \beta \tag{1}$$

$${E_1}^{-1}(\widetilde{C}) \oplus {E_1}^{-1}(C) = E_1^{-1}(\widetilde{C'}) \oplus {E_1}^{-1}(C') = \gamma \tag{2}$$

If a quartet satisfies the two equations above, we have ${E_1}^{-1}(\widetilde{C'}) \oplus {E_1}^{-1}(\widetilde{C}) = \beta$. Since we have a differential $\alpha \rightarrow \beta$ in E_0 and $P' \oplus P = \alpha$, the probability of equation (1) is p. Similarly, the probability of equation (2) is q^2, as the probabilities of $\gamma \rightarrow \zeta$ and $\zeta \rightarrow \gamma$ are the same. Finally, there is another probability of p to get $\widetilde{P'} \oplus \widetilde{P} = \alpha$ from ${E_1}^{-1}(\widetilde{C'}) \oplus {E_1}^{-1}(\widetilde{C}) = \beta$. As a result, the probability to get a right quartet is p^2q^2. The quartets that pass the distinguisher but don't satisfy equations (1) and (2) are called wrong quartets. It's known that for a random permutation, $\widetilde{P'} \oplus \widetilde{P} = \alpha$ with probability 2^{-n}. Therefore, the probability of the boomerang distinguisher should be greater than 2^{-n}, i.e., $pq > 2^{-n/2}$.

The rectangle (amplified boomerang) attack works in a chosen plaintext situation, with a birthday-paradox to make the condition $E_0(P) \oplus E_0(\widetilde{P}) = \gamma$ hold. For details about rectangle attack and amplified boomerang attack, we refer to citation [2], [4].

3.2 Sandwich Attack

Based on boomerang attack, Dunkelman *et al.* proposed a new attack named sandwich attack [8]. They divided the cipher into three sub-ciphers: $E = E_1 \circ E_M \circ E_0$. See Fig. 1. $X = E_0(P), Y = E_M(X), C = E_1(Y)$. There is a differential path $\alpha \rightarrow \beta$ with probability p in E_0 and $\gamma \rightarrow \zeta$ with probability q in E_1. The attack manner is also the same as the boomerang attack, the only difference

is that there is a E_M in the middle. Given $(Y \oplus \widetilde{Y} = \gamma), (Y' \oplus \widetilde{Y'} = \gamma)$ and $(X \oplus X' = \beta)$, calculate the probability

$$r = Pr((\widetilde{X} \oplus \widetilde{X'} = \beta) | (Y \oplus \widetilde{Y} = \gamma) \wedge (Y' \oplus \widetilde{Y'} = \gamma) \wedge (X \oplus X' = \beta)).$$

Then the probability of the sandwich distinguisher is p^2q^2r. In [8] the authors computed the probability using the properties of the feistel structure, and the same phenomenon is introduced in [11]. The SPN structure can also apply the sandwich attack, as used in [3] with a name "ladder switch". In this paper, we also mount the sandwich attack on MMB, which is a block cipher with SPN structure.

4 5-Round Sandwich Distinguisher with Probability 1

In this section, we construct a sandwich distinguisher for 5-round MMB, and surprisingly, the distinguisher has probability 1.

As mentioned in the previous section, we decompose 5-round MMB into $E = E_1 \circ E_M \circ E_0$. E_0 contains the first 2 rounds, E_M is the third round and E_1 contains the last 2 rounds. See Fig. 1.

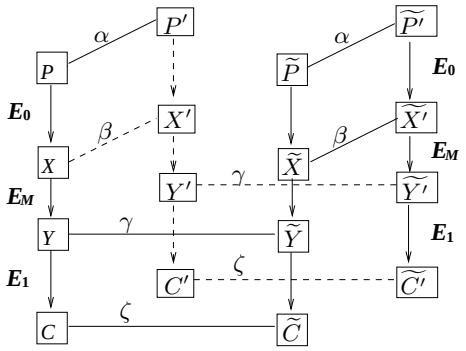

Fig. 1. 5 rounds sandwich distinguisher

We use the following 2-round differential characteristic with probability 1 proposed by Wang *et al.* [10] both in E_0 and E_1:

$$(0,\bar{0},\bar{0},0) \xrightarrow{\sigma[k^i]} (0,\bar{0},\bar{0},0) \xrightarrow{\gamma} (0,\bar{0},\bar{0},0) \xrightarrow{\eta} (0,\bar{0},\bar{0},0) \xrightarrow{\theta} (\bar{0},0,0,\bar{0})$$
$$\xrightarrow{\sigma[k^{i+1}]} (\bar{0},0,0,\bar{0}) \xrightarrow{\gamma} (\bar{0},0,0,\bar{0}) \xrightarrow{\eta} (\bar{0} \oplus \delta, 0, 0, \bar{0} \oplus \delta) \xrightarrow{\theta} (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0),$$

where '0' denotes a 32-bit zero difference word and $\bar{0} = 2^{32} - 1 = 0xffffffff$. So $\alpha = \gamma = (0,\bar{0},\bar{0},0)$, $\beta = \zeta = (0,\bar{0} \oplus \delta, \bar{0} \oplus \delta, 0)$, and $Pr(\alpha \to \beta) = 1$,

$Pr(\gamma \to \zeta) = 1$. Now it remains to estimate the probability $r = Pr((\widetilde{X} \oplus \widetilde{X'} = \beta) | (Y \oplus \widetilde{Y} = \gamma) \wedge (Y' \oplus \widetilde{Y'} = \gamma) \wedge (X \oplus X' = \beta))$. In the rest of this section, we will give the deduction of r.

Let's denote the i-th word of $X, X', \widetilde{X}, \widetilde{X'}$ by X_i, X'_i, $\widetilde{X}_i$, $\widetilde{X'}_i$, $i = 0, 1, 2, 3$. The subkey of the third round is denoted $k = (k_0, k_1, k_2, k_3)$.

We have

$$\begin{aligned} Y \oplus \widetilde{Y} &= (0, \bar{0}, \bar{0}, 0), \\ Y' \oplus \widetilde{Y'} &= (0, \bar{0}, \bar{0}, 0), \\ X \oplus X' &= (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0). \end{aligned} \tag{3}$$

Since θ and η are involutions, and θ is linear, we have

$$(\eta^{-1} \circ \theta^{-1}(Y)) \oplus (\eta^{-1} \circ \theta^{-1}(\widetilde{Y})) = (\bar{0} \oplus \delta, 0, 0, \bar{0} \oplus \delta), \tag{4}$$

$$(\eta^{-1} \circ \theta^{-1}(Y')) \oplus (\eta^{-1} \circ \theta^{-1}(\widetilde{Y'})) = (\bar{0} \oplus \delta, 0, 0, \bar{0} \oplus \delta). \tag{5}$$

Besides,

$$\begin{aligned} Y &= \theta \circ \eta \circ \gamma \circ \sigma[k](X), \\ Y' &= \theta \circ \eta \circ \gamma \circ \sigma[k](X'), \\ \widetilde{Y} &= \theta \circ \eta \circ \gamma \circ \sigma[k](\widetilde{X}), \\ \widetilde{Y'} &= \theta \circ \eta \circ \gamma \circ \sigma[k](\widetilde{X'}). \end{aligned} \tag{6}$$

We can get the following equations from equations (4), (5) and (6),

$$\begin{aligned} ((X_1 \oplus k_1) \otimes G_1) \oplus ((\widetilde{X}_1 \oplus k_1) \otimes G_1) = 0, \\ ((X_2 \oplus k_2) \otimes G_2) \oplus ((\widetilde{X}_2 \oplus k_2) \otimes G_2) = 0, \\ ((X'_1 \oplus k_1) \otimes G_1) \oplus ((\widetilde{X'}_1 \oplus k_1) \otimes G_1) = 0, \\ ((X'_2 \oplus k_2) \otimes G_2) \oplus ((\widetilde{X'}_2 \oplus k_2) \otimes G_2) = 0. \end{aligned}$$

So $X_1 = \widetilde{X}_1$, $X_2 = \widetilde{X}_2$, $X'_1 = \widetilde{X'}_1$, $X'_2 = \widetilde{X'}_2$. And we have $X_1 \oplus \widetilde{X}_1 = X'_1 \oplus \widetilde{X'}_1$, $X_2 \oplus \widetilde{X}_2 = X'_2 \oplus \widetilde{X'}_2$. Then we conclude

$$\begin{aligned} \widetilde{X}_1 \oplus \widetilde{X'}_1 = X_1 \oplus X'_1 = \bar{0} \oplus \delta, \\ \widetilde{X}_2 \oplus \widetilde{X'}_2 = X_2 \oplus X'_2 = \bar{0} \oplus \delta. \end{aligned} \tag{7}$$

Also from equation (4), (5) and (6), we have

$$\begin{aligned} ((X_0 \oplus k_0) \otimes G_0) \oplus ((\widetilde{X}_0 \oplus k_0) \otimes G_0) = \bar{0} \oplus \delta, \\ ((X_3 \oplus k_3) \otimes G_3) \oplus ((\widetilde{X}_3 \oplus k_3) \otimes G_3) = \bar{0} \oplus \delta, \\ ((X'_0 \oplus k_0) \otimes G_0) \oplus ((\widetilde{X'}_0 \oplus k_0) \otimes G_0) = \bar{0} \oplus \delta, \\ ((X'_3 \oplus k_3) \otimes G_3) \oplus ((\widetilde{X'}_3 \oplus k_3) \otimes G_3) = \bar{0} \oplus \delta. \end{aligned}$$

Then

$$((X_0 \oplus k_0) \otimes G_0) \oplus ((\widetilde{X}_0 \oplus k_0) \otimes G_0) = ((X'_0 \oplus k_0) \otimes G_0) \oplus ((\widetilde{X'}_0 \oplus k_0) \otimes G_0),$$
$$((X_3 \oplus k_3) \otimes G_3) \oplus ((\widetilde{X}_3 \oplus k_3) \otimes G_3) = ((X'_3 \oplus k_3) \otimes G_3) \oplus ((\widetilde{X'}_3 \oplus k_3) \otimes G_3).$$

From equation (3), we have $X_0 = X'_0 = 0$, $X_3 = X'_3 = 0$, so

$$((\widetilde{X}_0 \oplus k_0) \otimes G_0) = ((\widetilde{X'}_0 \oplus k_0) \otimes G_0),$$
$$((\widetilde{X}_3 \oplus k_3) \otimes G_3) = ((\widetilde{X'}_3 \oplus k_3) \otimes G_3).$$

Then $\widetilde{X}_0 = \widetilde{X'}_0$, and $\widetilde{X}_3 = \widetilde{X'}_3$. Finally, we have

$$\widetilde{X}_0 \oplus \widetilde{X'}_0 = 0,$$
$$\widetilde{X}_3 \oplus \widetilde{X'}_3 = 0. \tag{8}$$

Combining equation (7) and (8), we have

$$\widetilde{X} \oplus \widetilde{X'} = (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0) = \beta.$$

So

$$r = Pr((\widetilde{X} \oplus \widetilde{X'} = \beta) | (Y \oplus \widetilde{Y} = \gamma) \wedge (Y' \oplus \widetilde{Y'} = \gamma) \wedge (X \oplus X' = \beta)) = 1.$$

As a result, the probability of our 5-round sandwich distinguisher is 1.

5 Practical Sandwich Attack on the Full MMB

In this section, we use the distinguisher described in Sec. 6 to rounds 2-6 and recover some subkey bits of the first round. Then we apply the distinguisher to rounds 1-5 and recover some subkey bits of the last round. The key can be deduced from the recovered subkey bits.

5.1 The Key Recovery Attack

1. **Getting Right Quartets**
   Choose 2^{37} plaintexts P at random, compute $P' = P \oplus (0xfdff77ef, 0, 0, 0xdffbfeef)$, and encrypt P, P' to get ciphertexts pairs C, C'. Calculate $\widetilde{C} = C \oplus (0, \bar{0} \oplus \beta, \bar{0} \oplus \beta, 0)$, $\widetilde{C'} = C' \oplus (0, \bar{0} \oplus \beta, \bar{0} \oplus \beta, 0)$, and decrypt $\widetilde{C}$, $\widetilde{C'}$ to get $\widetilde{P}$, $\widetilde{P'}$. Store quartets $(P, P', \widetilde{P}, \widetilde{P'})$ only when $\widetilde{P} \oplus \widetilde{P'} = (*, 0, 0, *)$, where '*' stands for any non-zero 32-bit value.
   We know the differential characteristic

$$(0xfdff77ef, 0, 0, 0xdffbfeef) \xrightarrow{\sigma[k^i]} (0xfdff77ef, 0, 0, 0xdffbfeef) \xrightarrow{\gamma}$$
$$(\bar{0} \oplus \delta, 0, 0, \bar{0} \oplus \delta) \xrightarrow{\eta} (\bar{0}, 0, 0, \bar{0}) \xrightarrow{\theta} (0, \bar{0}, \bar{0}, 0)$$

holds with probability 2^{-36}, as the probability of $0xfdff77ef \xrightarrow{G_0} \bar{0} \oplus \delta$ and $0xdffbfeef \xrightarrow{G_3} \bar{0} \oplus \delta$ are both 2^{-18}. Thus, there are $2^{37} \cdot 2^{-36} = 2$ pairs of (P, P') satisfying the differential characteristic above, and we call it a right pair. Once a pair (P, P') is a right pair, the corresponding $(\widetilde{P}, \widetilde{P'})$ must satisfy $\widetilde{P} \oplus \widetilde{P'} = (*, 0, 0, *)$, that is because the probability of our distinguisher is 1. So the quartet $(P, P', \widetilde{P}, \widetilde{P'})$ is a right quartet.
For a wrong quartet $(P, P', \widetilde{P}, \widetilde{P'})$, $\widetilde{P} \oplus \widetilde{P'} = (*, 0, 0, *)$ holds with probability 2^{-64}. Therefore, there are $2^{37} \cdot 2^{-64} = 2^{-27}$ wrong quartets left, so we regard the quartets which satisfy $\widetilde{P} \oplus \widetilde{P'} = (*, 0, 0, *)$ as right quartets.

2. **Partial Key Recovery**
In order to make the attack faster, we use a pre-computation. Construct two tables T_0 and T_3, and each is about 2^{16} bytes. Because there are at most $2^{14.28}$ 32-bit words x to make the difference character $0xfdff77ef \xrightarrow{G_0} \bar{0} \oplus \delta$ hold. It is the same to the difference character of G_3.

$$T_0 = \{\, x \mid (x \otimes G_0) \oplus ((x \oplus 0xfdff77ef) \otimes G_0) = \bar{0} \oplus \delta \,\},$$
$$T_3 = \{\, x \mid (x \otimes G_3) \oplus ((x \oplus 0xdffbfeef) \otimes G_3) = \bar{0} \oplus \delta \,\}.$$

   (a) For each quartet $(P, P', \widetilde{P}, \widetilde{P'})$ stored, we calculate 32-bit values of $k_0^0 = P \oplus x, x \in T_0$, and filter the wrong keys with the following equations.

$$((P_0 \oplus k_0^0) \otimes G_0) \oplus ((P'_0 \oplus k_0^0) \otimes G_0) = \bar{0} \oplus \delta, \tag{9}$$
$$((\widetilde{P}_0 \oplus k_0^0) \otimes G_0) \oplus ((\widetilde{P'}_0 \oplus k_0^0) \otimes G_0) = \bar{0} \oplus \delta. \tag{10}$$

   There are $2^{32} \cdot 2^{-36} = 2^{-4}$ keys to make equations (9) and (10) hold, so a wrong key satisfies the equations with probability 2^{-36}.
   There are 2 quarters and 2^{-27} wrong pairs left, so the expect number of the right key is $2 + 2^{-36} \cdot 2^{-27} \approx 2$. For a wrong key, it will be left with probability at most 2^{-36}, so the total numbers of wrong keys are $(2 \cdot 2^{-36} + 2^{-36} \cdot 2^{-27}) \cdot 2^{32} = 2^{-3}$, and the expect number of a wrong key is $2^{-3}/2^{32} = 2^{-35}$. Then by Poisson distribution, the success rate is about 0.91.

   (b) Similarly, in use of the table T_3, we can recover 32-bit value k_3^0 with the same complexity and success rate as in (a).

   After recovering 64 bits of the first round subkey, we mount the distinguisher to rounds 1-5, and recover 64 bits of the subkey of last round.

3. **Recovering 64 Bits of the Last Round Subkey**
Construct two tables T_1 and T_2, and each is about 2^{16} bytes.

$$T_1 = \{\, x \mid (x \otimes G_1^{-1}) \oplus ((x \oplus 0xfcfbdfff) \otimes G_1^{-1}) = \bar{0} \oplus \delta\},$$
$$T_2 = \{\, x \mid (x \otimes G_2^{-1}) \oplus ((x \oplus 0xf3ef7fff) \otimes G_2^{-1}) = \bar{0} \oplus \delta\}.$$

   (a) We use the method similar to that described in Step 1 to gain right quartets. But this time, we choose 2^{37} ciphertexts C, calculate $\widetilde{C} = C \oplus (0xfcfbdfff, 0x0f14a000, 0x0f14a000, 0xf3ef7fff)$. Decrypt C,

$\widetilde{C}$ to get P, $\widetilde{P}$. Calculate $P' = P \oplus (0, \bar{0}, \bar{0}, 0)$, $\widetilde{P'} = P' \oplus (0, \bar{0}, \bar{0}, 0)$, then encrypt them to get C', $\widetilde{C'}$. And store the quartet with $C' \oplus \widetilde{C'} = (V_1, V_1 \oplus V_2, V_1 \oplus V_2, V_2)$, where V_1, V_2 are non-zero 32-bit words. There are about 2 quartets to be stored, which are the right quartets, for the following differential characteristic has probability 2^{-36}.

$$(0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0) \overset{\sigma^{-1}[k^5]}{\longleftarrow} (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0) \overset{\gamma^{-1}}{\longleftarrow}$$
$$(0, 0xfcfbdfff, 0xf3ef7fff, 0) \overset{\eta^{-1}}{\longleftarrow}$$
$$(0, 0xfcfbdfff, 0xf3ef7fff, 0) \overset{\theta^{-1}}{\longleftarrow}$$
$$((0xfcfbdfff, 0x0f14a000, 0x0f14a000, 0xf3ef7fff))$$

The probability of $0xfcfbdfff \xrightarrow{G_1^{-1}} \bar{0} \oplus \delta$ and $0xf3ef7fff \xrightarrow{G_2^{-1}} \bar{0} \oplus \delta$ are both 2^{-18}.

(b) Then we recover 64 bits of the equivalent key $k^{6'}$ of k^6, i.e., $k_1^{6'} = k_0^6 \oplus k_1^6 \oplus k_2^6$ and $k_2^{6'} = k_1^6 \oplus k_2^6 \oplus k_3^6$. As above, we calculate 32-bit value of $k_1^{6'} = C_0 \oplus C_1 \oplus C_2 \oplus x, x \in T_1$, and filter the wrong key with the following equation.

$$(G_1^{-1} \otimes (\widetilde{C}_0 \oplus \widetilde{C}_1 \oplus \widetilde{C}_2 \oplus k_1^{6'})) \oplus (G_1^{-1} \otimes (\widetilde{C'}_0 \oplus \widetilde{C'}_1 \oplus \widetilde{C'}_2 \oplus k_1^{6'})) = \bar{0} \oplus \delta.$$

In the similar way, we calculate 32-bit value of $k_2^{6'} = C_1 \oplus C_2 \oplus C_3 \oplus x, x \in T_2$, and filter the wrong key with the following equation

$$(G_2^{-1} \otimes (\widetilde{C}_1 \oplus \widetilde{C}_2 \oplus \widetilde{C}_3 \oplus k_2^{6'})) \oplus (G_2^{-1} \otimes (\widetilde{C'}_1 \oplus \widetilde{C'}_2 \oplus \widetilde{C'}_3 \oplus k_1^{6'})) = \bar{0} \oplus \delta.$$

According to the key schedule algorithm, $k_0^0 = k_0 \oplus B$, $k_3^0 = k_3 \oplus B$, $k_1^{6'} = k_0 \oplus k_2 \oplus k_3 \oplus (B \lll 6)$, $k_2^{6'} = k_0 \oplus k_1 \oplus k_3 \oplus (B \lll 6)$. As a result, we recover the whole 128 bits of the key.

The data complexity is 2^{40} adaptive chosen plaintexts and ciphertexts. The time complexity is about $2 \cdot 2 \cdot 2^{14} = 2^{16}$ one round MMB encryptions, which are equivalent to $2^{13.4}$ MMB encryptions. The memory complexity is about 2^{18} bytes.

5.2 Rectangle-Like Sandwich Attack on the Full MMB

We can transform the 5-round sandwich distinguisher into a rectangle-like sandwich distinguisher by only choosing the plaintexts. In the rectangle-like sandwich distinguisher, we can choose $P \oplus P' = (0, \bar{0}, \bar{0}, 0)$, $\widetilde{P} \oplus \widetilde{P'} = (0, \bar{0}, \bar{0}, 0)$, so $X \oplus X' = (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0)$, $\widetilde{X} \oplus \widetilde{X'} = (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0)$ with probability 1. But there is a probability of 2^{-128} to get $Y \oplus \widetilde{Y} = (0, \bar{0}, \bar{0}, 0)$. Once we have $Y \oplus \widetilde{Y} = (0, \bar{0}, \bar{0}, 0)$, $Pr((Y' \oplus \widetilde{Y'} = \gamma) | (Y \oplus \widetilde{Y} = \gamma) \wedge (\widetilde{X} \oplus \widetilde{X'} = \beta) \wedge (X \oplus X' = \beta)) = 1$. So using the birthday-paradox, if we randomly choose 2^{64} pairs of (P, P') with $P \oplus P' = (0, \bar{0}, \bar{0}, 0)$ and 2^{64} pairs of $(\widetilde{P}, \widetilde{P'})$ with $\widetilde{P} \oplus \widetilde{P'} = (0, \bar{0}, \bar{0}, 0)$, there will be one quartet $(P, P', \widetilde{P}, \widetilde{P'})$ passing the distinguisher. Then we can use the 5-round distinguisher to attack the full MMB.

The Rectangle-Like Sandwich Attack. We choose $2^{65.5}$ plaintexts P at random, construct the structure

$$S = \{ (P, P') \mid P \oplus P' = (0, \bar{0}, \bar{0}, 0) \},$$

and encrypt each $(P, P') \in S$ to get (C, C'). Store the ciphertext pairs with index $(C_0 \oplus C_1 \oplus C_2, C_1 \oplus C_2 \oplus C_3)$. There are $2^{65.5} \cdot 2^{65.5} \cdot 2^{-1} \cdot 2^{-128} = 4$ quartets $(P, P', \widetilde{P}, \widetilde{P'})$ satisfying the conditions of the distinguisher, and they are right quartets, where the pair $(\widetilde{P}, \widetilde{P'}) \in S$, whose ciphertexts are denoted as $(\widetilde{C}, \widetilde{C'})$.

If a quartet is a right quartet, then it must satisfy $C \oplus \widetilde{C} = (V_1, V_1 \oplus V_2, V_1 \oplus V_2, V_2)$ and $C' \oplus \widetilde{C'} = (W_1, W_1 \oplus W_2, W_1 \oplus W_2, W_2)$, because it satisfies the output differences of the distinguisher, where V_1, V_2, W_1, W_2 are non-zero 32-bit words. That is to say

$$\begin{aligned}
(C_0 \oplus C_1 \oplus C_2) \oplus (\widetilde{C_0} \oplus \widetilde{C_1} \oplus \widetilde{C_2}) &= 0, \\
(C_1 \oplus C_2 \oplus C_3) \oplus (\widetilde{C_1} \oplus \widetilde{C_2} \oplus \widetilde{C_3}) &= 0, \\
(C'_0 \oplus C'_1 \oplus C'_2) \oplus (\widetilde{C'_0} \oplus \widetilde{C'_1} \oplus \widetilde{C'_2}) &= 0, \\
(C'_1 \oplus C'_2 \oplus C'_3) \oplus (\widetilde{C'_1} \oplus \widetilde{C'_2} \oplus \widetilde{C'_3}) &= 0.
\end{aligned}$$

Furthermore, for a right quartet the input difference of the 6-th round is $(0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0)$, the number of possible output difference values given the input difference $\bar{0} \oplus \delta$ for G_1 or G_2 is about $2^{28.56}$. So we use all these conditions to sieve (C, C'),$(\widetilde{C}, \widetilde{C'})$, if a quartet doesn't pass the distinguisher, the probability for it to satisfy these conditions is $2^{-64} \cdot 2^{-64} \cdot 2^{(28.65-32)\times 4} = 2^{-141.76}$. So the number of wrong quartets is $2^{65.5} \cdot 2^{65.5} \cdot 2^{-1} \cdot 2^{-141.76} = 2^{-11.76}$.

With the right quartets stored, we recover the equivalent key $k_1^{6'}$ and $k_2^{6'}$ as in Subsection 5.1 with 2^{17} bytes memory and $2^{12.4}$ MMB encryptions by constructing the table T_1 and T_2. Then we guess the rest 64-bit keys, and filter the wrong keys by encrypting a plaintext whose ciphertext is known. It is about 2^{64} MMB encryptions. The data complexity of the attack is $2 \cdot 2^{65.5} = 2^{66.5}$ chosen plaintexts, the memory complexity is dominated by the complexity of storing the ciphertexts, that is $2^{66.5}$ 128-bit words, i.e., $2^{70.5}$ bytes.

The rest 64-bit key can be recovered with the 5-round rectangle-like sandwich distinguisher to rounds 2-6 by chosen ciphertexts attack. In this case the data complexity is $2^{66.5}$ chosen plaintexts and ciphertexts, the memory complexity is $2^{70.5}$ bytes too, and the time complexity is $2^{13.4}$ MMB encryptions.

6 The Improved Differential Cryptanalysis of MMB

In this section, we give a 6-round differential path for MMB. The differential path is described in the following,

$$\begin{aligned}
(0, \bar{0}, \bar{0}, 0) \xrightarrow{\rho[k^0]} (\bar{0}, 0, 0, \bar{0}) \xrightarrow{\rho[k^1]} (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0) \xrightarrow{\rho[k^2]} (\tau, 0, 0, \tau) \xrightarrow{\rho[k^3]} \\
(0, \bar{0}, \bar{0}, 0) \xrightarrow{\rho[k^4]} (\bar{0}, 0, 0, \bar{0}) \xrightarrow{\rho[k^5]} (0, \bar{0} \oplus \delta, \bar{0} \oplus \delta, 0),
\end{aligned}$$

where τ belong to the sets of the differences, which is chosen to make the differential characteristic $\bar{0} \oplus \delta \xrightarrow{G_1} \tau \xrightarrow{G_0} \bar{0}$ and $\bar{0} \oplus \delta \xrightarrow{G_2} \tau \xrightarrow{G_3} \bar{0}$ hold.

We search all τ satisfying the above differentia characteristic, which are used to produce the 5-round differential path. The probability for the differential path is 2^{-94}.

Then we use the last five rounds of the differential path, i.e.

$$(\bar{0},0,0,\bar{0}) \xrightarrow{\rho[k^0]} (0,\bar{0}\oplus\delta,\bar{0}\oplus\delta,0) \xrightarrow{\rho[k^1]} (\tau,0,0,\tau) \xrightarrow{\rho[k^2]} (0,\bar{0},\bar{0},0) \xrightarrow{\rho[k^3]}$$
$$(\bar{0},0,0,\bar{0}) \xrightarrow{\rho[k^4]} (0,\bar{0}\oplus\delta,\bar{0}\oplus\delta,0),$$

to attack the full round MMB. We mount the 5-round differential path to rounds 1-5 of the 6 rounds. In the rest of the section, we give the attack algorithm.

The Key Recovery Attack. We choose 2^{96} pairs of plaintext with difference $(\bar{0},0,0,\bar{0})$, then there are 4 right pairs. The output difference of the 5-th round for a right pair is $(0,\bar{0}\oplus\delta,\bar{0}\oplus\delta,0)$, so the difference of the ciphertext should be $(V_1, V_1 \oplus V_2, V_1 \oplus V_2, V_2)$, where V_1, V_2 are non-zero 32-bit words. We use this to sieve the ciphertext pairs, and there will be $2^{96} \cdot 2^{-64} = 2^{32}$ pairs left. Furthermore, the input difference of the 6-th round is $(0,\bar{0}\oplus\delta,\bar{0}\oplus\delta,0)$, the number of possible output difference values given the input difference $\bar{0}\oplus\delta$ for G_1 or G_2 is about $2^{28.56}$. So there are $2^{32} \cdot 2^{(28.56-32)\times 2} = 2^{25.12}$ pairs satisfying the output difference.

As the key recovery attack in Subsection 5.1. We construct two tables T_1 and T_2

$$T_1 = \{\, x \mid (x \otimes G_1^{-1}) \oplus ((x \oplus 0xfcfbdfff) \otimes G_1^{-1}) = \bar{0} \oplus \delta \,\},$$
$$T_2 = \{\, x \mid (x \otimes G_2^{-1}) \oplus ((x \oplus 0xf3ef7fff) \otimes G_2^{-1}) = \bar{0} \oplus \delta \,\}.$$

to recover the equivalent subkey words $k_1^{6'}$, $k_2^{6'}$ with the remaining ciphertext pairs (C, C'), where $C = (C_0, C_1, C_2, C_3)$ and $C' = (C'_0, C'_1, C'_2, C'_3)$. Calculate the 32-bit words $k_1^{6'} = C_0 \oplus C_1 \oplus C_2 \oplus x, x \in T_1$, $k_2^{6'} = C_1 \oplus C_2 \oplus C_3 \oplus x, x \in T_3$, and increase the counter corresponding to $(k_1^{6'}, k_2^{6'})$ by 1. For G_1 and G_2, the number of pairs with input difference $\bar{0}\oplus\delta$ and any given output difference is at most $2^{14.28}$, so the maximum count per counted pair of the wrong subkey words will be $2^{14.28} \cdot 2^{14.28} = 2^{28.56}$. The signal-to-noise ratio is :

$$S/N = \frac{p \cdot 2^k}{\alpha \cdot \beta} = \frac{2^{-96} \times 2^{64}}{2^{-64-6.88} \times 2^{28.56}} = 2^{10.32}.$$

According to citation [9], the success probability is

$$Ps = \int_{-\frac{\sqrt{\mu S/N} - \Phi^{-1}(1-2^{-a})}{\sqrt{S/N+1}}}^{\infty} \Phi(x)dx = 0.9542,$$

where $a = 64$ is the number of subkey bits guessed, μ is the number of right pairs and $\mu = 4$.

The data complexity of the attack is 2^{96} chosen plaintexts. The time complexity is about $2 \cdot 2^{14.28} \cdot 2^{25.12} = 2^{40.40}$ XOR operations and $2^{14.28} \cdot 2^{14.28} \cdot 2^{25.12} =$

$2^{53.68}$ counts, equivalent to 2^{43} MMB encryptions. The memory complexity is 2^{64} 64-bit counters, equivalent to 2^{66} bytes. The rest 64 bits of the key can be recovered by exhaustive search, which determine the time complexity is 2^{64} MMB encryptions.

We can mount the 5 round differential characteristic to round 2-6. Then we use the chosen ciphertext attack to recover the 64 bits subkey of the first round. Then we can compute the key. The data complexity is 2^{96} chosen plaintexts and ciphertexts, the memory complexity is 2^{66} bytes and the time complexity is 2^{44} MMB encryptions.

Note that we can even use the 6-round differential path to attack 7-round MMB with the same complexity as the 6-round attack. It means that even if MMB has 7 rounds it is still vulnerable to the differential attack.

7 Conclusion

In this paper, we construct a 5-round sandwich distinguisher for MMB with amazingly high probability 1. With the distinguisher, we recover the 128-bit key on the full MMB with 2^{40} adaptive chosen plaintexts and ciphertexts, $2^{13.4}$ MMB encryptions and 2^{18} bytes memory. On this bases, we present a rectangle-like sandwich attack to the full MMB, with $2^{66.5}$ chosen plaintexts, 2^{64} MMB encryptions and $2^{70.5}$ bytes memory. Besides, we improve the differential attack on the full MMB in [10]. The data complexity is 2^{96} chosen plaintexts, the time complexity is 2^{64} MMB encryptions and the memory complexity is 2^{66} bytes.

References

1. Biham, E., Shamir, A.: Differential Cryptanalysis of The Data Encryption Standard. Springer, London (1993)
2. Biham, E., Dunkelman, O., Keller, N.: The Rectangle Attack - Rectangling the Serpent. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 340-357. Springer, Heidelberg (2001)
3. Biryukov, A., Khovaratovich, D.: Related Key Cryptanalysis of the Full AES-192 and AES-256. Asiacrypt 2009, LNCS 5912, pp. 1-18, 2009.
4. Kelsey, J., Khono, T., Schneier, B.: Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 75-93. Springer, Heidelberg (2000)
5. Daemen, J., Govaerts, R., Vandewalle, J.: Block Ciphers Based on Modular Multiplication. In: Wolfowicz, W. (ed.) Proceedings of 3rd Symposium on State and Progress of Research in Cryptography, Fondazione Ugo Bordoni, pp. 80-89 (1993)
6. Daemen, J.: Cipher and Hash Function Design C Strategies based on Linear and Differential Cryptanalysis. PhD Thesis, Dept. Elektrotechniek, Katholieke Universiteit Leuven, Belgium (1995)
7. Lai, X.: On the Design and Security of Block Ciphers. In: Massey, J.L. (ed.) ETH Series in Information Processing, vol. 1. Hartung-Gorre Verlag, Konstanz (1995)
8. Dunkelman, O., Keller, N., Shamir, A.: A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony.

9. Selçuk, A.A., Biçak, A.: On Probability of Success in Linear and Differential Cryptanalysis. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 174-85. Springer, Heidelberg (2003)
10. Wang, M., Nakahara Jr., J., Sun, Y.: Cryptanalysis of the Full MMB Block Cipher. In: Jacobson Jr., M.J.,Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 231-248. Springer, Heidelberg (2009).
11. Wagner, D.: The Boomerang Attack. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 156-170. Springer, Heidelberg (1999)