Distinguishing Attack and Second-Preimage Attack on the CBC-like MACs *

Keting Jia[1], Xiaoyun Wang **[1,2], Zheng Yuan[2,3], and Guangwu Xu[4]

[1] Key Laboratory of Cryptographic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China
`kejia@mail.sdu.edu.cn`
[2] Center for Advanced Study, Tsinghua University, Beijing 100084, China
`xiaoyunwang@mail.tsinghua.edu.cn`
[3] Beijing Electronic Science and Technology Institute, Beijing 100070,China
`zyuan@mail.tsinghua.edu.cn`
[4] Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, USA
`gxu4uwm@uwm.edu`

Abstract. In this paper, we first present a new distinguisher on the CBC-MAC based on a block cipher in Cipher Block Chaining (CBC) mode. It can also be used to distinguish other CBC-like MACs from random functions. The main results of this paper are on the second-preimage attack on CBC-MAC and CBC-like MACs include TMAC, OMAC, CMAC, PC-MAC and MACs based on three-key encipher CBC mode. Instead of exhaustive search, this attack can be performed with the birthday attack complexity.
Keywords: CBC, MAC, Distinguishing attack, Second preimage attack.

1 Introduction

A message authentication code (MAC), also known as keyed hash function, is a short piece of information used to authenticate both the source of a message and its integrity without the use of any additional mechanisms. A MAC algorithm takes as input a secret key and an arbitrary-length message to be authenticated, and outputs a short tag. As an important cryptographic primitive, MAC has been widely used in practice. The applications include Internet communication protocols, e-commerce, ebanking etc. The analysis of MAC model usually involves three participants: a sender, a receiver and an adversary. The sender and the receiver have agreed on a secret key (or a set of keys). Prior to sending a message, the sender uses a signing algorithm to produce an authentication tag (or MAC) from the message and the secret key. On receipt, the receiver verifies the message and the MAC by the same calculation using their share secret

* This work is supported by the National Outstanding Young Scientist (No.60525201) and National 973 Program of China (No.2007CB807902)
** To whom correspondence should be addressed.

key. The goal of the adversary is to trick the receiver into accepting a message that was not sent by the sender. There are three main types of constructions for MAC algorithms: construction based on block ciphers (OMAC, CBC-MAC and PMAC), construction based on cryptographic hash functions (HMAC), and construction based on universal hashing. In this paper we shall focus on the first type of construction.

The Security of MAC Functions One of the most important requirements of MAC is that given a massage M and a l_k-bit secret key K, the computation of the MAC value $\mathrm{MAC}_K(M)$ should be easy. However, it should be computationally infeasible to find $\mathrm{MAC}_K(M)$ without knowing K. Security properties of MAC function include:

Existential Forgery An adversary is able, without initial knowledge of K, to get a corresponding MAC C for any message M, which has not been MACed by the legitimate MAC generator. The message M may not have any particular meaning.

Selective Forgery An adversary is able to determine the MAC for a message of his choice.

Second Preimage Resistance Second preimage resistance is sometimes referred to as weak collision resistance. If an adversary observes M and the corresponding MAC C, constructing a message $M' \neq M$ with $\mathrm{MAC}_K(M') = C$ should be computationally infeasible for him. The relation $\Pr[\mathrm{MAC}_K(M') = C] = 2^{-l_m}$ should hold in this case, where l_m is the length of the tag.

Universal forgery An adversary is able to find a MAC for every given message. This attack is much more powerful than previous cases.

Key Recovery Attack A key recovery attack is more devastating than forgery. In this case an adversary is able to recover K itself, and thus can perform arbitrary forgeries. Ideally, any attack allowing key recovery requires about 2^{l_k} operations. Verification of such an attack requires $\lceil \frac{l_k}{l_m} \rceil$ text-MAC pairs.

Related Work CBC-MAC [5], a technique for constructing a message authentication code from a block cipher through CBC mode, was first introduced for messages with fixed length by Bellare et. al. Variants of the CBC-MAC for variable length messages were then proposed, examples include EMAC [19], ECBC, FCBC, XCBC [4], TMAC [13], OMAC[12]and CMAC[15]. In [9], Daemen and Rijmen introduced a new MAC construction Alred and its special instance Alpha-MAC. Recently some provably secure MACs from differentially-uniform permutations was brought forward by Minematsu and Tsunoo [14]. Dodis et.al. [10] introduced a new mode of operation for block ciphers and length-preserving MACs.

In [17], Preneel and van Oorschot proposed a general forgery attack on all iterated MACs using the birthday paradox [23]. In [18] they presented a key recovery attack on the retail MAC based on DES, which requires $2^{32.5}$ known text-MAC pairs and $3 \cdot 2^{56}$ off-line computations to find the 112-bit key. Knudesn

[11] presented a forgery attack on CBC-MAC based on n-bit block cipher with truncation to l_m bit tag, with $2^{1+(n-l_m)/2}$ known text-MAC pairs and two known texts. In [8], Coppersmith and Mitchell proposed the key recovery attack against the DesMAC. This attack was further improved by Coppersmith, Knudsen, and Mitchell [7]. The new attack applies to DesMAC involves prefixing the data to be MACed with its length. Recently, new techniques to identify the underlying hash functions of MACs were developed Wang et. al. in [21,20]. In [21], Wang et. al. presented distinguishing attacks on HMAC/NMAC-MD5 and MD5-MAC, partial subkey recovery of MD5-MAC can be achieved. Their distinguisher makes use of internal near-collisions, which leaks more information than internal collisions. A distinguisher based on the internal near-collision was shown in [22]. Based on that, a forgery attack on Alred construction and equivalent subkey recovery attack on its AES-based Instance Alpha-MAC was suggested.

Our Contribution This paper explores a new distinguishing attack on MACs, which can be used to distinguish MAC based on block ciphers under CBC mode from random functions. This distinguishing attack is useful in second preimage attack, which is the main topic of this paper. Instead of exhaustive search, our second preimage attack can be achieved with birthday attack complexity. This reveals serious weakness in practice which has not been noted by the designers of the iterate block cipher based MACs. We show that MACs based on a block cipher in CBC mode and its variants are all vulnerable to weak collision attack. The attack also applies to the recent proposed provably secure MACs from differentially-uniform permutations [14] and the MAC suggested in [10].

Our approach is to utilize the CBC structure and turn the complexity of the second preimage attack into birthday attack complexity. More specifically, given a two-block message $x_1\|x_2$, we want to create another message $x'_1\|x'_2$ with the same MAC, i.e., $E_K(x'_1)\oplus x'_2 = E_K(x_1)\oplus x_2$. From the CBC operation, we get $E_K(x_1)\oplus x'_2 = E_K(x'_1)\oplus x_2$. We are able to choose x'_1, x'_2 at random to get a collision–$\text{MAC}_K(x_1\|x'_2)$ $=\text{MAC}_K(x'_1\|x_2)$. In particular, we explore the second preimage attack on the CBC-MAC [5], EMAC [19], ECBC, FCBC, XCBC [4], TMAC [13], OMAC[12], CMAC[15], PC-MAC and MT-MAC[14], etc.

Organization of the Paper. This paper is organized as follows. In section 2, after basic notations are reviewed, we give brief descriptions of some MACs based on the CBC encryption mode for block cipher. Section 3 shows a new method for distinguishing MAC based on block cipher under CBC mode from a random function. A second preimage attack on the CBC-like MACs is introduced in the section 4. Finally, we summary our results in Section 5.

2 Preliminaries and Notations

In this section, we first list some notations used in this paper and then give some brief descriptions of the related MAC functions.

2.1 Notations

x_i	:	the ith block of the message
y_i	:	the internal state after ith iteration
K	:	the secret key
C	:	the output of MAC taking secret key K and message M as input
$\|M\|$	:	the length of M
0^i	:	the strings of i 0s.
l_m	:	the length of the MAC output
$M\|N$	:	the concatenation of M and N
Σ^n	:	$\{0,1\}^n$, the set of all strings of length n
$\lceil a \rceil$	:	the smallest integer greater than or equal to a
$\lfloor a \rfloor$	:	the greatest integer less than or equal to a

2.2 The CBC-like MAC Functions

MAC functions based on block ciphers are of great practical significance. CBC-MAC [5,6] is a well-known method to generate a MAC based on a block cipher in CBC mode. In general, we will call MAC functions based on block cipher in CBC encryption mode the *CBC-like* MAC functions.

CBC-MAC is used to compress the message M of a fixed length mn with a secret key K, where n is the length of a block and m is the number of blocks. See Fig. 1. More precisely, the CBC-MAC is defined as:

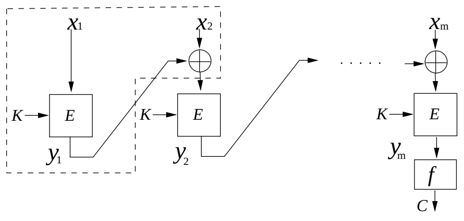

Fig. 1. CBC-MAC Algorithm

$$
\begin{aligned}
&y_0 = 0,\\
&y_i = E_K(y_{i-1} \oplus x_i), 1 \le i \le m,\\
&CBC_K(M) = f(y_m),
\end{aligned}
$$

where f is the truncation function. For messages with fixed length, Bellare, Kilian, and Rogaway [5] established the security of the CBC-MAC. However, a well known fact says that the CBC-MAC without truncation is not secure for variable-length messages. In fact, suppose that an adversary has known $C = CBC_K(M)$, $C' = CBC_K(M')$, then for any single block Y, the messages $M\|Y$ and $M'\|(Y \oplus (C \oplus C'))$ have the same MAC. Several variants of CBC-MAC have been proposed for variable length messages in order to counter this attack.

EMAC Proposed by Berrendschot et. al., EMAC is the first encrypted MAC [3]. It is obtained by encrypting the CBC-MAC value by E again with a new key K_2. That is,

$$EMAC_{K_1,K_2}(M) = E_{K_2}(CBC_{K_1}(M)),$$

where K_1 is the key of the CBC-MAC and $CBC_{K_1}(M)$ is the CBC-MAC value of M. Petrank and Rackoff [19] proved that EMAC is secure if the message length is a positive multiple of n. It is remarked that, however, EMAC requires two key scheduling of the corresponding block cipher E. The ANSI retail MAC [2], a variant of EMAC, selects as its output transformation a decryption D with a second key K_2 followed by an encryption with K_1, (such that the last block undergoes a two key triple encryption):

$$f(y_m) = E_{K_1}(D_{K_2}(y_m)) = E_{K_1}(D_{K_2}(E_{K_1}(y_{m-1} \oplus x_m)))$$

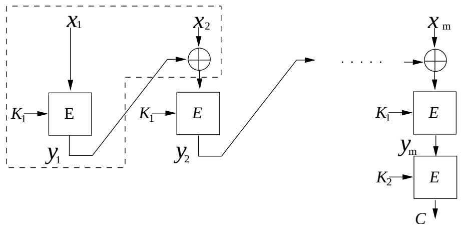

Fig. 2. EMAC Algorithm

XCBC Black and Rogaway [4] suggested some simple variants of the CBC-MAC i.e., ECBC, FCBC and XCBC. These modified CBC-MACs can be used to MAC messages of arbitrary lengths efficiently. Their favorite construction is XCBC which requires only one key scheduling of the built-in block cipher E. In general, XCBC takes three keys: one block cipher key K_1, and two n-bit keys K_2 and K_3. Fig. 3 depicts the process of XCBC. If the message length is the multiple of n, let $K = K_2$, $P = M$. Otherwise, let $K = K_3$, $P = M\|10^i$, where $i = n - 1 - (|M| \bmod n)$. Write $P = x_0\|x_1\| \cdots \|x_t$, $y_0 = 0$. Then $y_i = E_{K_1}(x_i \oplus y_{i-1})$, for $i = 1, \ldots, t-1$. The XCBC value is $C = E_{K_1}(x_t \oplus y_{t-1} \oplus K)$.

TMAC Kurosawa and Iwata introduced Two-key CBC-MAC (TMAC) in [13]. TMAC takes two keys, $(l_k + n)$ bits in total: a block cipher key K_1 of l_k bits and a key K_2 of n bits. TMAC is obtained from XCBC by replacing (K_2, K_3) with $(K_2 \cdot u, K_2)$, where u is some non-zero constant and "$\cdot$" denotes multiplication in the field $GF(2^n)$.

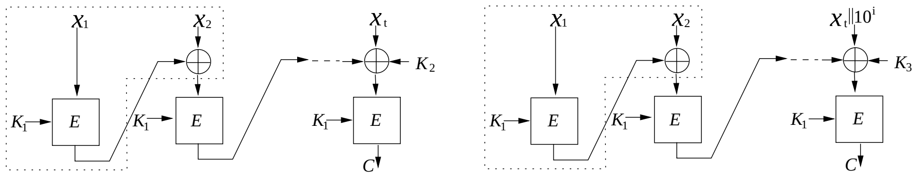

Fig. 3. XCBC MAC Algorithm

OMAC OMAC, also proposed by Iwata and Kurosawa [12], is a generic name for OMAC1 and OMAC2. OMAC1 is obtained from XCBC by replacing (K_2, K_3) with $(L \cdot u, L \cdot u^2)$ for some non-zero constant u in $GF(2^n)$, where L is given by $L = E_K(0^n)$. OMAC2 is constructed in a similar manner by using $(L \cdot u, L \cdot u^{-1})$. Note that $L \cdot u, L \cdot u^{-1}$ and $L \cdot u^2 = (L \cdot u) \cdot u$ can be computed efficiently by one shift and one conditional XOR from L, L and $L \cdot u$, respectively.

CMAC CMAC [15] is equivalent to OMAC1, which is recommended by NIST.

PC-MAC and MT-MAC PC-MAC and MT-MAC was proposed by by Minematsu and Tsunoo [14] for more performance. To facilitate the description of the algorithm, let $G_U = (G_{U_1}, G_{U_2}, \ldots, G_{U_d})$ be the sequence of the so called *auxiliary permutations* with corresponding key $U = (U_1, U_2, \cdots, U_d)$, where for $1 \le i \le d$, $U_i \in \Sigma^{l_U}$ and G_{U_i} is permutation with key U_i. G_{U_i} can be abbreviated to G_i unless it is confused. d is called the interval. Let $F^{\oplus K}(\alpha) = F(K \oplus \alpha)$.
PC-MAC PC-MAC, a very efficient periodic CBC-like construction. To describe PC-MAC, let

$$Ch[F_1, \ldots, F_m](x) = x_{m+1} \oplus F_m(x_m \oplus F_{m-1}(\cdots F_2(x_2 \oplus F_1(x_1)) \cdots),$$

where for each $i = 1, \cdots, m$, F_i is an n-bit block keyed function. Assume that $(d-1)$ n-bit keys, denoted by $K_1^{xor}, \ldots, K_{d-1}^{xor}$, are available. Then the Periodic CBC Hash (PCH) with interval d is a keyed function: $(\Sigma^n)^+ \to \Sigma^n$ defined as:

$$PCH_d[E_K, G_U] = Ch[E_K, G_1, G_2^{\oplus K_1^{xor}}, \ldots, G_d^{\oplus K_{d-1}^{xor}}].$$

Here, $PCH_d[E_K, G_U]$ terminates as soon as the last input block is XORed.

The PC-MAC$_d[E_K, L|G_U](M)$ is defined as follows, where the message is M, the key is (K, L), G_U is the auxiliary permutations with interval d.

Preprocessing Let $U = (U_1, U_2, \cdots, U_d)$ be the first dl_U bits of $E_K^{\oplus L}(0), \ldots, E_K^{\oplus L}(a_1)$, where $a_1 = \lfloor dl_U/n \rfloor$. Let $K_j^{xor} = E_K^{\oplus L}(j + a_1)$ for $j = 1, \ldots, d-1$.
Tag Computation

$$\text{PC-MAC}_{K,L}(M) = \begin{cases} E_K(PCH_d[E_K, G_U](M) \oplus L \cdot u), & \text{if } |M| \text{ is a multiple of } n; \\ E_K(PCH_d[E_K, G_U](M\|10^i) \oplus L \cdot u^2), & \text{otherwise.} \end{cases}$$

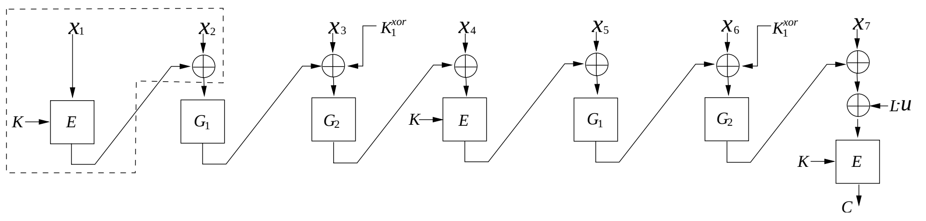

Fig. 4. PC-MAC with $d = 2$ and $M = x_1 \cdots x_7$.

Here, u is an element of $GF(2^n)$ and $i = n - 1 - (|M| \mod n)$. See Fig.4 for an example of PC-MAC$_2[E_K, L|G_U](M)$, and $M = x_1\| \cdots \|x7$.

MT-MAC MT-MAC is an efficient MAC with provable security based on MTH (modified tree hash). (See Fig. 5.) The precise description of MTH for the binary case is give as follows:

Let $H = (H_1, H_2, \ldots,)$ be an infinite sequence of keyed functions: $(\Sigma^n)^2 \to \Sigma^n$. Let $X = (x_1\| \cdots \|x_m) \in (\Sigma^n)^m$. For all $i \geq 1$, let L_{H_i} be a function defined as:

$$L_{H_i}(X) = \begin{cases} H_i(x_1, x_2)\|H_i(x_3, x_4)\| \cdots \|H_i(x_{m-1}, x_m), & \text{if } m \mod 2 = 0; \\ H_i(x_1, x_2)\|H_i(x_3, x_4)\| \cdots \|H_i(x_{m-1}, x_m)\|x_m, & \text{if } m \mod 2 = 1; \end{cases}$$

The output of the MTH using H for input X is

$$MTH_H(X) = L_{H_b} \circ L_{H_{b-1}} \circ \cdots \circ L_{H_1}(X),$$

where $b = \lceil \log_2 m \rceil$ and $\circ$ denotes the serial composition (i.e., $F_2 \circ F_1(x) = F_2(F_1(x))$).

Now give a description of MT-MAC$[E_K|G_U](M)$, where the key is K and G_U is the auxiliary permutations with interval $b = \lceil log_2(\lceil |M|/n \rceil) \rceil$.

Preprocess Let $a_2 = \lceil bl_U/n \rceil$, $U = (U_1, U_2, \cdots, U_b)$ be the first bl_U bits of $E_K(1), \ldots, E_K(a_2)$

Tag Computation Let

$$H_i(\alpha, \beta) = G_{U_i}(\alpha \oplus E_K(i + a_2)) \oplus \beta), \text{for} 1 \leq i \leq b;$$

$$\text{MT-MAC}_K(M) = \begin{cases} E_K(MTH_H(M) \oplus L \cdot u), & \text{if } |M| \text{ is a multiple of } n; \\ E_K(MTH_H(M\|10^i) \oplus L \cdot u^2), & \text{otherwise.} \end{cases}$$

Here, $L = E_K(0)$, u is an element of $GF(2^n)$ and $i = n - 1 - (|M| \mod n)$.

Three-Key Enciphered CBC Mode Three-key enciphered CBC mode [10], a variable input length MAC, was constructed using the preserving fixed input

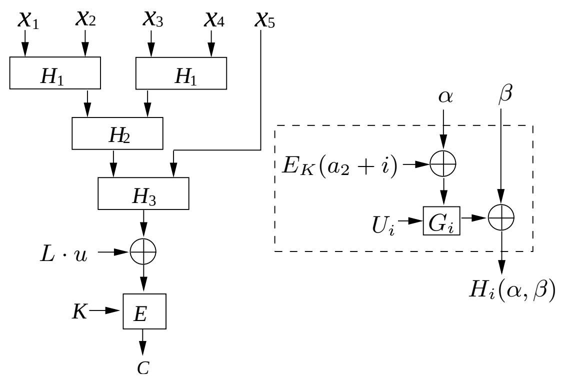

Fig. 5. MT-MAC with $M = x_1 \| \cdots \| x_5$.

length MAC by Dodis et.al. First some auxiliary definitions are given. For $f_1, f_2 : \Sigma^n \to \Sigma^n$, let $g[f_1, f_2](x_2 \| x_1) = f_1(x_1) \oplus f_2(x_2)$. The compression function $G[f_1, f_2]$ is defined as

$$G[f_1, f_2](x_1 \| \cdots \| x_t) = g[f_1, f_2](x_t \| g[f_1, f_2](\cdots g[f_1, f_2](x_2 \| x_1) \cdots)).$$

The three key enciphered CBC construction (See Fig. 6), $H[f_1, f_2, f_3]$, uses three length-preserving functions $f_1, f_2, f_3 : \Sigma^n \to \Sigma^n$ and takes a variable-length input $M = x_1 \| \cdots \| x_t$ (w.l.o.g., we assume the length to be a multiple of n; if not, then append a 1 followed by 0s to achieve this). The mode is defined as:

$$H[f_1, f_2, f_3](x_1 \| \cdots \| x_t) = f_3(G[f_1, f_2](x_1 \| \cdots \| x_t)).$$

Enhanced enciphered CBC suggested in [10] which can be used for the "un-

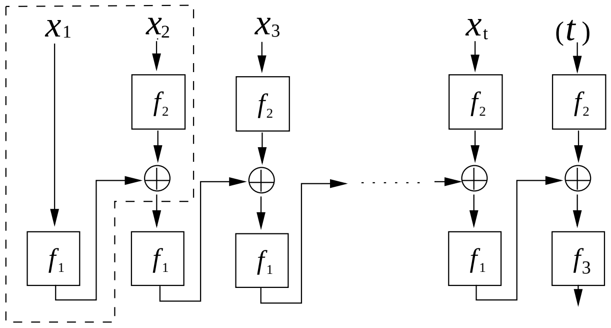

Fig. 6. The basic three-key enciphered CBC mode

keyed" settings (RO and CRHF) still works for the "keyed" settings (PRF and MAC). The mode is depicted in Fig. 7 and is denoted $H^*[\pi_1, \pi_2, \pi_3]$, which is the basic enciphered CBC mode $H[f_1, f_2, f_3]$ with length-preserving functions f_1, f_2, f_3 defined as follows:$f_i = \pi_i(x) \oplus x$ for i=1, 2 and $f_3(x) = \pi_3(x) \oplus \pi_3^{-1}(x)$.

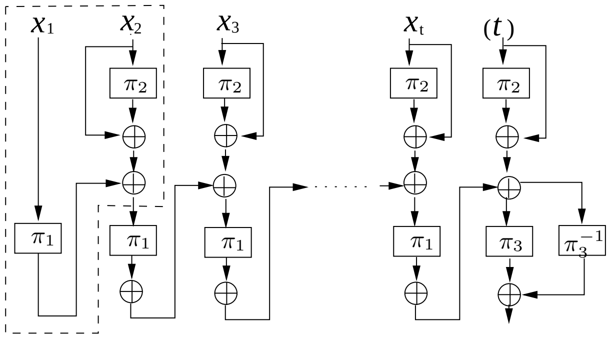

Fig. 7. The enhanced three-key enciphered CBC mode

2.3 Related Attacks

Finally in this section, we give several attacks that are closely related to our attacks. Preneel and van Oorschot proposed a general forgery attack on all iterated MACs [17]. Their idea is that the adversary explores a pair of messages (M, M') with the same MAC using the birthday attack. Then he appends the same blocks in the end of the message pair to recognize whether this is an *internal collision* (whose values are equal before the last transformation) or not. If it is an internal collision, the adversary can use MAC of $M\|y$ to forge the MAC of $M'\|y$.

In [18], Preneel and van Oorschot presented a key recovery attack on the retail MAC based on DES. The attack requires $2^{32.5}$ known text-MAC pairs and $3 \cdot 2^{56}$ off-line computations to find the 112-bit key. Firstly they search an internal collision by the birthday attack, and then filter the key K with the collision. At last the K_2 can be obtain by an exhaustive search. Knudsen and Preneel [16] suggested another key recovery attack, which requires only one known text-MAC pair. However it needs 2^{55} online MAC verifications and $2 \cdot 2^{56}$ DES encryptions. Suppose that the adversary has observed a MAC $C = f(\text{MAC}_{K_1,K_2}(x_1\|x_2\| \cdots \|x_m))$. He can create a message $M' = (x_1'\|x_2'\|x_3\| \cdots \|x_m)$, guess the key K_1, and compute $x_2' = x_2 \oplus E_{K_1}(x_1) \oplus E_{K_1}(x_1')$. If the guess for K_1 is correct, then this choice gives $y_2 = y_2'$. Therefore $f(\text{MAC}_{K_1,K_2}(M')) = C$, i.e. the verification will result in a positive answer. On average, K_1 will be found after 2^{k-1} attempts. Then K_2 can be obtained by an exhaustive search.

3 Distinguishing Attack on the MACs Based on Block Cipher

In this section, we consider the non-randomness of MACs based on block ciphers in CBC mode. A new distinguisher is constructed based on our observation.

Recall that Preneel and van Oorschot [17] proposed a general forgery attacks on the iterate MACs, which can be used to distinguish the MAC functions from a random function. Our new method to distinguish the block cipher-based MAC from a random function relies on that the built-in block cipher being a permutation.

The following is our observation.

Proposition 1. *Let $n, m \geq 2$ be integers. Given the block cipher E: $\Sigma^n \rightarrow \Sigma^n$ and a secret key K, the CBC-MAC values of $x_1 \| x_2 \| \cdots \| x_m$ and $x_1' \| x_2 \| \cdots \| x_m$ must be different provided $x_1 \neq x_1'$, where x_1' and $x_i, i = 1, 2, \ldots, m$ are single-block messages.*

Proof. Write
$$y_1 = E_K(x_1) \text{ and } y_1' = E_K(x_1').$$
It is immediate that $y_1 \neq y_1'$ since $E_K(\cdot)$ is a permutation on Σ^n.

For $i = 2, \ldots, m$, y_i and y_i' can be recursively defined as
$$y_i = E_K(x_i \oplus y_{i-1}) \text{ and } y_i' = E_K(x_i \oplus y_{i-1}').$$
Similarly, we note that $y_i \neq y_i'$ for $i = 2, \ldots, m$. In particular, $y_m \neq y_m'$ can be deduced, and hence
$$C \neq C'.$$
□

To perform the distinguishing attack, the adversary is given an oracle $C = CBC_K(\cdot)$. He makes $2^{(n+1)/2}$ queries with m-block messages that have the same last $m-1$ blocks. The following is the explicit structure S:
$$S = \{\, M^i \mid M^i = x_1^i \| x_2 \| \cdots \| x_m, i = 1, 2, \ldots, 2^{(n+1)/2} \,\},$$
where x_1^i's are distinct random single-block messages. If there is a collision in the structure, the MAC function is a random function. Otherwise it is the CBC-MAC.

This attack requires about $2^{(n+1)/2}$ chosen messages. By the birthday paradox, the probability that the adversary will succeed is 0.63. If we double the number of the messages, the success rate can be increased to 0.98.

This distinguishing technique can also be used to attack other block cipher-based MACs mentioned earlier, such as EMAC, XCBC, TMAC and OMAC. For these cases, we only need messages with one block to query the corresponding oracle.

4 Second Preimage Attack

Second preimage resistance is sometimes referred to as weak collision resistance. It means that if an adversary observes M and the corresponding tag C, it must be computationally infeasible to construct another message M' s.t. $\mathrm{MAC}_K(M')$ $= C$. Ideally one would expect the complexity of finding such M' to be 2^{l_m}. However, as we shall prove in this section, most of the block cipher-based MACs are vulnerable to the second preimage attack.

In this section, we will demonstrate how the complexity of the second preimage attack on some MACs can be reduced to birthday attack complexity.

4.1 A General Statement

In this section, we propose a new method for constructing a message that is different from a given message (with at least two blocks) but with the same MAC. This task can be achieved with birthday attack complexity. We also identify three types of building blocks for MAC iterate construction that are vulnerable to our attack.(See Fig. 8)

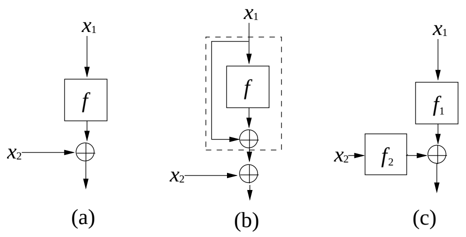

Fig. 8. Three types of insecurity elements to MAC iterate construction

Our first result in this section is the following.

Proposition 2. *Let $b \geq n$ and $f_1,\ f_2 : \Sigma^b \rightarrow \Sigma^n$ be two random maps. Set $g(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$. Then with probability* 0.63*, a pair (x_1', x_2') can be obtained such that $g(x_1, x_2) = g(x_1', x_2')$, with (birthday attack) complexity* $2^{(n+3)/2}$.

To facilitate the proof of the proposition, let us first present the algorithm to construct (x_1', x_2').

Algorithm 1. Find another pair (x_1', x_2') to make $g(x_1', x_2') = g(x_1, x_2)$.

```
  INPUT   :  $x_1$, $x_2$, $g$.
 OUTPUT   :  $x_1'$, $x_2'$.
  1.   $S_1 \leftarrow \emptyset$
  2.   For $i \leftarrow 0$ to $2^{(n+1)/2}$ do
           Choose $x_1^i \notin \{x_1, x_1^0, \ldots, x_1^{i-1}\}$ at random and compute $y_1^i \leftarrow g(x_1^i, x_2)$
           $S_1 \leftarrow S_1 \bigcup \{(x_1^i, y_1^i)\}$
       End For
  3    For $i \leftarrow 0$ to $2^{(n+1)/2}$ do
           Choose $x_2^i \notin \{x_2, x_2^0, \ldots, x_2^{i-1}\}$ at random and compute $y_2^i \leftarrow g(x_1, x_2^i)$
           If $y_2^i = y_1^k$ where $y_1^k$ is the second component of an element of $S_1$
               Return $(x_1^k, x_2^i)$
       End For
```

Now let us prove the proposition.

Proof. To show that this is a success attack, we need to check two things. First, the forgery is valid, i.e. $g(x_1', x_2') = g(x_1, x_2)$. Second, the messages are different, i.e., $(x_1', x_2') \neq (x_1, x_2)$.

If a pair $(x_1',\ x_2')$ is obtained from the Algorithm 1, $(x_1', x_2') \neq (x_1, x_2)$ is obvious.

Next, we prove $g(x_1',\ x_2') = g(x_1,\ x_2)$. By the algorithm, we have $g(x_1', x_2) = g(x_1, x_2')$, i.e.,

$$f_1(x_1') \oplus f_2(x_2) = f_1(x_1) \oplus f_2(x_2').$$

This implies that $f_1(x_1') \oplus f_2(x_2') = f_1(x_1) \oplus f_2(x_2)$, and hence

$$g(x_1', x_2') = g(x_1, x_2).$$

Step 2 needs $2^{(n+1)/2}$ computations of the function g, so does the Step 3. Therefore, the complexity of the algorithm is $2^{(n+3)/2}$. The success possibility is

$$1 - \prod_{i=1}^{2^{(n+1)/2}-1} (1 - \frac{i}{2^n}) \approx 1 - e^{-1} \approx 0.63.$$

□

Remark. Fig 8 (a) and (c) depict the case $f_2(x) = x$ and general case respectively.

4.2 The Second-Preimage Attack on CBC-MAC

The following is a straightforward statement whose proof is omitted.

Proposition 3. *Let $n, m \geq 2$ be integers. Let E: $\Sigma^n \to \Sigma^n$ be the block cipher and K the secret key for the CBC-MAC. Then the CBC-MAC values of $x_1 \| x_2 \| x_3 \| \cdots \| x_m$ and $x_1' \| x_2' \| x_3 \| \cdots \| x_m$ are the same if and only if $E_K(x_1) \oplus x_2 = E_K(x_1') \oplus x_2'$, where x_1', x_2' and $x_i, i = 1, 2, \ldots, m$ are single-block messages.*

Suppose that the adversary has an oracle $C = CBC_K(\cdot)$, which can be used to query any message. He has intercepted a message $M_0 = x_1\|x_2\|\cdots\|x_m$ and its MAC is C_0. A collision is shown in Fig. 9.

The following procedure can be used to forge the MAC C_0 by finding another message M'.

1. Construct two structures

$$S_1 = \{\, M_1^i \mid M_1^i = x_1\|x_2^i\|x_3\|\cdots\|x_m, i = 1, \ldots, 2^{(n+1)/2}\,\},$$
$$S_2 = \{\, M_2^i \mid M_2^i = x_1^i\|x_2\|x_3\|\cdots\|x_m, i = 1, \ldots, 2^{(n+1)/2}\,\},$$

   where $x_2^1, x_2^2, \ldots$ are distinct random message blocks that are different from x_2, and so are $x_1^1, x_1^2, \ldots$.
2. Query the oracle for the messages in the two structures and obtain $C_1^i = CBC_K(M_1^i)$, $C_2^i = CBC_K(M_2^i)$, where i=1, 2, ..., $2^{(n+1)/2}$.
3. A collision can be obtained by the birthday attack. Assume w.l.o.g. that $C_1^k{=}C_2^j$. By Proposition 3,

$$E_{K_1}(x_1) \oplus x_2^j = E_{K_1}(x_1^k) \oplus x_2,$$

   then

$$E_{K_1}(x_1) \oplus x_2 = E_{K_1}(x_1^k) \oplus x_2^j.$$

   So the message $M' = x_1^k\|x_2^j\|x_3\|\cdots\|x_m$ has the same MAC value with the message M.

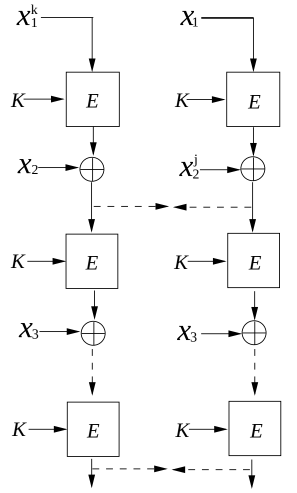

Fig. 9. A Collision of the CBC-MAC

It is not difficult to see that our technique still works for the following structures

$$S_1 = \{\, M_1^i \mid M_1^i = x_1 \| \cdots \| x_{j-1} \| x_j^i \| x_{j+1} \| \cdots \| x_m,\ i = 1, \ldots, 2^{(n+1)/2} \,\},$$

where $x_j^1, x_j^2, \cdots$ are distinct random message blocks that are different from x_j; and

$$S_2 \subset \{\, M' \mid M' = x_1' \| \cdots \| x_{j-1}' \| x_j \| x_{j+1} \| \cdots \| x_m,\ x_1', \ldots, x_{j-1}' \text{ are random} \,\},$$

with $\#S_2 = 2^{(n+1)/2}$. S_2 should not contain $x_1 \| \cdots \| x_{j-1} \| x_j \| x_{j+1} \| \cdots \| x_m$.

A collision can be obtained by the birthday paradox, i.e. $E_K(y_{j-1} \oplus x_j^k) = E_K(y_{j-1}' \oplus x_j)$, thus $E_K(y_{j-1} \oplus x_j) = E_K(y_{j-1}' \oplus x_j^k)$. The messages $x_1' \| \cdots \| x_{j-1}' \| x_j^k \| x_{j+1} \| \cdots \| x_m$ and M has the same tag.

This attack is apparently working for the following MACs with arbitrary message length such as EMAC, XCBC, TMAC, OMAC, CMAC, PC-MAC and MT-MAC.

We would like to point out that above attack can be applied CFB-MAC[1]. For this, we just observe that the iteration is defined as follows:

$$y_i = f(y_{i-1}, x_i) = E_K(y_{i-1}) \oplus x_i.$$

4.3 The Second Preimage Attacks for Other Constructions

Our attack applies to other constructions as well.

It is obvious that PC-MAC is a variant of CBC-MAC, so the attack to the PC-MAC still works.

Note that the function of H_i of MT-MAC contains the basic building blocks in Fig. 8. Therefore it is vulnerable to the second preimage attack by Proposition 2.

Our attack can be also used to the three-key encipher CBC mode. When f_1 and f_3 are permutations, it is obvious that three-key encipher CBC mode is a variant of CBC-like MAC, with message transform function $f_2(x)$ instead of x. So it is not resistant to the second preimage attack. If f_1 and f_3 are compress functions, then we need to decide where the collision comes out. The internal collision is useful to the second preimage attack.

We construct the structures as:

$$S_1 = \{\, M_1^i \mid M_1^i = x_1 \| x_2^i, i = 1, \ldots, 2^{(n+1)/2} \,\},$$
$$S_2 = \{\, M_2^i \mid M_2^i = x_1^i \| x_2, i = 1, \ldots, 2^{(n+1)/2} \,\},$$

By birthday paradox, there exist two collisions: an internal collision before $f3$ function and an external collision (not internal collision). Assume w.o.l.g that the internal collision is MAC(M_1^j) =MAC(M_2^k). Then the messages $x_1 \| x_2^j \| x_3$ and $x_1^k \| x_2 \| x_3$ have the same MAC, which is not the case if the pair (M_1^j, M_2^k) is an external collision. By the equitation $f_1(x_1) \oplus f_2(x_2^j) = f_1(x_1^k) \oplus f_2(x_2)$, we can deduced $f_1(x_1) \oplus f_2(x_2) = f_1(x_1^k) \oplus f_2(x_2^j)$. Namely MAC($x_1^k \| x_2^j$) =MAC($x_1 \| x_2$). The success probability is 0.63. Of course the success rate can be improved by enlarging the message structures.

To determine whether the underlying functions are permutations or compressing functions, we can use the distinguishing technique described in Section 3.

5 Conclusion

This paper first explores a new distinguishing attack on CBC-like MACs. The distinguisher can decide weather the MAC function is based on block ciphers or random functions. The main results of this paper is to propose a second preimage attack on a variety MACs based on block cipher with CBC mode. In particular, we prove that the MACs such as CBC-MAC [5], EMAC, ECBC, FCBC, XCBC [4], TMAC [13], OMAC[12], CMAC[15], PC-MAC, MT-MAC[14], three-key encipher CBC mode [10] etc. are all vulnerable to second preimage attack.

References

1. ANSI X9.9 (revised), Financial Institution Message Authentication (wholesale), Amer. Bankers Assoc., Apr. 7, 1986.
2. ANSI X9.19, Financial Institution Retail Message Authentication, American Bankers Association, August 13, 1986.
3. A. Berendschot, B. den Boer, J. P. Boly, A. Bosselaers, J. Brandt, D. Chaum, I. Damgard, M. Dichtl, W. Fumy, M. van der Ham, C. J. A. Jansen, P. Landrock, B. Preneel, G. Roelofsen, P. de Rooij, and J. Vandewalle, Final Report of RACE Integrity Primitives, LNCS 1007, 1995.
4. J. Black, P. Rogaway, CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions, CRYPTO 2000, LNCS 1880, pp. 197-215, 2000.
5. M., Bellare, J., Kilian, and P. Rogaway, The Security of the Cipher Block Chaining Message Authentication Code, CRYPTO'94, LNCS 839, pp. 341-358, 1994.
6. M., Bellare, J., Kilian, and P. Rogaway, The Security of the Cipher Block Chaining Message Authentication Code, Journal of Computer and System Sciences, vol. 61, no.3, pp.362-399, 2000.
7. D. Coppersmith, L. R. Knudsen, and C. J. Mitchell, Key Recovery and Forgery Attacks on the MacDES MAC Algorithm, CRYPTO 2000, LNCS 1880, pp. 184-196, 2000.
8. D. Coppersmith and C.J. Mitchell, Attacks on MacDES MAC algorithm, Electronics Letters, 35:1626-1627, 1999.
9. J.Daemen, V. Rijmen, A New MAC Construction ALRED and A Specific Instance ALPHA-MAC, FSE 2005, LNCS 3557, pp.1-17, 2005.
10. Y. Dodis, K. Pietrzak, and P. Puniya, A New Mode of Operation for Block Ciphers and Length-Preserving MACs, EUROCRYPT 2008, LNCS 4965, pp. 198-219, 2008.
11. L.R. Knudsen, Chosen-text Attack on CBC-MAC, Electronic Letters, vol.33, no.1, 1997.

12. T. Iwata, K. Kurosawa, OMAC: One-Key CBC MAC, FSE 2003, LNCS 2887, pp. 129-153, 2003.
13. K. Kurosawa, T. Iwata, TMAC: Two-Key CBC MAC, CT-RSA 2003, LNCS 2612, pp. 265-273, 2003.
14. K. Minematsu and Y. Tsunoo, Provably Secure MACs from Differentially-Uniform Permutations and AES-Based Implementations, FSE 2006, LNCS 4047, pp. 226-241, 2006.
15. NIST, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, NIST Special Publication 800-38B, 2005.
16. B. Preneel and L.R. Knudsen, MacDES: MAC algorithm based on DES, Electronic Letters, vol.33, no.1, 1997.
17. B. Preneel, P. C. van Oorschot, MDx-MAC and Building Fast MACs from Hash Functions. CRYPTO 1995, LNCS 963, pp. 1-14, 1995.
18. B. Preneel and P.C. van Oorschot, Key Recovery Attack on ANSI X9.19 Retail MAC, Electronic Letters, vol.32, no.17, 1996.
19. E. Petrank and C. Rackoff, CBC MAC for Real-Time Data Sources. J.Cryptology, vol. 13, no. 3, pp. 315-338, Springer-Verlag, 2000.
20. X. Wang, W. Wang, K. Jia, M. Wang, New Distinguishing Attack on MAC using Secret-Prefix Method, submitted to FSE 2009.
21. X. Wang, H. Yu, W. Wang, H. Zhang, T. Zhan, Cryptanalysis on HMAC/NMAC-MD5 and MD5-MAC, submitted to EUROCRYPT 2009.
22. Z. Yuan, K. Jia, W. Wang, X. Wang, Distinguishing and Forgery Attacks on Alred and Its AES-based Instance Alpha-MAC, http://eprint.iacr.org/2008/516.
23. G. Yuval, How to Swindle Rabin, *Cryptologia*, vol. 3, pp. 187-189, 1979.