# Distinguishing Attacks on MAC/HMAC Based on A New Dedicated Compression Function Framework *

Zheng Yuan[1,2],** Xiaoqiu Ren[1], Jintao Liu[1]

[1] Beijing Electronic Science and Technology Institute, Beijing 100070,China
[2] Center for Advanced Study, Tsinghua University, Beijing 100084,China
yuanzheng@besti.edu.cn
zyuan@mail.tsinghua.edu.cn

**Abstract.** By the birthday attack, a new distinguisher with an inner partial collision is first presented. Using the distinguisher can attack on MAC/HMAC based on a dedicated compression function framework proposed in ChinaCrypt2008, with $2^{16.5}$ data complexity and $2^{16.5}$ MAC queries. More important, using the new distinguishing attack can recover the secret key of NMAC with the data complexities of $2^{16.5}$.
**Keywords:** distinguishing attacks, an inner partial collision, a dedicated compression function framework, hash function, MAC, HMAC.

## 1 Introduction

Message Authentication Code (MAC) is a kind of fixed-length information used to ensure data integrity and authenticity. A MAC algorithm takes a secret key and a message of arbitrary length as input, and the output is a short digest. HMAC and NMAC are hash-based message authentication codes proposed by Bellare, Canetti and Krawczyk[1]. NMAC is the theoretical foundation of HMAC, and HMAC has been implemented in widely-used protocols including SSL/TLS, SSH and IPsec. The securities of NMAC and HMAC has been carefully analyzed in [1,3]. It was proved that NMAC is a pseudo-random function family (PRF) under the assumption that the compression function of the keyed hash function is a PRF. This proof can be extended to HMAC by an additional assumption: the key derivation function in HMAC is a PRF. However, if the underlying hash function is weak, the above proofs may not work. How to estimate the security of a hash function?

Recently, NIST proposed the security requirements of the hash function, the details are as follows:

* This work is supported by the Beijing Natural Science Foundation (No. 4102055), the Foundation of State Key Laboratory of Information Security (Institute of Software, Chinese Academy of Sciences)(No.01-01), the Foundation of Key Laboratory of Information Security of BESTI(No.YZDJ0905), and National 973 Program of China (No.2007CB311201)
** To whom correspondence should be addressed.

1. It may be provided in a wide variety of cryptographic applications, including digital signatures, key derivation, hash-based message authentication codes, deterministic random bit generators, and additional applications that may be brought up by NIST or by the public during the evaluation process.
2. It can support HMAC, Pseudo Random Functions (PRFs), and Randomized Hashing.
3. The hash algorithm of message digest size $n$ to meet the following security requirements at a minimum.
   • Collision resistance of approximately $n/2$ bits,
   • Preimage resistance of approximately $n$ bits,
   • Second-preimage resistance of approximately $n$-$k$ bits for any message shorter than $2^k$ bits,
   • Resistance to length-extension attacks, and
   • Any $m$-bit hash function specified by taking a fixed subset of the candidate function's output bits is expected to meet the above requirements with m replacing n.
   Additionally, increasing the second preimage resistance property and resistance against other attacks, such as multicollision attacks.
4. Evaluations relating to attack resistance.

In ChinaCrypt2008, a new dedicated compression function framework (i.e. hash function, later, we simply call the hash function $\boldsymbol{H}$) and two improvement schemes for MD construction were proposed. The compression function is the hardcore of a hash function, the authors proved that the first scheme had the property of pseudo collision resistant and could withstand some attacks making use of the weakness of MD constructions, such as the multi-collision attack. It was also proved that the second scheme with a random number perhaps was securer than the first one. But the hash function $\boldsymbol{H}$ couldn't resist some attacks on MAC/HMAC based on it. Namely, the hash function $\boldsymbol{H}$ didn't satisfy the NIST's requirement criterion 2 above.

The main contribution of this paper is first to present novel distinguishing attacks on the MAC/HMAC based on the hash function $\boldsymbol{H}$, which leads to forgery attacks directly. More important, the distinguishing attack on MAC/HMAC based on the hash function $\boldsymbol{H}$ can be applied to recover the secret key $k$, and hence results in a second preimage attack.

There are two kinds of distinguishing attacks on MACs, which are distinguishing-R and distinguishing-H attacks respectively [6]. Distinguishing-R attack distinguishes a MAC from a random function, and distinguishing-H attack detects an instantiated MAC constructed by an underlying hash function or block cipher from a MAC constructed by a random function. Preneel et al. [11]introduced a general distinguishing-R attack on all iterated MACs by the birthday paradox, which requires about $2^{\frac{n}{2}}$ messages with a success rate of 0.63, where $n$ is the length of the hash output. Their attack can immediately be converted into a general forgery attack. The other kind of attack was suggested by Kim et. al.[6], which distinguishes the cryptographic primitive embedded in a MAC construc-

tion from a random function. This paper focuses on the distinguishing-H attack. For simplicity, we call it distinguishing attack.

In recent works[5,12,14], new distinguishing attacks on MACs were discovered. Under the new distinguishing attack, a forgery attack, a second-preimage attack, a recovery on the internal state or a partial key recovery attack on the MACs might be done. Especially some attacks on MACs based on block ciphers or reduced block ciphers, such as Alred structure and its AES-based instance Alpha-MAC[14], CBC-like MACs[5] including CBC-MAC[2], TMAC[7], OMAC[4], CMAC[9], PC-MAC[10] and MACs with three-key enciphered CBC mode, with the complexity of the birthday paradox, have undermined these MACs' actual securities. Additionally, Wang et al.[13] gave distinguishing attacks on HMAC and NMAC based on MD5 without related keys. All distinguishing attacks above[5,12,13,14] utilized some new techniques to detect an inner near-collision differential paths of the cryptographic primitive, embedded in a MAC/HMAC, by the birthday attack.

In this paper, new distinguishing attacks on MAC/HMAC based on the hash function $\boldsymbol{H}$ are presented. Adopting segmenting techniques, we first propose a new idea to detect a inner partial-collision with only $2^{16.5}$ data, which can be used to identify the hash function $\boldsymbol{H}$, embedded in the HMAC. Furthermore, the distinguishing attack on HMAC can be applied to recover its secret key $k$ with same data complexity.

This paper is organized as follows: Section 2 is backgrounds including brief descriptions of the hash function $\boldsymbol{H}$ and HMAC based on it; In Section 3, we present a partial collision attack on the hash function $\boldsymbol{H}$; A distinguishing attack on MAC based on the hash function $\boldsymbol{H}$ is first suggested in Section 4; In Section 5, we first introduce a distinguishing attack on HMAC constructed by the hash function $\boldsymbol{H}$, then recover its secret key $k$; Finally, our results in Section 6.

## 2 Backgrounds

### 2.1 A Brief Description of the hash function $\boldsymbol{H}$

Firstly, we introduce a new block cipher, which is used to construct the new dedicated compression function. Later, a hash function based on the compression function is recommended.

### 2.2 Block Cipher $\boldsymbol{E}$

The block cipher $E$ is an iteration algorithm possessing 32 same rounds and a 512-bit key, the length of the block is 128-bit. Let $i \in [1, 32]$, given a 32-bit plaintext $B_i = b_{i0} \parallel b_{i1} \parallel b_{i2} \parallel b_{i3}$ and a 32-bit subkey $K_i = k_{i0} \parallel k_{i1} \parallel k_{i2} \parallel k_{i3}$, the $i^{th}$ round iteration operation consists of XOR transformation, SubBytes transformation and RowColumns transformation. The details of the three transformations are as follows:

1. XOR: $B_i \oplus K_i = (b_{i0} \oplus k_{i0}) \| (b_{i1} \oplus k_{i1}) \| (b_{i2} \oplus k_{i2}) \| (b_{i3} \oplus k_{i3})$

2. SubBytes: denote $f \in GF(2^8)$ as a reversible affine transformation. Put four bytes into a S-box respectively, which is similar to the S-box of AES, and the output byte $b_j = f((b_{ij} \oplus k_{ij})^{-1}), (0 \le j \le 3)$.
3. RowColumns: the four bytes multiply a $4 \times 4$ MDS matrix, which is similar to the MixColumn of AES.

$$\begin{pmatrix} b_{i+1,0} \\ b_{i+1,1} \\ b_{i+1,2} \\ b_{i+1,3} \end{pmatrix} = MDS_{4\times4} \times \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = MDS_{4\times4} \times \begin{pmatrix} f((b_{i0} \oplus k_{i0})^{-1}) \\ f((b_{i1} \oplus k_{i1})^{-1}) \\ f((b_{i2} \oplus k_{i2})^{-1}) \\ f((b_{i3} \oplus k_{i3})^{-1}) \end{pmatrix}$$

Thus, the $i^{th}$ round output is $B_{i+1} = b_{i+1,0} \parallel b_{i+1,1} \parallel b_{i+1,2} \parallel b_{i+1,3}$, which is also the $(i+1)^{th}$ round input.

**Key-extension:** The 32-round iteration algorithms require 32 sub-keys. The initial 512-bit key $K$ provides previous 16-round sub-keys, and each sub-key of the following 16 rounds can be computered from the formula: $K_i = (K_{i-3} \oplus K_{i-5} \oplus K_{i-8} \oplus K_{i-11} \oplus K_{i-14} \oplus K_{i-16})^{<<<1}, (i > 15)$

### 2.3 Construct A New Dedicated Compression Function $h$ Using the Block cipher $E$

The compression function (i.e. iteration function) $h$ is a concatenation of eight block ciphers(See Fig.1), whose input are a 256-bit initial vector $IV_i = IV_{i0} \parallel IV_{i1} \parallel IV_{i2} \parallel IV_{i3} \parallel IV_{i4} \parallel IV_{i5} \parallel IV_{i6} \parallel IV_{i7}$ and a 512-bit message block $M_i = B_{i0} \parallel B_{i1} \parallel B_{i2} \parallel B_{i3}$, and output is a 256-bit value $H_i = h(IV, M_i)$, where $H_i = H_{i0} \parallel H_{i1} \parallel H_{i2} \parallel H_{i3} \parallel H_{i4} \parallel H_{i5} \parallel H_{i6} \parallel H_{i7}$.

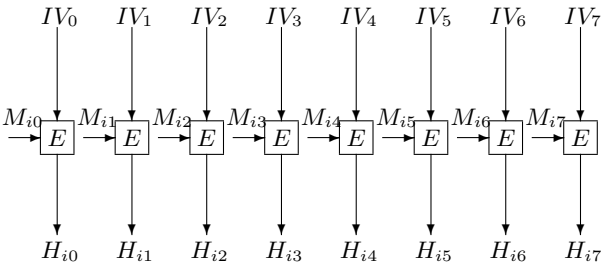


**Fig. 1.** The Construction of the New Dedicated Compression Function $h$

For each block cipher $E$, the input is a 32-bit initial vector component $IV_j, (0 \leq j \leq 7)$ and a 512-bit message expansion block $M_{ij}$, where

$$M_{i0} = M_i = B_{i0} \parallel B_{i1} \parallel B_{i2} \parallel B_{i3} \tag{1}$$
$$M_{i1} = B_{i3} \parallel B_{i0} \parallel B_{i1} \parallel B_{i2} \tag{2}$$
$$M_{i2} = B_{i2} \parallel B_{i3} \parallel B_{i0} \parallel B_{i1} \tag{3}$$
$$M_{i3} = B_{i1} \parallel B_{i2} \parallel B_{i3} \parallel B_{i0} \tag{4}$$
$$M_{i4} = B_{i3} \parallel B_{i2} \parallel B_{i1} \parallel B_{i0} \tag{5}$$
$$M_{i5} = B_{i0} \parallel B_{i3} \parallel B_{i2} \parallel B_{i1} \tag{6}$$
$$M_{i6} = B_{i1} \parallel B_{i0} \parallel B_{i3} \parallel B_{i2} \tag{7}$$
$$M_{i7} = B_{i2} \parallel B_{i1} \parallel B_{i0} \parallel B_{i3} \tag{8}$$

$IV_j$ and $M_{ij}$ are regarded as plaintext $B_i$ and subkey $K_i$, respectively. So the internal state and output of compression function $h$ can be split into eight 32-bit blocks.

### 2.4 Construct the hash function $H$ using the Compression function $h$

For the original MD construction:

$$H_1 = h(M_1, IV_0); H_i = h(M_i, H_{i-1}), (i > 1) \tag{9}$$

### 2.5 MACs based on the hash function $H$

A keyed-hash function is a hash function whose fixed initial vector $IV$ is replaced by a key $k$, i.e. $H_k(M) = H(k, M)$. Let $(k_1, k_2)$ be an independent key pair, according to paper[1], NMAC algorithm is defined as:

$$NMAC_{(k_1,k_1)} = H_{k_1}(H_{k_2}(M)) \tag{10}$$

If

$$k_1 = h(IV, \boldsymbol{k} \oplus ipad) \tag{11}$$
$$k_2 = h(IV, \boldsymbol{k} \oplus opad) \tag{12}$$

HMAC algorithm is defined as:

$$HMAC_k(m) = NMAC_{k_1,k_2}(m) = H_{k_1}(H_{k_2}(m)), \tag{13}$$

where $\boldsymbol{k}$ is obtained by padding number 0 at the end of $k$ to make the length $|\boldsymbol{k}| = b$. Both $ipad$ and $opad$ are $b$-bit constants, and get $ipad$ and $opad$ by repeating concatenating $0x5c$ and $0x36$, respectively.

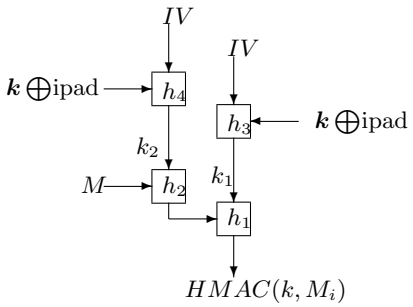


**Fig. 2.** HMAC Based the hash function $\boldsymbol{H}$

## 3 Partial Collision Attack on the hash function $\boldsymbol{H}$

In Fig.1, message block $M_i$ can uniquely determine any message expansion block $M_{ij}(0 \leq j \leq 7)$ by Equ.(1-8). Select a difference $\alpha$, get message block $M_i' = M_i \oplus \alpha$, whose message expansion block $M_{ij}' = M_{ij} \oplus \alpha_j$. Obviously, $(M_{ij}, M_{ij}')$ can be deduced by the message pair $(M_i, M_i')$.

Suppose $t$ message block pairs $(M, M')$, where $M = M_0||M_1||M_2||\ldots||M_{t-1}$ and $M' = M_0'||M_1'||M_2'||\ldots||M_{t-1}'$. Likewise, $(M, M')$ can determine any message expansion block $(M_{ij}, M_{ij}')(0 \leq i \leq t-1, 0 \leq j \leq 7)$.

Choose about $2^{16.5}$ message pairs $(M, M')$ randomly, we can obtain $2^{16.5}$ random message expansion block $(M_{ij}, M_{ij}')$, such as $(M_{t-17}, M_{t-17}')$. Using the hash function $\boldsymbol{H}$ (See Equ.(9)), we can get $2^{16.5}$ hash pairs of $(H_t, H_t')$.

By the birthday attack, at least one pair $(H_{tj}, H_{tj}')(0 \leq j \leq 7)$ leads to partial collision with a data complexity of $2^{16.5}$, for example, $(H_{t7}, H_{t7}')$ is a partial collision.

**Remark 1:** If $t$ is big enough to make all message expansion blocks $(M_{t-1j}, M_{t-1j}')(0 \leq j \leq 7)$ random, $(M, M')$ can obtain at least a collision $(H_t, H_t')$ by $2^{16.5}$ data. It is recommended to choose $t = 9$.

**Remark 2:** If the message pair $(M, M')$ satisfy that their hash values $(H_t, H_t')$ is a collision, then we can regard $M'$ as the second-preimage of $M$.

## 4 Distinguishing Attack on MAC Based on the hash function $\boldsymbol{H}$

For the hash function $\boldsymbol{H}$, using key $k$ instead of initial vector $IV$ can obtain a MAC algorithm:

$$A : \{0,1\}^{256} \times \{0,1\}^{n-512t} \longrightarrow \{0,1\}^{256},$$

whose output and input are completely independent, both of them can be split into eight 32-bit blocks.

Here, we propose a distinguish attack on the MAC algorithm $A$ from a random function, our method relies on that the built-in hash function $\boldsymbol{H}$ having a partial collision with a data complexity of $2^{16.5}$. That is to say, choose randomly about $2^{16.5}$ adaptive message $M_i$, by the birthday attack, we can discover at least one partial collision. The particulars are as follows:

1. For $1 \leq i \leq 2^{16.5}$, randomly select about $2^{16.5}$ messages $M_i = M_{i0}||M_{i1}||\ldots||M_{it-1}$, ask their MAC values $C_i = C_{i0}||C_{i1}||\ldots||C_{it-1}$.
2. Find out all the message pairs $(M_j, M_k)$,$(1 \leq j, k \leq 2^{16.5})$, whose MAC values have at least a partial collision, i.e. $C_{jv} = C_{kv}(1 \leq v \leq 7)$.
3. Given a nonzero string $P$, for each message pair $(M_j, M_k)$ above, inquire MAC value of $(M_j \parallel P, M_k \parallel P)$, denoted as $C_j^p = C_k^p$.
   −If each of them has at least a partial collision, i.e. $C_{jv}^p = C_{kv}^p$, $(1 \leq v \leq 7)$, the MAC algorithm is based on the hash function $\boldsymbol{H}$.
   −otherwise, the MAC algorithm is a random function or based on a random function.

**Complexity.** This attack requires about $2^{16.5}$ chosen messages, at most $2^{17.5}$ queries, a success rate of 0.63 by the birthday paradox.
**Remark:** Using our method in paper [14], we can easily obtain forgery attack on the MAC based the hash function $\boldsymbol{H}$ with the data complexity of $2^{16.5}$.

## 5 Recovering the Key $k$ of NMAC

It is remarked that the partial collision above can be regarded as an inner partial collision of HMAC, which can be used as a distinguisher to distinguish HMAC based on the hash function $\boldsymbol{H}$ from a random function. With this distinguisher, we can recover the key $k$ of HMAC.

### 5.1 Distinguishing Attacks on HMAC Based on the hash function $\boldsymbol{H}$

We first simply introduce an usual distinguishing attack method, which can distinguish HMAC based on a specific non-random hash function from a random function or HMAC based on a random function, then show our new distinguishing attack method.

**Distinguishing Attack 1**: Input a 256-bit message pair $(M_i, M_i')$, ask their MAC values $(C_i, C_i')$, which can be separated into eight equal length blocks, respectively.
−For a random function or HMAC based on a random function, the probability of having one partial collision pair $(C_{ij}, C_{ij}')(0 \leq j \leq 7)$ is $2^{-32}$.
−For HMAC based on a specific non-random hash function, such as the HMAC defined in Equ.(13). Let the $j^{th}$ 32-bit block of $k_1$ and $k_2$ be $k_{1,j}$ and $k_{2,j}$, respectively. If at least one pair $(C_{ij}, C_{ij}')(0 \leq j \leq 7)$ is a collision, the probability is $q = Pr_{k_{2,j}, M_{ij}} = (H_{k_{2,j}}(M_{ij}) \oplus H_{k_{2,j}}(M_{ij}' + \alpha') = 0)$, obviously $q > 2^{-32}$.

Thus we can distinguish the HMAC defined in Equ.(13) from the HMAC based on a random function or a random function.

**Distinguishing Attack 2**: Here, we will give a different distinguishing method by the birthday attack. We adopt segmenting techniques to detect an inner partial collision, which can identify the hash function $\boldsymbol{H}$, embedded in the HMAC. The specific steps are as follows:

1. Randomly selected about $2^{16.5}$ messages $M_i = M_{i0}||M_{i1}||M_{i2}||\ldots||M_{it-1}$ $(t \geq 9)$, calculate anther same length messages $M_i^{'} = M_i + \alpha(\alpha \neq 0))$. From Equ.(1-8), get their expansion message pair $(M_{ij}, M'_{ij})(1 \leq i \leq 2^{16.5}, 0 \leq j \leq 7)$.
2. For each message pair $(M_i, M'_i)$, inquire its HMAC value $(C_i, C'_i)$.
   −If at least one pair satisfies that $C_{ij} \bigoplus C'_{ij} = 0(0 \leq j \leq 7)$, for example, $C_{i7} \bigoplus C'_{i7} = 0$, the HMAC algorithm is based on a specific non-random hash function, and go to step 3.
   −or else, it is based on a random function.
3. In the case of chosen message attack, randomly select a nonzero message $M_b$, get message pair $(M_i \parallel M_b, M'_i \parallel M_b)$, ask their HMAC values:

$$C_i^a = H_{k_1}(H_{k_2}(M_i \parallel M_b)) = C_{i0}^a \parallel C_{i1}^a \parallel C_{i2}^a \parallel C_{i3}^a \parallel C_{i4}^a \parallel C_{i5}^a \parallel C_{i6}^a \parallel C_{i7}^a$$

$$C^b = H_{k_1}(H_{k_2}(M'_i \parallel M_b)) = C_{i0}^b \parallel C_{i1}^b \parallel C_{i2}^b \parallel C_{i3}^b \parallel C_{i4}^b \parallel C_{i5}^b \parallel C_{i6}^b \parallel C_{i7}^b.$$

   From picture 1, their $j^{th}$ components values are $C_{ij}^a = H_{k_{1,j}}(H_{k_{2,j}}(M_{ij} \parallel M_{bj}))$ and $C_{ij}^b = H_{k_{1,j}}(H_{k_{2,j}}(M'_{ij} \parallel M_{bj}))$.
4. Validated $C_{ij}^a \bigoplus C_{ij}^b$ value is 0 or not.
   −If all of the HMAC component pairs are meet $C_{ij}^a \bigoplus C_{ij}^b = 0$, the HMAC algorithm is based on the hash function $\boldsymbol{H}$, store the value $(M_i, M'_i, C_{ij})$ in Table A.
   −Otherwise, it is based on other hash function.

**Complexity**. The complexity is $2^{16.5}$ MAC queries in step 2. There is only 2 queries with $2^{16.5}$ entries in step 4. So the total complexity is dominated by step 2, which is about $2^{16.5}$ MAC queries for $2^{16.5}$ chosen messages.

**Success Rate.** The probability that there is a partial collision is 0.63 according to the birthday paradox, which is also the success rate of our attack.

**Remark:** Using our method in paper [14], we can easily obtain forgery attack on the HMAC based the hash function $\boldsymbol{H}$ with the data complexity of $2^{16.5}$.

### 5.2 Recovering the Key $k$ of NMAC Based on the hash function $H$

Let the secret key $k$ of NMAC be eight 32-bit blocks, i.e. $k = k^0||k^1||k^2||k^3||k^4||k^5||k^6||k^7$. By distinguishing attack 2 above, if the HMAC algorithm is based on the hash function $\boldsymbol{H}$, we can get a inner partial collision of $(C_{ij}, C_{ij})$ with $2^{16.5}$ data. This means that $H_{k_{1,7}}(H_{k_{2,7}}(M_{i7}))$=$H_{k_{1,7}}(H_{k_{2,7}}(M'_{i7}))$. From Fig.2 , we have $h_{2,7} = H_{k_{2,7}}(M_{i7})$=$H_{k_{2,7}}(M'_{i7})$, which is a inner partial collision. When the inner partial collision is identified, using force search attack, we can directly recover the $8^{th}$ key block $k^7$. The key $k^7$ is recovered in the following manner:

1. Choose a group values $(M_i, M'_i, C_{i7})$ from Table $A$.
2. Create Table $B$, which contains $2^{32}$ pairs $(k_{1,7}, k_{2,7})$, where $k_{1,7}$ value is from 0 to $2^{32}-1$, and $k_{2,7}$ can be deduced by Equ.(11-12). In turn, take a pair value $(k_{1,7}, k_{2,7})$, Apart encrypt $k_{1,7}$ by key $M_{i7}$ and $M'_{i7}$,
   − If $E_{M_{i7}}(k_{1,7}) = E_{M'_{i7}}(k_{1,7})$, then $h_{2,7} = E_{M_{i7}}(k_{1,7})$, goto step 3.
   − or else discard the value from Table B.
3. Decrypt $C_{i7}$ by $h^{-1}_{2,7}$. obtain 32-bit $k^d_{2,7}$=$E^{-1}_{h^{-1}_{2,7}}(C_{i7})$.
4. Verify $k^d_{2,7}$=$k_{2,7}$ is true or not
   −If it is true, Goto step 5.
   −Or else, discard the value from Table $B$.
5. Check remained value in Table $B$
   −If there is only one value $(k_{1,7}, k_{2,7})$ remained, it is correct.
   −Or else, goto step 1.

Two group $(M_i, M'_i, C_{i7})$ are enough to ascertain secret key $k^7$. In the same way, we can recover $k^0, k^1, k^2, k^3, k^4, k^5$ and $k^6$.

**Complexity.** Recover the secret key $k^7$ with at most $2^{16.5}$ data and $2^{16.5}$ encryptions. So the complexity of recovering the secret key $k$ is $2^{16.5}$ data and $2^{16.5}$ encryptions.

**Remark:** Using our method in paper [14], we can easily suggest a second preimage attack on the HMAC based the hash function $\boldsymbol{H}$ with the data complexity of $2^{16.5}$.

## 6 Conclusions

Utilizing segmenting techniques, we first construct a inner partial-collision distinguisher, which can distinguish a MAC/HMAC based on the hash function $\boldsymbol{H}$ from a random function. The distinguishing attack can be converted to forgery attacks easily. Furthermore, the distinguishing attack on HMAC can be applied to recover its secret key $k$. All attacks require $2^{16.5}$ data, which is the complexity of the partly birthday paradox.

## Acknowledgement

Thanks Professor Xiaoyun Wang who gave us many new attack ideas.

## References

1. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1-15. Springer, Heidelberg (1996)
2. Bellare, M., Kilian, J., Rogaway, P.: The Security of the Cipher Block Chaining Message Authentication Code. In: Desmedt, Y. G. (ed.) CRYPTO'94. LNCS 839, pp. 341-358. Springer, Heidelberg (1994)

3. Bellare, M.: New Proofs for NMAC and HMAC: Security without collisionresistance. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 602-619. Springer, Heidelberg (2006)
4. Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129C153. Springer, Heidelberg (2003)
5. Keting Jia, Xiaoyun Wang, Zheng Yuan, Guangwu Xu. Distinguishing and Second-Preimage Attack on CBC-like MACs. Accepted by Cryptology and Network Security-Cans 2009.
6. Kim, J., Biryukov, A., Preneel, B., Hong, S.: On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0, and SHA-1. In: De Prisco et al.(eds.) SCN 2006. LNCS, vol. 4116, pp. 242-256. Springer, Heidelberg (2006)
7. Kurosawa, K., Iwata, T.: TMAC: Two-Key CBC MAC. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp.265-273. Springer, Heidelberg (2003)
8. Yong Liu, Yu Chen, Zhong Chen, Lin Yang. A New Dedicated Compression Function Framework. In: Huanguo.Z et al.(eds.) ChinaCrypt'2008. pp. 214-219, 2008.
9. NIST, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B (2005)
10. Minematsu, K. Tsunoo, Y.: Provably Secure MACs from Differentially-Uniform Permutations and AES-Based Implementations, In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 226C241. Springer, Heidelberg (2006)
11. Preneel, B., van Oorschot, P.: MDx-MAC and Building Fast MACs from Hash Functions. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 1-14. Springer, Heidelberg (1995)
12. Xiaoyun Wang, Wei Wang, Keting Jia, Meiqin Wang. New Distinguishing Attack on MAC using Secret-Prefix Method. In: O. Dunkelman (Ed.): FSE 2009, LNCS 5665, pp. 363-374, 2009.
13. Xiaoyun Wang, Hongbo Yu, Wei Wang, Haiha Zhang, Tao Zhan. Cryptanalysis on HMAC/NMAC-MD5 and MD5-MAC. In: Joux, A. (ed.): EUROCRYPT 2009, LNCS 5479, Springer, Heidelberg, pp. 121-133, 2009.
14. Zheng Yuan, Wei Wang, Keting Jia, Guanfwu Xu, Xiaoyun Wang. New Birthday Attacks on Some MACs Based on Block Ciphers. In: Shai Halevi: Advances in Cryptology-Crypto 2009, LNCS 5677, Springer-Verlag, Berlin Heidelberg New York, pp. 209-230, 2009.