Improved Cryptanalysis of APOP-MD4 and NMAC-MD4 using New Differential Paths

Donghoon Chang[1], Jaechul Sung[2], Seokhie Hong[1], Sangjin Lee[1]

[1] Center for Information and Security Technologies
Korea University, Seoul, Korea
{dhchang, hsh, sangjin}@cist.korea.ac.kr
[2] Department of Mathematics, University of Seoul, Korea
jcsung@uos.ac.kr

Abstract. In case of security analysis of hash functions, finding a good collision-inducing differential paths has been only focused on. However, it is not clear how differential paths of a hash function influence the securities of schemes based on the hash function. In this paper, we show that any differential path of a hash function can influence the securities of schemes based on the hash function. We explain this fact with the MD4 hash function. We first show that APOP-MD4 with a nonce of fixed length can be analyzed efficiently with a new differential path. Then we improve the result of the key-recovery attack on NMAC-MD4 described by Fouque *et al.* [4] by combining new differential paths. Our results mean that good hash functions should have the following property : *It is computationally infeasible to find differential a path of hash functions with a high probability.*

Keywords : MD4, Differential Path, APOP, NMAC.

1 Introduction

The key recovery attack on block ciphers focuses mainly on finding a good property (of a block cipher) independent from a secret random key with a high probability and then recovers partial information of a round key with the property. On the other hand, security analyses of hash functions [9, 10, 12–17] focus on finding a collision-inducing differential path and then obtain a second-preimage or a collision with the path. When the securities of schemes based on hash functions such as HMAC and APOP protocol are analyzed, collision-inducing differential paths of the underlying hash function of the schemes are also used. However, we can ask the following questions ; *how can any differential path of a hash function influence the securities of schemes based on the hash function? How come the problem of finding any good differential path of hash functions doesn't be concerned?*

In fact, the method that finds collisions by connecting differential paths with high probabilities was utilized for collision-finding attacks of MD5 and SHA-1

[14, 16]. However, we can say that the method is also to find collision-inducing differential paths for multi blocks. To the best of our knowledge, there is no result on how any differential path of a hash function influences the securities of schemes based on the hash function. In this paper, we show that good differential paths can be used to reduce the attack complexity of schemes based on hash functions. For concrete examples, we propose new MD4 differential paths. Then we analyze APOP-MD4 and NMAC-MD4 with them. Our results show that good hash functions should have the property that it is difficult to find a differential path with a high probability. Table 1 and 2 summarizes our results.

Table 1. The Comparison of previous results with our result on a partial key-recovery attack on APOP-MD4

	the length of a nonce	attack ?	♯ of bits discovered	♯ of queries
Leurent [5]	arbitrary	Yes	56	57
Leurent [5]	fixed	No	·	·
This paper	fixed	Yes	56	2^{13}

Table 2. The Comparison of previous results with our result on a partial key-recovery attack on NMAC-MD4 : Given K_1, the second column indicates the number of bits of K_2 recovered with query-complexity in the third column.

	♯ of bits recovered	♯ of queries	success prob.	attack type
Fouque *et al.* [4]	1	2^{80}	1	standard
This paper	4	2^{23}	2^{-3}	related-key setting

2 Notations and Definitions

NMAC and HMAC. Fig. 1 and 2 show NMAC and HMAC based on a compression function f from $\{0,1\}^n \times \{0,1\}^b$ to $\{0,1\}^n$. K_1 and K_2 are n bits. $\overline{K} = K||0^{b-n}$ where K is n bits. opad is formed by repeating the byte '0x36' as many times as needed to get a b-bit block, and ipad is defined similarly using the byte '0x5c'. $H : \{IV\} \times (\{0,1\}^b)^* \rightarrow \{0,1\}^n$ is the iterated hash function. H is defined as follows : $H^f(IV, x_1||x_2|| \cdots ||x_t) = f(\cdots f(f(IV, x_1), x_2) \cdots, x_t)$ where x_i is b bits. Let g be a padding method. $g(x) = x||10^t||\mathsf{bin}_{64}(x)$ where t is smallest non-negative integer such that $g(x)$ is a multiple of b and $\mathsf{bin}_i(x)$ is the i-bit binary representation of x. Then, NMAC and HMAC are defined as follows. M is a any message of an arbitrary length. each M_i is b-bits. For example, in case of MD4, b is 512, n is 128 and f is the compression function of MD4.

$$\mathrm{NMAC}_{K_1,K_2}(M) = H(K_2, g(H(K_1, g(M))))$$
$$\mathrm{HMAC}_K(M) = H(IV, g(\overline{K} \oplus \mathsf{opad}||H(IV, g(\overline{K} \oplus \mathsf{ipad}||M)))).$$

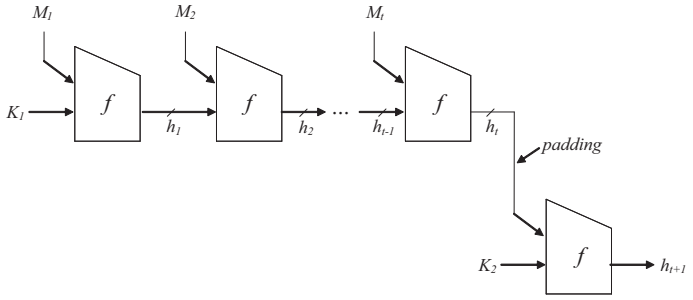

Fig. 1. NMAC ($g(M) = M_1||M_2||\cdots||M_t$)

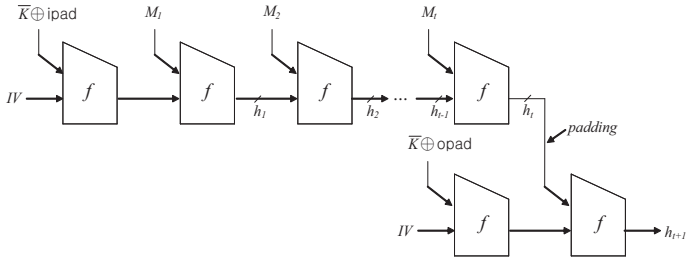

Fig. 2. HMAC ($g(\overline{K} \oplus \mathsf{ipad}||M) = \overline{K} \oplus \mathsf{ipad}||M_1||M_2||\cdots||M_t)$

MD4. The hash function MD4 uses a 128-bit fixed initial value IV. We write the MD4 hash value of a message M by MD4(M) or MD4(IV,M). The compression function f of $MD4$ is denoted by coMD4. Then MD4(IV,M) can be described by MD4(IV,M)=$H^{\mathrm{coMD4}}(IV, g(M))$.

Initial Value (IV) of MD4. The initial value of MD4 is as follows.

$$IV = (\mathsf{0x67452301}, \mathsf{0xefcdab89}, \mathsf{0x98badcfe}, \mathsf{0x10325476})$$

Boolean Functions of MD4. coMD4 consists of 48 steps (the first step is called step 1). The first 16 steps are called round 1, the second 16 steps are called round 2 and the last 16 steps are called round 3. The boolean function of each round is as follows.

$$f_1(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$$
$$f_2(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$$
$$f_3(x, y, z) = x \oplus y \oplus z$$

The Ordering of Message Words of MD4. The compression function coMD4 processes a 512-bit message block M per each compression function. A message block M is divided by 16 words as $M = m_1||m_2||\cdots||m_{16}$ where each m_i is

32-bit. The leftmost bit of a word is called 32-th bit of the word. The rightmost bit of a word is called 1-th bit of the word. The ordering of message words of coMD4 is as follows.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\psi(i)$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\psi(i+16)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16
$\psi(i+32)$	1	9	5	13	3	11	7	15	2	10	6	14	4	12	8	16

The Shift Rotations of MD4. The shift rotation S_i at step i is defined as follows.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S_i	3	7	11	19	3	7	11	19	3	7	11	19	3	7	11	19
S_{i+16}	3	5	9	13	3	5	9	13	3	5	9	13	3	5	9	13
S_{i+32}	3	9	11	15	3	9	11	15	3	9	11	15	3	9	11	15

Round Constants of MD4. Round constants are as follows.

$$K_1 = 0, K_2 = \mathsf{0x5a827999}, K_3 = \mathsf{0x6ed9eba1}$$

the Step Function of MD4. Let A_i be a updated value in step i. The initial value IV or the output of previous compression function is denoted by $(A_{i-3}, A_0, A_{-1}, A_{-2})$. A_i at round j is updated as follows.

$$A_i = (A_{i-4} + f_j(A_{i-1}, A_{i-2}, A_{i-3}) + m_{\psi(i)} + K_j)^{\lll S_i},$$

where S_i is the value of the left rotation in step i.

And the final output value of the compression function coMD4 is computed as follows.

$$\text{coMD4}(IV, M) = (A_{-3} + A_{45}, A_0 + A_{48}, A_{-1} + A_{47}, A_{-2} + A_{46}).$$

APOP [6]. APOP is a protocol for the authentication between a client and a mail server. The client and the mail server share a secret password. The mail server wants to identify the client with the shared password. Let H be a public hash function and P be a shared password. In order to authenticate the identity of the client, the mail server generates a random number *Nonce* and sends it to the client. If the client knows the password exactly, he can generate the hash value $H(Nonce||P)$ and return it to the mail server. Finally, the mail server can check if the value is correct, and then authenticate the identity of the client.

Notations in Table 4 $\sim$ 9 in the Appendix. Let A_i be the updated value in step i when a message M is processed. Let A'_i be the updated value in step i when a message M' is processed. $A_i = g^h$ means that all bits from g-th bit of A_i to $(g+h-1)$-th bit of A_i are the bit '1' and $(g+h)$-th bit of A_i is the bit '0'. And $A_i = g$ means that g-th bit of A_i is the bit '0'. Reversely, $A_i = -g^h$ means that all bits from g-th bit of A_i to $(g+h-1)$-th bit of A_i are the bit '0'

and $(g+h)$-th bit of A_i is the bit '1'. And $A_i = -g$ means that g-th bit of A_i is the bit '1'. $\Delta A_i = g^h$ means that A_i is g^h and A'_i is $-g^h$. Reversely, $\Delta A_i = -g^h$ means that A_i is $-g^h$ and A'_i is g^h. Also, the output of boolean function in each step and message words are defined in the same way. From the third column to the sixth column in Table 5, 7 and 9 in the appendix are conditions on A_i. The last column in Table 5, 7 and 9 say the number of conditions of A_i. $A_{i,j} = a$ means that the j-th bit of A_i is same as the j-th bit of A_{i-1}. $A_{i,j} = b$ means that the j-th bit of A_i is same as the j-th bit of A_{i-2}.

3 A New MD4 Differential Path I and Its Application to APOP-MD4

In 2007, Gaetan Leurent [5] proposed an attack method to recover a partial key of a secret password used in APOP-MD4 and APOP-MD5 authentication protocols. Sasaki *et al.* [11] also discovered independently a partial key-recovery attack on APOP-MD5. Their works are based on the collision-inducing differential paths discovered by Wang *et al.* [12, 14]. In their attack scenario, an attacker is a forged mail server who doesn't know a target-client's secret password but wants to know a partial information of the password by the communication with the client. their attack idea is based on the fact that the length of the nonce generated by a mail server has no limit. In other words, an attacker can choose a nonce of an arbitrary length. On the other hand, in case when the nonce has a fixed length, their attack doesn't work.

In this paper, *even when the nonce has a fixed length*, we show that we can recover efficiently a partial information of the password with *the MD4 differential path I* explained in the appendix. The differential paths used in previous partial-key recovery attack are all collision-inducing differential paths. However, any good differential path with a high probability is enough for analyzing APOP. This is because, even though the output difference of a path is not zero, we can check if two messages satisfy conditions in the path just by checking if the output difference satisfies the pattern of the output difference of the path. By this idea, we can recover efficiently a partial information of the secret password. We can write this problem as follows.Here, we consider only a case that the secret password P is 96-bit and the nonce is 416-bit.

Problem 1. Let IV be the fixed initial value of MD4. The secret password P is 96-bit and the nonce is 416-bit. An attacker can ask queries to a target-client, where each query $Nonce$ is a nonce in APOP. Then the client returns coMD4$(IV, Nonce||P)$ to the attacker. In this scenario, the attacker can recover 56 bits of the password P with 2^{13} queries and with the time complexity that corresponds to the time of sorting 2^{12} elements 2^{30} times.

Proof. Problem 1 can be solved by the idea discovered by Contini and Yin [2]. We write $P = P_1||P_2||P_3$ where each P_i is 32-bit. We want to recover 56 bits of P. More precisely, we will recover lower 30 bits of P_1 and lower 26 bits of P_2.

See Table 4 and 5. There are only six conditions from step 14 to the last step. MD4's IV also satisfies one condition corresponding to the initial value in Table 4 and 5. Since the attacker can choose nonces arbitrarily by himself, he choose two nonces which satisfy all conditions corresponding to from step 1 to step 13 in Table 4 and 5. We denote such two nonces by $N = (n_1||n_2||\cdots||n_{13})$ and $N' = (n'_1||n'_2||\cdots||n'_{13})$ such that $n'_1 - n_1 = -2^4$, $n'_4 - n_4 = 2$ and $n'_i = n_i$ for $i \neq 1, 4$. Next, he sends them to the client, and then obtains coMD4($N||P$) and coMD4($N'||P$). Since there are only six conditions from step 14 to the last step in Table 4 and 5, the following statements hold with the probability 2^{-6}.

$$\text{coMD4}(N'||P)\text{-coMD4}(N||P)=(2^4, ?, ?, ?)\cdots\cdots(1)$$

See Table 4 and 5. If we obtain (N, N') satisfying (1) with probability 2^{-6}, then such (N, N') also satisfies all conditions in Table 2 and 3 with the probability almost 1. This is because if (N, N') doesn't satisfy just one condition in Table 4 and 5, the probability that (N, N') satisfies (1) is 2^{-32}. Based on this observation, we can recover lower 30 bits of P_1 with the following algorithm 1.

Algorithm 1. Recovery of lower 30 bits of P_1.

1. The attacker chooses 2^{12} (N, N') pairs such that $n'_1 - n_1 = -2^4$, $n'_4 - n_4 = 2$ and $n'_i = n_i$ for $i \neq 1, 4$. Then he gives them to the client and obtains coMD4($N'||P$) and coMD4($N||P$) for all 2^{13} (N, N') pairs from the client.
2. He guesses lower 30 bits of P_1 used in step 14.
3. He selects all (N, N') pairs which make the fifth bit of A_{14} be '1', where A_{14} is the updated value in step 14.
4. If there is no (N, N') (which is selected in the phase 3) satisfying the statement (1), then outputs the guessed 30 bits of P_1 and stops this algorithm, otherwise go to the phase 2.

Now we explain how the algorithm 1 works. See Table 4 and 5 in the appendix. According to the Table 5, there is a condition that the fifth bit of A_{14} is '0'. Since the value of the shift rotation in step 14 is 7, lower 30 bits of P_1 influence the fifth bit of A_{14} in step 14. Therefore, if the attacker guesses the password wrongly, in the phase 3 of the algorithm 1, 2^{10} (N, N') pairs among 2^{11} (N, N') pairs selected on average in the phase 3 make the fifth bit of A_{14} be '0' on average when the corrected lower 30 bits of P_1 is applied to such 2^{10} (N, N'). So, in case of a wrong guess, each (N, N') among 2^{10} (N, N') pairs satisfies the statement (1) with the probability 2^{-5}. We expect that $2^5 (= 2^{-5} \cdot 2^{10})$ (N, N') pairs satisfy the statement (1).

Now, let's compute the number of wrong passwords such that there is no (N, N') pair satisfies the statement (1). The probability that there is no (N, N') (which is selected in the phase 3 satisfying the statement (1) in case of a wrong-guessed password is $(1 - 2^{-5})^{2^{10}}$. Since there are $2^{30} - 1$ wrong passwords, the number of wrong passwords that there is no (N, N') (which is selected in the phase 3 satisfying the statements (1) is $(2^{30} - 1) \cdot (1 - 2^{-5})^{2^{10}} < 2^{-19}$. On the

other hand, in case of the correct-guessed password, since we select correctly (N, N') pairs which make the fifth bit of A_{14} be '1' in the phase 3, no (N, N') (which is selected in the phase 3 follows the differential path I in Table 4 and 5. So, the number of (N, N') (which is selected in the phase 3) satisfying the statements (1) becomes $2^{-21}(= 2^{-32} \cdot 2^{11})$. Therefore, we can recover correctly lower 30 bits of P_1 by the phase 4 of the algorithm 1.

Next, we can recover lower 26 bits of P_2 with the algorithm 2.

Algorithm 2. Recovery of lower 26 bits of P_2.

1. The attacker guesses higher two bits of P_1.
2. He guesses lower 26 bits of P_1 used in step 15.
3. Among 2^{12} (N, N') pairs obtained in the algorithm 1, he selects all (N, N') pairs which make the fifth bit of A_{15} be '0', where A_{15} is the updated value in step 15.
4. If there is no (N, N') (which is selected in the phase 3) satisfying the statement (1), then outputs the guessed 30 bits of P_1 and stops this algorithm, otherwise go to next phase.
5. For a fixed higher two bits of P_1, if he has not yet searched all 2^{26} candidates of lower 26 bits of P_1, go to the phase 2, otherwise go to the phase 1.

According to the Table 5, there is a condition that the fifth bit of A_{15} is '0'. Since the value of the shift rotation in step 15 is 11, lower 26 bits of P_2 influence the fifth bit of A_{15} in step 15. Similarly, we can recover correctly lower 26 bits of P_2 with the algorithm 2. ■

4 New MD4 Differential Paths II-A,B and Their Applications to NMAC-MD4

Fouque, Leurent and Nguyen [4] constructed 22 collision-inducing differential paths. Their differential paths have eighty conditions per each path. Among eigthty conditions, one condition is for the initial value. In case of NMAC and HMAC, on the assumption that the values of K_1 and coMD4$(IV, \bar{K} \oplus ipad)$ are known, they could find one bit of K_2 or coMD4$(IV, \bar{K} \oplus opad)$ with the time complexity 2^{80} because the initial value has one condition in the path. The assumption is reasonable because Contini *et al.* [2] introduced a method to recover K_1 and coMD4$(IV, \bar{K} \oplus ipad)$ with complexity 2^{63}. So, if we find two messages which produce the same MAC value with their path, we know that one bit of K_2 or coMD4$(IV, \bar{K} \oplus opad)$ satisfies the condition in their path, otherwise one bit of K_2 or coMD4$(IV, \bar{K} \oplus opad)$ doesn't satisfy the condition in their path, that is, satisfies the opposite condition of a given condition. Since 2^{80} is so big, their attack seems to be impossible to implement their attack in the current computing power.

In this paper, we show that the complexity that requires to recover four bits can be improved very efficiently if we use our new differential paths II-A,B in the appendix and the attacker model is in the related-key setting and K_1 is known. More precisely, we can recover four bits of K_2 of NMAC with the time complexity 2^{23} and the probability 2^{-3} in *the related-key setting*. Since 2^{23} is much less than 2^{80}, our attack scenario can be implemented efficiently. Our attack result shows that the analyzing method based on differential paths is more powerful than that based on collision-inducing differential paths. In the following subsection, we will explain our attack scenario in detail.

4.1 Attack Scenario on NMAC-MD4

The following two figures in Fig. 3 are the expressions of NMAC-MD4$_{K_1,K_2}(M) = h_2$, where the bit length of M is less than 448 and g is the padding method explained in section 2. For $|M| < 448$, two expressions are identical.

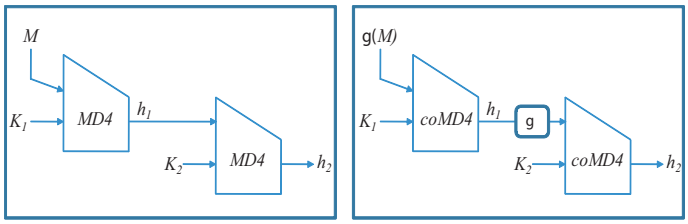

Fig. 3. Identical Two Expressions : NMAC-MD4$_{K_1,K_2}(M) = h_2$, $|M| < 448$

The goal of this subsection is to describe an efficient partial-key recovery attack on K_2 in the related-key setting, on the assumption that K_1 is known and satisfies two conditions corresponding to the initial value in Table 6 and 7. For this goal, only messages of bit length 447 are used as NMAC-MD4 queries. This is because we want to simulate NMAC-MD4 with using coMD4 two times and control bits as many as possible, where at least lower 65 bits of $g(M)$ is determined by the length of M. In case of $|M| = 447$, the lower 65 bits of $g(M)$ is fixed as $1||bin_{64}(447)$. See Fig. 4. We make queries M and M', where $\alpha_1 = g(M') - g(M)$, $\beta_1 = h'_1 - h_1$, $\alpha_2 = g(h'_1) - g(h_1)$, $\beta_2 = h'_2 - h_2$, $\Delta K_1 = K'_1 - K_1$ and $\Delta K_2 = K'_2 - K_2$.

For fixed values $\alpha_1(\neq 0)$, $\beta_1(\neq 0)$, β_2, ΔK_1 and ΔK_2, we denote the probability that M' and M satisfy the relation $\alpha_1 \rightarrow_{\Delta K_1} \beta_1$ by p and the relation $\alpha_2 \rightarrow_{\Delta K_2} \beta_2$ by q, on the assumption that $\Delta K_1 = K'_1 - K_1$ and $\Delta K_2 = K'_2 - K_2$ in the related-key setting. Here, α_2 is determined by β_1 from the relations $\beta_1 = h'_1 - h_1$ and $\alpha_2 = g(h'_1) - g(h_1)$. Then, the equation $\beta_2 = h'_2 - h_2$ holds with the probability $p \cdot q$ in Fig. 4. For example, according to Table 6, α_1, β_1 and ΔK_1 are defined as follows. $*$ can be $+$ or $-$.

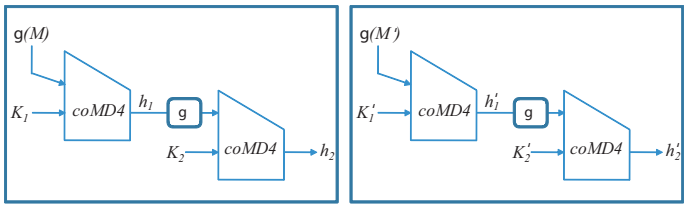

Fig. 4. Processing M and M' : $|M|, |M'| < 448$

- $\alpha_1 = g(M') - g(M)$
  : m'_1-m_1=$*2^{31}$, $m'_i = m_i$ for $i \neq 1$, where $g(M') = m'_1||\cdots||m'_{16}$ and $g(M) = m_1||\cdots||m_{16}$.
- $\beta_1 = h'_1 - h_1$
  : $h'_1 - h_1$=coMD4$(K'_1, g(M'))$−coMD4$(K_1, g(M))$=$(*2^{31}, 0, 0, -2^{28})$.
- $\Delta K_1 = K'_1 - K_1$
  : $K'_1 - K_1 = (*2^{31}, 0, 0, -2^{28})$ and K_1 satisfies two condition in Table 7.

And according to Table 8, α_2, β_2 and ΔK_2 are also defined as follows. $*$ can be $+$ or $-$. Here, α_2 is determined by β_1 from the relations $\beta_1 = h'_1 - h_1$ and $\alpha_2 = g(h'_1) - g(h_1)$.

- $\alpha_2 = g(h'_1) - g(h_1)$
  : l'_1-l_1=$*2^{31}$, l'_4-l_4=-2^{28} and $l'_i = l_i$ for $i \neq 1, 4$, where $g(h'_1) = l'_1||\cdots||l'_{16}$ and $g(h_1) = l_1||\cdots||l_{16}$.
- $\beta_2 = h'_2 - h_2$
  : $h'_2 - h_2$=coMD4$(IV', g(h'_1))$−coMD4$(IV, g(h_1))$=$((-1)^t \cdot 2^{19}, ?, ?, *2^8)$.
- $\Delta K_2 = K'_2 - K_2$
  : $K'_2 - K_2 = (*2^{31} + (-1)^t \cdot 2^{19}, 2^{28}, 0, 0)$. And the 29th bit of the second word of K_2 is '0' and the 20th bit of the first word of K_2 is 't'.

Since we analyze NMAC-MD4 when K_1 is known and satisfies two conditions in Table 6 and 7, we can get easily M and M' satisfying the relation $\alpha_1 \rightarrow_{\Delta K_1} \beta_1$ with the probability 1 by the advanced modification method explained in [12, 10], where $\Delta K_1 = K'_1 - K_1$ in the related-key setting. Then, if K_2 satisfies three conditions in Table 8 and 9, then a (h'_1, h_1) pair satisfies the relation $\alpha_2 \rightarrow_{\Delta K_2} \beta_2$ with the probability 2^{-22} by twenty-two conditions corresponding to the updated values from step 1 to step 48 in Table 8 and 9. In other words, the following equation (2) holds with the probability 2^{-22}.

$$h'_2 - h_2 = ((-1)^t \cdot 2^{19}, ?, ?, *2^8) \cdots\cdots (2)$$

If K_2 doesn't satisfy three conditions corresponding to the initial value in Table 8 and 9, the equation (2) holds with the probability 2^{-64}. Therefore, when K_1 is given and K_1 satisfies two conditions in Table 6 and 7, if there exists a (M,M') pair (among 2^{22} (M', M) pairs) satisfying the above relation $\alpha_1 \rightarrow_{\Delta K_1} \beta_1$) such

that NMAC-MD4(M')−NMAC-MD4(M)=$((-1)^t \cdot 2^{19}, ?, ?, *2^8)$, we can know that K_2 satisfies three conditions in Table 8 and 9, that is, we can recover three bits of K_2 with the probability 2^{-3}. From the equation (2), we can also get the value of t which is one bit of K_2. Therefore, we can recover four bits of K_2 in total.

Remark 1. Fouque, Leurent and Nguyen [4] had to simulate coMD4 $2^{64} \cdot \frac{1}{22}$ times (which is based on the birthday attack complexity) in order to get (M, M') such that $g(\text{coMD4}(K_1, M')) - g(\text{coMD4}(K_1, M))$ is same as the message difference according to a path among their twenty-two paths. On the other hands, as we said above, in the related-key setting we can find (M, M') very easily with the path in Table 4 and 5 such that $g(\text{coMD4}(K_1, M')) - g(\text{coMD4}(K_1, M))$ is same as the message difference according to the path in Table 8 and 9. This means that the attack method based on differential paths is more powerful than that based on collision-inducing differential paths.

5 Conclusion

In this paper, we described how differential paths of a hash function influence the security of schemes based on the hash function. Good differential paths can be used to reduce the security of schemes based hash functions. It seems that finding good differential paths with a high probability is easier than finding a collision-inducing differential path with a high probability. Our results show that good hash function should have the property that *it is difficult to find any good differential path with a high probability*. Full key-recovery attacks on APOP-MD4 and NMAC-MD4 with using ideas explained in this paper are future works.

References

1. M. Bellare, R. Canetti and H. Krawczyk, *Keying Hash Functions for Message Authentication*, Advances in Cryptology-Crypto'96, LNCS 1109, pp. 1–15, Springer-Verlag, 1996.
2. S. Contini and Y. L. Yin, *Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions*, Asiacrypt'06, LNCS 4284, pp. 37–53, Springer-Verlag, 2006.
3. FIPS 180-1, Secure Hash Standard, US Department of Commerce, Washington D. C, 1996.
4. P. A. Fouque, G. Leurent, and P. Q. Nguyen, *Full Key-Recovery Attacks on HMAC/NMAC-MD4 and NMAC-MD5*, Advances in Cryptology-Crypto'07, LNCS 4622, pp. 13–30, Springer-Verlag, 2007.
5. G. Leurent, *Message Freedom in MD4 and MD5 Collisions: Application to APOP*, FSE 2007, LNCS 4593, pp. 309–328, Springer-Verlag, 2007.
6. J. Myers and M. Rose, *Post Office Protocol - Version 3*, RFC 1939 (Standard) (May 1996) Updated by RFCc 1957, 2449.
7. Ronald L. Rivest, *The MD4 message-digest algorithm*, Request for comments (RFC 1320), Internet Activities Board, Internet Privacy Task Force, 1992.

8. Ronald L. Rivest, *The MD5 message-digest algorithm*, Request for comments (RFC 1321), Internet Activities Board, Internet Privacy Task Force, 1992.
9. Y. Sasaki, Y. Naito, N. Kunihiro and K. Ohta, *Improved Collision Attacks on MD4 and MD5*, IEICE TRANS. FUNDAMENTALS, VOL. E90-A, NO. 1, pp. 36–47, Jan. , 2007.
10. Y. Sasaki, L. Wang, K. Ohta and N. Kunihiro, *New Message Difference for MD4*, FSE 2007, LNCS 4593, pp. 329–348, Springer-Verlag, 2007.
11. Y. Sasaki, Go. Yamamoto and K. Aoki, *Practical Password Recovery on an MD5 Challenge and Response*, Cryptology ePrint Archive, Report 2007/101, 2007.
12. X. Wang, X. Lai, D. Feng, H. Chen and X. Yu, *Cryptanalysis of the Hash Functions MD4 and RIPEMD*, Advances in Cryptology-Eurocrypt'05, LNCS 3494, pp. 1–18, Springer-Verlag, 2005.
13. X. Wang, A. C. Yao and F. Yao, *Cryptanalysis on SHA-1*, CRYPTOGRAPHIC HASH WORKSHOP, October 31-November 1, 2005.
14. X. Wang and H. Yu, *How to Break MD5 and Other Hash Functions*, Advances in Cryptology-Eurocrypt'05, LNCS 3494, pp. 19–35, Springer-Verlag, 2005.
15. X. Wang, H. Yu and Y. L. Yin, *Efficient Collision Search Attacks on SHA-0*, Advances in Cryptology-Crypto'05, LNCS 3621, pp. 1–16, Springer-Verlag, 2005.
16. X. Wang, Y. L. Yin and H. Yu, *Finding Collisions in the Full SHA-1*, Advances in Cryptology-Crypto'05, LNCS 3621, pp. 17–36, Springer-Verlag, 2005.
17. H. Yu, G. Wang, G. Zhang and X. Wang, *The Second-Preimage Attack on MD4*, SCN 2005, LNCS 3810, pp. 1–12, Springer-Verlag, 2005.

Appendix

5.1 New MD4 Differential Path I

In this subsection, we propose a new MD4 differential path I which has a specific output difference. Our proposed differential path has four conditions corresponding from step 16 to step 48. This is smaller than the number of conditions of any other known paths. Table 3 shows the differences between our path and previous known paths. Table 4 and 5 show our new MD4 differential path I and

Table 3. The Comparison of the Numbers of Conditions of MD4

	Step 14 $\sim$ 48	Step 15 $\sim$ 48	Step 16 $\sim$ 48	Step 17 $\sim$ 48
Wang *et al.* [12]	43	37	31	25
Yu *et al.* [17]	44	42	40	38
Sasaki *et al.* [10]	29	22	16	12
Table 4 and 5 in Our paper	6	5	4	4

conditions corresponding to the initial value and the updated values. Here, M and M' are 512-bit. The notations in Table 4 and 5 were already described in

section 2. 2^i in the following statements 1) and 2) means i multiples of two.

1) The relation between M and M' : $m'_1 - m_1 = -2^4$, $m'_4 - m_4 = 2$ and $m'_i = m_i$ for $i \neq 1, 4$.
2) The output difference : coMD4(IV, M')−coMD4(IV, M)=$(2^4, ?, ?, ?)$.

5.2 New MD4 Differential Path II-A

Table 6 and 7 show our new MD4 differential path II-A and conditions corresponding to the initial value and the updated values. Here, M and M' are 512-bit. The notations in Table 6 and 7 were already described in section 2. 2^i in the following statements 1), 2) and 3) means i multiples of two.

1) The relation between M and M' : $m'_1 - m_1 = *2^{31}$, $m'_i = m_i$ for $i \neq 1$.
2) The difference of initial values : $IV' - IV = (*2^{31}, 0, 0, -2^{28})$, and the 29th bit of the forth word of IV is '1'.
3) The output difference : coMD4(IV, M')−coMD4(IV, M)=$(*2^{31}, 0, 0, -2^{28})$.

5.3 New MD4 Differential Path II-B

Table 8 and 9 show our new MD4 differential path II-B and conditions corresponding to the initial value and the updated values. Here, M and M' are 512-bit. The notations in Table 8 and 9 were already described in section 2. 2^i in the following statements 1), 2) and 3) means i multiples of two. $\pm$ means 'all are +' or 'all are −'. In Table 9, all t's are 1 or 0.

1) The relation between M and M' : $m'_1 - m_1 = *2^{31}$, $m'_4 - m_4 = -2^{28}$, and $m'_i = m_i$ for $i \neq 1, 4$.
2) The difference of initial values : $IV' - IV = (*2^{31} + (-1)^t \cdot 2^{19}, 2^{28}, 0, 0)$, and the 29th bit of the second word of IV is '0' and the 20th bit of the first word of IV is 't'.
3) The output difference : coMD4(IV, M')−coMD4(IV, M)=$((-1)^t \cdot 2^{19}, ?, ?, *2^8)$.

Table 4. New MD4 Differential Path I : i is any value.

Step	Output Diff. of Bool. Func.	Diff. of Message Word	Diff. of Updated Value
			ΔA_0
1		-5	$\Delta A_1 = -8$
2			ΔA_2
3			ΔA_3
4		2	$\Delta A_4 = 21^9$
5	28		$\Delta A_5 = 31^2, -11^1$
6			ΔA_6
7	-29,-32		$\Delta A_7 = -8, 11$
8			$\Delta A_8 = 8$
9	11		$\Delta A_9 = 2^{10}$
10			ΔA_{10}
11	8,-11		ΔA_{11}
12	-8		ΔA_{12}
13			$\Delta A_{13} = 5$
14			ΔA_{14}
15			ΔA_{15}
16			ΔA_{16}
17		-5	ΔA_{17}
18			ΔA_{18}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
27			ΔA_{27}
28			ΔA_{28}
29		2	$\Delta A_{29} = 5$
30			ΔA_{30}
31			ΔA_{31}
32			ΔA_{32}
33		-5	ΔA_{33}
34			ΔA_{34}
35			ΔA_{35}
36			ΔA_{36}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
41			ΔA_{41}
42			ΔA_{42}
43			ΔA_{43}
44			ΔA_{44}
45		2	ΔA_{45}
46	?		$\Delta A_{46} =?$
47	?		$\Delta A_{47} =?$
48	?		$\Delta A_{48} =?$
			$\Delta(A_{-3} + A_{45}) = 5^i$
			$\Delta(A_0 + A_{48}) =?$
			$\Delta(A_{-1} + A_{47}) =?$
			$\Delta(A_{-2} + A_{46}) =?$

Table 5. Conditions of New MD4 Differential Path I : i is any value.

Step	Updated Value	$32 \sim 25$	$24 \sim 17$	$16 \sim 9$	$8 \sim 1$	♯ of Conditions
	A_0				a	1
1	$A_1 = -8$				1	1
2	A_2	1			0	2
3	A_3	a a 0 a a a	a a a a		1	11
4	$A_4 = 21^9$	a a 0 1 1 1 1 1	1 1 1 1	a a	a	15
5	$A_5 = 31^2, -11^1$	1 1 0 0 0 0 0 0	0 0 0 0	1 0	0 0	16
6	A_6	1 0 1 0 1 1 1 1	1 1 1 1	0 0	0 0	16
7	$A_7 = -8, 11$	1 1		1 0	1 1	6
8	$A_8 = 8$			a 1 a a	0 a a a a a a	11
9	$A_9 = 2^{10}$			0 1 1 1	1 1 1 1 1 1 1	11
10	A_{10}			0 1 0 0	0 0 0 0 0 0 0	11
11	A_{11}			1 1 1 1	0 1 1 1 1 1 1	11
12	A_{12}				a	1
13	$A_{13} = 5$				0	1
14	A_{14}				0	1
15	A_{15}				1	1
16	A_{16}					
17	A_{17}					
18	A_{18}					
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	A_{27}					
28	A_{28}				a	1
29	$A_{29} = 5$				0	1
30	A_{30}				b	1
31	A_{31}				a	1
32	A_{32}					
33	A_{33}					
34	A_{34}					
35	A_{35}					
36	A_{36}					
⋮	⋮	⋮	⋮	⋮	⋮	⋮
41	A_{41}					
42	A_{42}					
43	A_{43}					
44	A_{44}					
45	A_{45}					
46	$A_{46} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
47	$A_{47} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
48	$A_{48} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
	$A_{-3} + A_{45} = 5^i$					
	$A_0 + A_{48} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
	$A_{-1} + A_{47} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
	$A_{-2} + A_{46} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	

Table 6. New MD4 Differential Path II-A

Step	Output Diff. of Bool. Func.	Diff. of Message Word	Diff. of Updated Value
			$\Delta A_{-3} = *32$
			$\Delta A_{-2} = -29$
			ΔA_{-1}
			ΔA_0
1		*32	ΔA_1
2			$\Delta A_2 = -4^{[1]}$
3			ΔA_3
4			ΔA_4
5			ΔA_5
6			$\Delta A_6 = -11^{[16]}$
7			ΔA_7
8	24		$\Delta A_8 = 11$
9	14,26		$\Delta A_9 = 29, 17^{[3]}$
10	11,-17,-18,-19,20		$\Delta A_{10} = 24$
11	-19		$\Delta A_{11} = -30$
12			$\Delta A_{12} = 30$
13			$\Delta A_{13} = 32, 20^{[11]}$
14	-24		ΔA_{14}
15	30		ΔA_{15}
16	-30		ΔA_{16}
17		*32	$\Delta A_{17} = 23$
18			ΔA_{18}
19			ΔA_{19}
20			ΔA_{20}
21			$\Delta A_{21} = 26$
22			ΔA_{22}
23			ΔA_{23}
24			ΔA_{24}
25			$\Delta A_{25} = 29$
26			ΔA_{26}
27			ΔA_{27}
28			ΔA_{28}
29			$\Delta A_{29} = 32$
30			ΔA_{30}
31			ΔA_{31}
32			ΔA_{32}
33		*32	ΔA_{33}
⋮	⋮	⋮	⋮
48			ΔA_{48}
			$\Delta(A_{-3} + A_{45}) = *32$
			$\Delta(A_0 + A_{48})$
			$\Delta(A_{-1} + A_{47})$
			$\Delta(A_{-2} + A_{46}) = -29$ *or* $-29^{[1]}$ *or* $-29^{[2]}$ *or* $-29^{[3]}$ *or* $(29, 30, 31, 32)$

Table 7. Conditions of New MD4 Differential Path II-A

Step	Updated Value	$32 \sim 25$	$24 \sim 17$	$16 \sim 9$	$8 \sim 1$	♯ of Conditions
	$A_{-3} = *32$					
	$A_{-2} = -29$	1				1
	A_{-1}					
	A_0	1				1
1	A_1				a a	2
2	$A_2 = -4^{1}$				1 0	2
3	A_3				0 0	2
4	A_4				1 1	2
5	A_5	a a a	a a a a a a a a	a a a a a a		17
6	$A_6 = -11^{16}$	1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0		17
7	A_7	0 0 0	1 0 0 0 0 0 0 0	0 0 0 0 0 0		17
8	$A_8 = 11$	a 1 0 1	1 1 1 1 1 1 1 1	1 1 0 1 1 0		18
9	$A_9 = 29, 17^{3}$	0	a 0 1 1 1	1		7
10	$A_{10} = 24$	0 0	0 0 1 0 0	1		8
11	$A_{11} = -30$	1 1	0 1 1 1 1			7
12	$A_{12} = 30$	a a 0 a a a a a	1 a a a a			13
13	$A_{13} = 32, 20^{11}$	0 0 1 1 1 1 1 1	1 1 1 1 1			13
14	A_{14}	0 0 0 0 0 0 0 0	0 0 0 0 0			13
15	A_{15}	1 1 0 1 1 1 1 1	1 1 1 1 1			13
16	A_{16}		a			1
17	$A_{17} = 23$		0			1
18	A_{18}		b			1
19	A_{19}		a			1
20	A_{20}	a				1
21	$A_{21} = 26$	0				1
22	A_{22}	b				1
23	A_{23}	a				1
24	A_{24}	a				1
25	$A_{25} = 29$	0				1
26	A_{26}	b				1
27	A_{27}	a				1
28	A_{28}	a				1
29	$A_{29} = 32$	0				1
30	A_{30}	b				1
31	A_{31}	a				1
32	A_{32}					
33	A_{33}					
⋮	⋮	⋮	⋮	⋮	⋮	⋮
48	A_{48}					
	$A_{-3} + A_{45} = *32$					
	$A_0 + A_{48}$					
	$A_{-1} + A_{47}$					
	$A_{-2} + A_{46} =$ -29 or -29^{1} or -29^{2} or -29^{3} or (29,30,31,32)					

Table 8. New MD4 Differential Path II-B : i and j are any value.

Step	Output Diff. of Bool. Func.	Diff. of Message Word	Diff. of Updated Value
			$\Delta A_{-3} = *32, (-1)^t \cdot 20$
			ΔA_{-2}
			ΔA_{-1}
			$\Delta A_0 = 29$
1		*32	$\Delta A_1 = (-1)^t \cdot 23$
2			ΔA_2
3			ΔA_3
4		-29	ΔA_4
5			$\Delta A_5 = (-1)^t \cdot 26$
6			ΔA_6
7			ΔA_7
8			ΔA_8
9			$\Delta A_9 = (-1)^t \cdot 29$
10			ΔA_{10}
11			ΔA_{11}
12			ΔA_{12}
13			$\Delta A_{13} = (-1)^t \cdot 32$
14			ΔA_{14}
15			ΔA_{15}
16			ΔA_{16}
17		*32	ΔA_{17}
18			ΔA_{18}
⋮	⋮	⋮	⋮
27			ΔA_{27}
28			ΔA_{28}
29		-29	$\Delta A_{29} = -32$
30			ΔA_{30}
31			ΔA_{31}
32			ΔA_{32}
33		*32	ΔA_{33}
34			ΔA_{34}
35			ΔA_{35}
⋮	⋮	⋮	⋮
43			ΔA_{43}
44			ΔA_{44}
45		-29	$\Delta A_{45} = -32$
46	*32		$\Delta A_{46} = *9$
47	?		$\Delta A_{47} =?$
48	?		$\Delta A_{48} =?$
			$\Delta(A_{-3} + A_{45}) = (-1)^t \cdot 20^i$
			$\Delta(A_0 + A_{48}) =?$
			$\Delta(A_{-1} + A_{47}) =?$
			$\Delta(A_{-2} + A_{46}) = *9^j$

Table 9. Conditions of New MD4 Differential Path II-B : all t's are same. i is any value.

Step	Updated Value	$32 \sim 25$	$24 \sim 17$	$16 \sim 9$	$8 \sim 1$	♯ of Con -ditions
	$A_{-3} = *32, (-1)^t \cdot 20$		t			
	A_{-2}					
	A_{-1}	a				1
	$A_0 = 29$	0	a			2
1	$A_1 = (-1)^t \cdot 23$	0	t			2
2	A_2	1	0			2
3	A_3		1			1
4	A_4	a				1
5	$A_5 = (-1)^t \cdot 26$	t				1
6	A_6	0				1
7	A_7	1				1
8	A_8	a				1
9	$A_9 = (-1)^t \cdot 29$	t				1
10	A_{10}	0				1
11	A_{11}	1				1
12	A_{12}	a				1
13	$A_{13} = (-1)^t \cdot 32$	t				1
14	A_{14}	0				1
15	A_{15}	1				1
16	A_{16}					
17	A_{17}					
18	A_{18}					
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	A_{27}					
28	A_{28}	a				1
29	$A_{29} = -32$	1				1
30	A_{30}	b				1
31	A_{31}	a				1
32	A_{32}					
33	A_{33}					
34	A_{34}					
35	A_{35}					
⋮	⋮	⋮	⋮	⋮	⋮	⋮
43	A_{43}					
44	A_{44}					
45	$A_{45} = -32$	1				1
46	$A_{46} = *9$					
47	$A_{47} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
48	$A_{48} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
	$A_{-3} + A_{45} =$ $= (-1)^t \cdot 20^i$					
	$A_0 + A_{48} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
	$A_{-1} + A_{47} =?$	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	
	$A_{-2} + A_{46} = *9^j$					