ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2010/04(03)-0202-12
DOI:10.3778/j.issn.1673-9418.2010.03.002

E-mail:fcst@public2.bta.net.cn
http://www.ceaj.org
Tel:+86-10-51616056

Meaningful Collision Attack on MD4*

JIA Keting[1+], WANG Xiaoyun[1,2]

1. Key Lab of Cryptologic Technology and Information Security, MOE, Shandong University, Jinan 250100, China

2. Institute for Advanced Study, Tsinghua University, Beijing 100084, China

+ Corresponding author: E-mail: ktjia@mail.sdu.edu.cn

有意义的 MD4 碰撞攻击*

贾珂婷[1+],王小云[1,2]

1. 山东大学 密码技术与信息安全教育部重点实验室,济南 250100

2. 清华大学 高等研究院,北京 100084

摘 要:2005 年的欧密会,Wang 等提出了一种构造 MD4 碰撞的有效方法,该方法不仅对寻找随机碰撞有效,还可以用于构造有意义的碰撞。以 Wang 的技术为基础,进一步分析和探讨了对构造纯文本文件有意义的 MD4 碰撞,给出了一种构造纯文本文件的有意义 MD4 碰撞的方法,概率为 $2^{-33.77}$。在 1996 年的 FSE 会议上,Dobbertin 的"Cryptanalysis of MD4"给出了一个有意义的碰撞,而其在开头包含了 16 个随机字符。这里给出了一个基于 Latin-1 字符集的有意义的碰撞。

关键词:有意义的碰撞;MD4 算法;美国信息互换标准码

文献标识码:A **中图分类号**:TP309

JIA Keting, WANG Xiaoyun. Meaningful collision attack on MD4. Journal of Frontiers of Computer Science and Technology, 2010, 4(3): 202-213.

Abstract: In Eurocrypt'05, Wang et al. presented new techniques to find collisions of the hash function MD4. The techniques are not only efficient to find random collisions, but also applicable to find meaningful collisions on MD4. This paper reports a further research on the meaningful collisions of plain text of MD4, and provides how to construct meaningful collisions of ASCII text on MD4 according to Wang's techniques with probability $2^{-33.77}$ in gen-

* The National Natural Science Foundation of China under Grant No.60525201 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2007CB807902 (国家重点基础研究发展规划(973)).

Received 2009-05, Accepted 2009-07.

eral. In FSE'96, Dobbertin gave a meaningful collision in his paper on cryptanalysis of MD4, which contains meaningless words at the beginning of the text. Furthermore, a complete meaningful collisions on MD4 based on Latin-1 character set is shown.

Key words: meaningful collision; MD4; ASCII

1 Introduction

Hash function is defined as a function that compresses an input string of arbitrary length to an output string of fixed length. Cryptographic hash functions are fundamental primitives used in many cryptographic schemes and protocols, such as electronic signature, authentication of message, electronic commerce and bit commitment, etc. A number of dedicated hash functions have been proposed, but most of them have been shown to be cryptographically weak. The hash function MD4[1] is one of them, but it is still used in some applications where speed is more important.

The cryptanalysis of hash function has achieved tremendous progress in the construction of collisions in the past years. Dobbertin[2] gave a collision attack for the full MD4 in 1996, which could also be used to find meaningful collisions. Wang et al.'s breakthrough work[3-6] presented new techniques to find efficiently collisions on the main hash functions from the MD4 family, e.g. MD4[3], RIPEMD[3], MD5[4], SHA-0[5] and SHA-1[6], moreover the techniques can be applicable to explore the second-preimage of MD4[4]. Yu et al.[7] gave another differential path with fewer conditions, which could be used to find the second-preimage of MD4 more efficiently. Recently, some papers explain how concrete collisions can be used to obtain meaningful collisions of files with special formats: Certificates[8], postscript documents[9], PDF, TIFF and MS Word 97 documents[10], etc. They produce meaningful collisions by inserting some random collisions in the controlling command of the documents (e.g. "if-then-else" in postscript documents). The documents share the same hash value but display completely different contents.

All these collision attacks can lead to a grave threat in practice. In addition it's very significant to construct meaningful collisions in terms of the encoding of the character. It's much more complex to find the meaningful collisions than random collisions, as there are many restrictions on the message value. Dobbertin[2] gave a meaningful collision example with 4 meaningless words at the beginning of the collision. In this paper, a further research on the meaningful collisions of MD4 is given. An approach is proposed to find a sentence with the first 14 words which satisfy the conditions in the first round of MD4, then some conditions by the way of modifying the last two words are corrected to get a meaningful collision with probability $2^{-33.77}$.

This paper is organized as follows: Section 2 describes the MD4 hash function in details. Some basic terminologies and notations are given in Section 3. Section 4 presents our method to find meaningful collisions of ASCII text on MD4. In Section 5, a concrete collision example is given to demonstrate that this attack is of practical relevance. The paper is summarized in Section 6.

2 Description of MD4

The message digest algorithm MD4 takes an arbitrary length message M^* and produces a 128 bits hash value $H(M^*)$. First, the input message is padded to be a multiple of 512 bits. Since padding does not affect collision attack, its explanation is omitted. The padded message $\overline{M^*}$ is then divided into 512 bits message blocks, i.e. $\overline{M^*}=(M_0,M_1,\cdots,M_{n-1})$, and processed by Merkle-Damgård iterative structure. Each iteration in-

vokes a compression function which takes a 128 bits chaining value and 512 bits message blocks and outputs another 128 bits chaining value. The initial chaining value (called IV) is a set of fixed constants H_0 = (0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476). The output of the compression function H_{j+1} is computed using M_j and H_j, j=0,1,…,n-1. Let H_n be the hash value of M.

2.1 The Compression Function of MD4

The compression function has three rounds (R_1, R_2, and R_3) and every round has 16 steps and employs a round function. The three round functions are defined as follows:

$$F(X,Y,Z)=(X\wedge Y)\vee(\neg X\wedge Z)$$

$$G(X,Y,Z)=(X\wedge Y)\vee(Y\wedge Z)\vee(X\wedge Z)$$

$$H(X,Y,Z)=X\oplus Y\oplus Z$$

The inputs to the compression function are H_j and M_j. $M_j=(m_0,m_1,\cdots,m_{15})$, where each m_i is a 32 bits word. All computations in the compression function are word-oriented. The compression function performs the following 48 steps (a_0,b_0,c_0,d_0 are initialized by H_j):

R_1, For i=0, 1, 2, 3

$$a_{i+1}=(a_i+F(b_i,c_i,d_i)+m_{4i})\lll 3$$

$$d_{i+1}=(d_i+F(a_i,b_i,c_i)+m_{4i+1})\lll 7$$

$$c_{i+1}=(c_i+F(d_i,a_i,b_i)+m_{4i+2})\lll 11$$

$$b_{i+1}=(b_i+F(c_i,d_i,a_i)+m_{4i+3})\lll 19$$

R_2, For i=0, 1, 2, 3

$$a_{i+5}=(a_{i+4}+G(b_{i+4},c_{i+4},d_{i+4})+m_i+0x5a827999)\lll 3$$

$$d_{i+5}=(d_{i+4}+G(a_{i+4},b_{i+4},c_{i+4})+m_{i+4}+0x5a827999)\lll 5$$

$$c_{i+5}=(c_{i+4}+G(d_{i+4},a_{i+4},b_{i+4})+m_{i+8}+0x5a827999)\lll 9$$

$$b_{i+5}=(b_{i+4}+G(c_{i+4},d_{i+4},a_{i+4})+m_{i+12}+0x5a827999)\lll 13$$

R_3, For i=0, 1, 2, 3

$$a_{i+9}=(a_{i+8}+H(b_{i+8},c_{i+8},d_{i+8})+m_i+0x6e9eba1)\lll 3$$

$$d_{i+9}=(d_{i+8}+H(a_{i+8},b_{i+8},c_{i+8})+m_{i+8}+0x6e9eba1)\lll 9$$

$$c_{i+9}=(c_{i+8}+H(d_{i+8},a_{i+8},b_{i+8})+m_{i+4}+0x6e9eba1)\lll 11$$

$$b_{i+9}=(b_{i+8}+H(c_{i+8},d_{i+8},a_{i+8})+m_{i+12}+0x6e9eba1)\lll 15$$

Here, $\lll s$ denotes left rotation by s bits.

After the 48 steps, H_{j+1} is computed as follows: $H_{j+1}=(aa_0\parallel bb_0\parallel cc_0\parallel dd_0)$, where $aa_0=a_{12}+a_0$, $bb_0=b_{12}+b_0$, $cc_0=c_{12}+c_0$, $dd_0=d_{12}+d_0$.

3 Preliminaries

3.1 The Transform between Word and Byte

A word is 32 bits and a byte is 8 bits. A sequence of bits can be interpreted as a sequence of bytes in a natural manner, where each consecutive group of 8 bits is interpreted as a byte with the most significant bit of each byte listed firstly. Similarly, a sequence of bytes can be interpreted as a sequence of 32 bits words, where each consecutive group of 4 bytes is interpreted as a word with little endian, i.e. the least significant byte given firstly. For example, the sequence of bytes "ABCD, 1234" can be transformed into the sequence of words "0x44434241, 0x34333231".

3.2 Character Set

A character encoding maps each character in a character set to a numeric value that can be represented by a computer. These numbers can be represented by a single byte or multiple bytes. For example, the ASCII encoding uses seven bits to represent the Latin alphabet, punctuation, and control characters seen Table 16. ISO 8859 character set series encoding is a 8 bits character set in the basis of ASCII, created based on many national or regional standards to support different European languages. For example, ISO 8859-1 (Latin-1) character set (Table 17) is supposed by Western Europe. In this paper, the character set represented by a single byte is in the use. In ASCII character set, the character is unreadable if its 8th bit is 1.

3.3 Basic Notations

(1) $M=(m_0,m_1,\cdots,m_{15})$ and $M'=(m'_0,m'_1,\cdots,m'_{15})$ represent two 512 bits messages. $\Delta M=M'-M=(\Delta m_0,\Delta m_1,\cdots,\Delta m_{15})$ denotes difference between M and M', where $\Delta m_i=m'_i-m_i$, i=1,2,…,15.

(2) $a_{i,j}$, $b_{i,j}$, $c_{i,j}$, $d_{i,j}$ and $m_{i,j}$ represent respectively the j-th bit of a_i, b_i, c_i, d_i, m_i, where the least significant bit is the 1st bit, and the most significant bit is the 32nd bit.

(3) $x_i[j]$, $x_i[-j]$ (x can be a,b,c,d) are the resulting values by only changing the j-th bit of the word x_i. $x_i[-j]$ is obtained by changing the j-th bit of x_i from 0 to 1. $x_i[-j]$ is obtained by changing the j-th bit of x_i from 1 to 0.

(4) The symbols $\wedge$, $\vee$, $\oplus$, $<<<$, $>>>$, + and $\boxplus$ respectively denote logical AND, logical OR, logical exclusive-or (XOR), left rotation, right rotation, addition modulo 2^{32} and addition modulo 32 in this paper.

3.4 The Basic Message Modification

The basic message modification technique is a kind of simple message modification which uses the changes of corresponding single bit message to ensure all the conditions in the first round hold. For example, in order to correct the condition $a_{2,4}=1$ to $a_{2,4}=0$, just need to change

$$m_4 \leftarrow m_4 \oplus 0x1$$

from

$$a_2=(a_1+f(b_1,c_1,d_1)+m_4)<<<3$$

4 How to Find Meaningful Collisions on MD4

In this Section, a method is described to find meaningful collisions of ASCII text on MD4 in details.

4.1 Choosing Appropriate Differential Paths

There are many differential paths to find MD4 collisions[7,11-12] after Wang et al.'s attack[4]. Some of them are mainly constructed to find collisions as fast as possible, e.g. citations[5-6], which have excessive conditions in R_1, but fewer conditions in R_2 and R_3. Others are used to find second-preimage of MD4, which have fewer conditions in all, but have relatively more conditions in R_2 and R_3, e.g. the citations[7].

Generally speaking, the fewer conditions of differential paths in R_2 and R_3, the easier to find collisions by message modification. However, it isn't absolutely made for us to find meaningful collisions of ASCII text. Firstly, choosing the right difference is very important to construct meaningful collisions. If the difference contains $\pm 2^7$, $\pm 2^{15}$, $\pm 2^{23}$, $\pm 2^{31}$, one of M and M' is unreadable. Secondly, it's vital for us to make a comprehensive view of all the conditions in the differential paths. The sparser of the conditions, the better, for the fewer conditions imply the higher probability of a message pair becoming a collision. By this principle stated above, check all the differential paths in citations[4,7,11-12], only the path presented by Yu et al.[7] is right, although it has more conditions than others in R_2 and R_3. Convenient to describe, the differential paths of the citation[7] are given in Table 1.

$$M=(m_0,m_1,\cdots,m_{15})$$

$$\Delta M=(0,0,0,0,e2^i,0,\cdots,0)$$

$$M'=M+\Delta M$$

Where $e=\pm 1$ and $0\leqslant i\leqslant 31$.

If M satisfies all the 62 conditions given in Table 2, (M,M') is a collision. The last column of Table 2 is the bits of the message used to correct conditions of the previous column by basic message modification. In order to get meaningful sentences, need to consider the restrictions on the message value. One of M and M' is unreadable when $i=7,15,23,31$, and there are more conditions when $i=17,26$. So, it's unnecessary to regard the differential paths, when $i=\pm k$, $k=7,15,17,23,26,31$.

In the first round, when some conditions are corresponding to the 8th, 16th, 24th, or 32nd of the m_i, it's infeasible to correct the wrong conditions according the basic message modification, because if $m_{i,k}=1, k=8,16,24,32$, m_i has an unreadable byte. Table 3 provides the bits of the last column of Table 2 is corresponding to the bit position in a byte. It's obvious that the number of the changes of sufficient conditions imme-

Table 1 The collision differential paths of MD4

表 1 MD4 的碰撞差分路线

Step	Output for M	m_i	s_i	Δm_i	Output difference	Output for M'	Sufficient conditions
1	a_1	m_0	3				
2	d_1	m_1	7				
3	c_1	m_2	11				
4	b_1	m_3	19				
5	a_2	m_4	3	2^i	$2^{i\boxplus 3}$	$a_2[i\boxplus 4]$	$a_{2,i\boxplus 4}=0$
6	d_2	m_5	7			d_2	$b_{1,i\boxplus 4}=c_{1,i\boxplus 1}$
7	c_2	m_6	11			c_2	$d_{2,i\boxplus 4}=0$
8	b_2	m_7	19			b_2	$c_{2,i\boxplus 4}=1$
9	a_3	m_8	3		$2^{i\boxplus 6}$	$a_3[-(i\boxplus 7),i\boxplus 8]$	$a_{3,i\boxplus 7}=1,a_{3,i\boxplus 8}=0$
10	d_3	m_9	7			d_3	$b_{2,i\boxplus 7}=c_{2,i\boxplus 7},b_{2,i\boxplus 8}=c_{2,i\boxplus 8}$
11	c_3	m_{10}	11		$-2^{i\boxplus 17}$	$c_3[-(i\boxplus 18)]$	$c_{3,i\boxplus 18}=1,d_{3,i\boxplus 7}=1,d_{3,i\boxplus 8}=0$
12	b_3	m_{11}	19			b_3	$c_{3,i\boxplus 7}=1,c_{3,i\boxplus 8}=1,d_{3,i\boxplus 8}=a_{3,i\boxplus 18}$
13	a_4	m_{12}	3		$2^{i\boxplus 9}$	$a_4[i\boxplus 10]$	$a_{4,i\boxplus 10}=0,b_{3,i\boxplus 18}=0$
14	d_4	m_{13}	7				$b_{3,i\boxplus 10}=c_{3,i\boxplus 10},a_{4,i\boxplus 18}=0$
15	c_4	m_{14}	11		$-2^{i\boxplus 28}$	$c_4[-(i\boxplus 29)]$	$c_{4,i\boxplus 29}=1,d_{4,i\boxplus 10}=0$
16	b_4	m_{15}	19				$c_{4,i\boxplus 10}=1,d_{4,i\boxplus 29}=a_{4,i\boxplus 29}$
17	a_5	m_0	3		$2^{i\boxplus 12}$	$a_5[i\boxplus 13]$	$a_{5,i\boxplus 13}=0,b_{4,i\boxplus 29}=d_{4,i\boxplus 29}$
18	d_5	m_4	5	2^i	$2^{i\boxplus 5}+2^{i\boxplus 17}$	$d_5[i\boxplus 6,i\boxplus 18]$	$d_{5,i\boxplus 6}=0,d_{5,i\boxplus 18}=0,a_{5,i\boxplus 29}=b_{4,i\boxplus 29},b_{4,i\boxplus 13}=c_{4,i\boxplus 13}+1$
19	c_5	m_8	9		$-2^{i\boxplus 5}$	$c_5[-(i\boxplus 6)]$	$c_{5,i\boxplus 6}=1,a_{5,i\boxplus 18}=b_{4,i\boxplus 18},d_{5,i\boxplus 13}=b_{4,i\boxplus 13},a_{5,i\boxplus 6}=b_{4,i\boxplus 6}$
20	b_5	m_{12}	13			b_5	$c_{5,i\boxplus 13}=d_{5,i\boxplus 13},c_{5,i\boxplus 18}=a_{5,i\boxplus 18}$
21	a_6	m_1	3		$2^{i\boxplus 10}$	$a_6[i\boxplus 16]$	$a_{6,i\boxplus 16}=0,b_{5,i\boxplus 18}=c_{5,i\boxplus 18}$
22	d_6	m_5	5		$2^{i\boxplus 22}$	$d_6[i\boxplus 23]$	$d_{6,i\boxplus 23}=0,b_{5,i\boxplus 16}=c_{5,i\boxplus 16},a_{6,i\boxplus 6}=b_{5,i\boxplus 6}+1$
23	c_6	m_9	9		$-2^{i\boxplus 14}$	$c_6[i\boxplus 15,-(i\boxplus 16)]$	$c_{6,i\boxplus 15}=0,c_{6,i\boxplus 16}=1,d_{6,i\boxplus 16}=b_{5,i\boxplus 16},a_{6,i\boxplus 23}=b_{5,i\boxplus 23}$
24	b_6	m_{13}	13			b_6	$d_{6,i\boxplus 15}=a_{6,i\boxplus 15},c_{6,i\boxplus 23}=a_{6,i\boxplus 23}$
25	a_7	m_2	3			a_7	$b_{6,i\boxplus 15}=d_{6,i\boxplus 15},b_{6,i\boxplus 23}=c_{6,i\boxplus 23},b_{6,i\boxplus 16}=d_{6,i\boxplus 16}+1$
26	d_7	m_6	5		$2^{i\boxplus 27}$	$d_7[i\boxplus 28]$	$a_{7,i\boxplus 15}=b_{6,i\boxplus 15},d_{7,i\boxplus 28}=0,a_{7,i\boxplus 16}=b_{6,i\boxplus 16}$
27	c_7	m_{10}	9		$-2^{i\boxplus 23}$	$c_7[-(i\boxplus 24)]$	$c_{7,i\boxplus 24}=1,a_{7,i\boxplus 28}=b_{6,i\boxplus 28}$
28	b_7	m_{14}	13			b_7	$d_{7,i\boxplus 24}=a_{7,i\boxplus 24},c_{7,i\boxplus 28}=a_{7,i\boxplus 28}$
29	a_8	m_3	3			a_8	$b_{7,i\boxplus 24}=d_{7,i\boxplus 24},b_{7,i\boxplus 28}=c_{7,i\boxplus 28}$
30	d_8	m_7	5		2^i	$d_8[i\boxplus 1]$	$d_{8,i\boxplus 1}=0,a_{8,i\boxplus 24}=b_{7,i\boxplus 24}$
31	c_8	m_{11}	9		-2^i	$c_8[-(i\boxplus 1)]$	$c_{8,i\boxplus 1}=1,a_{8,i\boxplus 1}=b_{7,i\boxplus 1}$
32	b_8	m_{15}	13			b_8	
33	a_9	m_0	3			a_9	
34	d_9	m_8	9			d_9	$a_{9,i\boxplus 1}=b_{8,i\boxplus 1}$
35	c_9	m_4	11	2^i		c_9	
36	b_9	m_{12}	15			b_9	

Table 2 The set of sufficient conditions for the MD4 differential paths

表 2 MD4 差分路线的充分条件

Step	Output for M	m_i	s_i	Sufficient conditions	Corresponding message bits
1	a_1	m_0	3		
2	d_1	m_1	7		
3	c_1	m_2	11		
4	b_1	m_3	19	$b_{1,i}=c_{1,i}$	$m_{3,i}$
5	a_2	m_4	3	$a_{2,i}=0$	$m_{4,i}$
6	d_2	m_5	7	$d_{2,i}=0$	$m_{5,i}$
7	c_2	m_6	11	$c_{2,i}=1$	$m_{6,i}$
8	b_2	m_7	19	$b_{2,i}=c_{2,i}$, $b_{2,i}=c_{2,i}$	$m_{7,i}$, $m_{7,i}$
9	a_3	m_8	3	$a_{3,i}=1$, $a_{3,i}=0$	$m_{8,i}$, $m_{8,i}$
10	d_3	m_9	7	$d_{3,i}=1$, $d_{3,i}=0$, $d_{3,i}=a_{3,i}$	$m_{9,i}$, $m_{9,i}$, $m_{9,i}$
11	c_3	m_{10}	11	$c_{3,i}=1$, $c_{3,i}=1$, $c_{3,i}=1$	$m_{10,i}$, $m_{10,i}$, $m_{10,i}$
12	b_3	m_{11}	19	$b_{3,i}=0$, $b_{3,i}=c_{3,i}$	$m_{11,i}$, $m_{11,i}$
13	a_4	m_{12}	3	$a_{4,i}=0$, $a_{4,i}=1$	$m_{12,i}$, $m_{12,i}$
14	d_4	m_{13}	7	$d_{4,i}=0$, $d_{4,i}=a_{4,i}$	$m_{13,i}$, $m_{13,i}$
15	c_4	m_{14}	11	$c_{4,i}=1$, $c_{4,i}=1$	$m_{14,i}$, $m_{14,i}$
16	b_4	m_{15}	19	$b_{4,i}=c_{4,i}+1$, $b_{4,i}=d_{4,i}$	$m_{15,i}$, $m_{15,i}$
17	a_5	m_0	3	$a_{5,i}=b_{4,i}$, $a_{5,i}=0$, $a_{5,i}=b_{4,i}$, $a_{5,i}=b_{4,i}$	
18	d_5	m_4	5	$d_{5,i}=0$, $d_{5,i}=b_{4,i}$, $d_{5,i}=0$	
19	c_5	m_8	9	$c_{5,i}=1$, $c_{5,i}=d_{5,i}$, $c_{5,i}=a_{5,i}$	
20	b_5	m_{12}	13	$b_{5,i}=c_{5,i}$, $b_{5,i}=c_{5,i}$	
21	a_6	m_1	3	$a_{6,i}=0$, $a_{6,i}=b_{5,i}$, $a_{6,i}=b_{5,i}+1$	
22	d_6	m_5	5	$d_{6,i}=b_{5,i}$, $d_{6,i}=a_{6,i}$	
23	c_6	m_9	9	$c_{6,i}=0$, $c_{6,i}=1$, $c_{6,i}=a_{6,i}$	
24	b_6	m_{13}	13	$b_{6,i}=d_{6,i}$, $b_{6,i}=c_{6,i}$, $b_{6,i}=d_{6,i}+1$	
25	a_7	m_2	3	$a_{7,i}=b_{6,i}$, $a_{7,i}=b_{6,i}$, $a_{7,i}=b_{6,i}$	
26	d_7	m_6	5	$d_{7,i}=a_{7,i}$, $d_{7,i}=0$	
27	c_7	m_{10}	9	$c_{7,i}=1$, $c_{7,i}=a_{7,i}$	
28	b_7	m_{14}	13	$b_{7,i}=d_{7,i}$, $b_{7,i}=c_{7,i}$	
29	a_8	m_3	3	$a_{8,i}=b_{7,i}$, $a_{8,i}=b_{7,i}$	
30	d_8	m_7	5	$d_{8,i}=0$	
31	c_8	m_{11}	9	$c_{8,i}=1$	
32	b_8	m_{15}	13		
33	a_9	m_0	3	$a_{9,i}=b_{8,i}$	
34	d_9	m_8	9		
35	c_9	m_4	11		
36	b_9	m_{12}	15		

diately impacted by the 8th bit of a byte is 1, when $\Delta m_4 = 2^{0+8k}$ or 2^{2+8k}, (k=0,1,2,3) from Table 3. In order to make the basic message modification much easier, it selects Δm_4=1 as an example in the remainder of this paper.

Table 3 The number of sufficient conditions immediately impacted by the every bit of a byte in R_1

表 3 消息字节比特对应的第一轮充分条件数

Δm_4	The bit of a byte							
	1st	2nd	3rd	4th	5th	6th	7th	8th
2^{0+8k} (k=0,1,2,3)	4	3	2	3	5	1	5	1
2^{1+8k} (k=0,1,2,3)	1	4	3	2	3	5	1	5
2^{2+8k} (k=0,1,2,3)	5	1	4	3	2	3	5	1
2^{3+8k} (k=0,1,2,3)	1	5	1	4	3	2	3	5
2^{4+8k} (k=0,1,2,3)	5	1	5	1	4	3	2	3
2^{5+8k} (k=0,1,2,3)	3	5	1	5	1	4	3	2
2^{6+8k} (k=0,1,2,3)	2	3	5	1	5	1	4	3
2^{7+8k} (k=0,1,2,3)	3	2	3	5	1	5	1	4

4.2 Choosing Appropriate Message

Now describe how to choose the meaningful messages which satisfy the sufficient conditions of Table 2.

4.2.1 Choosing the First 6 Words

To find a meaningful message block M, so that the new message block M' is meaningful by adding 1 on the Δm_4 of M. Therefore, the 17th character may be the number 0~8, or letter, etc. It can construct message as follows:

M="You would pay 543$ for"

M'="You would pay 544$ for";

M="Geogre bought a boat,"

M'="Geogre bought a coat,";etc.

4.2.2 Choosing Remainder Words

The sentence may become meaningless after the basic message modification in R_1 in two ways: Firstly, the word including the changed bit has an unreadable byte, for the condition is corresponding to the 8th, 16th, 24th, or 32nd bit of the message; secondly, the word containing the changed bit is nonsemantic or inconsecutive in the sentence.

After basic message modification, it's necessary to check the message as follows:

Case 1 Check if the word is readable. If not, firstly, use the bit-carry technique to correct the wrong conditions. For example, it will correct the condition $d_{3,7}$=0 to $d_{3,7}$=1, with probability 2^{-7} as follows:

```
for (i=1;i<8;i++)
{
  if (d3,7-i ≠ m9,32-i)
  {
    if (d3,7-i=1)
    {
      m9,32-i = m9,32-i +1;
      return success;
    }
    else
    {
      m9,32-i = m9,32-i -1;
      return success;
    }
  }
}
return failure.
```

Generally, the probability of success correcting the ith bit by the bit-carry technique is $1-1/2^{i-1}$. If success, goto Case 2, otherwise, the former words should be instead of, then correct the conditions by basic message modification and goto Case 1.

Case 2 Check the consistency of the sentence. If not, need to adjust the sentence. In order to keep the modified sentence consecutive, change one or two characters adjacent to it or replace it with another character with supposed bit.

For example, M="You would pay 543$ for the boat.", after basic message modification, M="You would pay 543$ for the boa|.…". The character "t" becomes "|" after modification, which implies the 4th bit of the

31st character should be 1, so the word "radio" can replace "boat", for the 4th bit of "i" is 1. So we adjust M="You would pay 543$ for the radio.…".

Besides, it is very useful that the changes of the first register and the changes of F function in the modification. It's very flexible.

For example, it can correct the condition $d_{3,18} \neq a_{3,18}$ to $d_{3,18}=a_{3,18}$ by changing $d_{2,11}$, if m_5 has freedom degree to change. It's obvious from Table 4 that the change on $d_{2,11}$ doesn't work on the conditions on c_2, b_2 and a_3, if $c_{2,11}=0$ or $a_{3,30}=0$ or $d_{3,5}=0$ or $d_{3,6}=0$. The success probability is 15/16.

Table 4 The possible chaining variable bit by the change of $d_{2,11}$

表 4 $d_{2,11}$ 改变,链接变量可能会变化的比特

Chaining variables R	Possible variable bit	conditional bit
c_2	22	4
b_2	9,30	7,8
a_3	1,14,25	7,8
d_3	5,18,21,29	7,8,18

Then choose next word, implement basic message modification, and goto Case 1.

Practically, the methods in Case 1 and Case 2 can all be applied with each other. The dividing line is not always marked and clear. It can correct all the sufficient conditions in R_1 by message modification method mentioned above, then get an example after message modification in the first round as follows:

M="You would pay 543$ for the radio. We are sure your cosT is out of"

M'="You would pay 544$ for the radio. We are sure your cosT is out of"

4.2.3 Sentence-Making

After message modification in R_1, 38 conditions in the last two rounds have to be corrected. It mainly makes use of the last two words to correct them, so the last two words need enongh freedom degree. Thereby password, signature, batch number, code number, or ID etc. may be at the end of the sentence.

It can make use of the result in Table 5 to choose messages and make sentences. "*" denotes any readable byte; "?" denotes any readable byte with restrictions in R_1. One of many sentence structures is given in Table 6. " " denotes blank; "#" denotes number [0-9]; m_0 represents the debit; m_2 and m_3 represent the borrower; m_{10} and m_{11} represent the witness; m_{12} and m_{13} are used to keep the consecutive of the sentence.

It has reserved enough freedom degree in m_0, m_2, m_3, m_4 to make conditions of the first 10 chaining variables satisfied, because the debit, borrower, and witness can represent many names.

Table 5 The bytes immediately impact on the sufficient conditions in R_1

表 5 R_1 中直接对应充分条件的字节

m_0	m_1	m_2,	m_3	m_4	m_5	m_6	m_7
****	****	****	**?*	?***	***?	***?	**?*
m_8	m_9	m_{10},	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
?***	??*?	?**?	**??	??**	?*?*	?*?*	*??*

Table 6 A structure of a meaningful collision

表 6 一个有意义碰撞的结构

m_0	m_1	m_2,	m_3	m_4	m_5	m_6	m_7
****	+len	nt+*	****	####	+dol	lars	.+Wi
m_8	m_9	m_{10},	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
tnes	s+is	+***	****	****	****	****	****

Give an example with all the sufficient conditions met in R_1 as follows:

M="Bill lent Frank 4585 dollars. Witness is Mike. Num LkCWxxcdg9ghjk"

M'="Bill lent Frank 5585 dollars. Witness is Mike. Num LkCWxxcdg9ghjk"

4.2.4 Message Modification in R_2

In order to make M and M' become a collision on MD4, the remainder 38 conditions have to be corrected.

It can correct part of conditions in the second round by advanced message modification, which includes various technique details. In order to keep the sentence consecutive, correct the wrong conditions only by modifying the message words m_{14} and m_{15}.

(1) The correction of $a_{5,13}$

If $a_{5,13} \neq b_{4,6}$, correct it as follows:

```
for (i=1; i<8; i++)
{
  if (b_{4,(35-i) mod 32} ≠ m_{15,16-i})
  {
    if (m_{15,16-i}=0)
    {
      m_{15,16-i}=m_{15,16-i}+1;
      return success;
    }
    else
    {
      m_{15,16-i}=m_{15,16-i}-1;
      return success;
    }
  }
}
return failure.
```

The probability of success correction is 127/128.

(2) Correcting other conditions

Indeed, there are many kinds of methods to correct a condition and the advanced message modification is very flexible. A kind of modification method is given to correct the conditions about $a_{5,6}$, $a_{5,18}$, $a_{5,29}$, $d_{5,6}$, $d_{5,13}$, $c_{5,13}$, $c_{5,18}$, more details seen in Table 7, Table 10, Table 11, Table 12, Table 13 and Table 14.

Table 7 The modification for correcting conditions on $a_{5,13}$

表 7 $a_{5,13}$ 的条件修改

15	m_{14}	11	c_4, d_4, a_4, b_3	
16	m_{15}	19	$b'_4=b_4 \oplus 2^9, c_4, d_4, a_4$	$m_{15} \leftarrow b'_4>>>19-b_3-F(c_4,d_4,a_4)$
17	m_0	3	a'_5, b'_4, c_4, d_4	$c_{4,10} \neq d_{4,10}$

The probability of $m_{15,24}=1$ is 1/2 in the modification

After $2^{33.77}$ MD4 compressing computations, if use last two words to correct the conditions in R_2 and R_3, it will get a meaningful collision with selected 14 words.

5 Collisions for Cheat

Jonai and Bill both lived in Western Europe. Jonai had some trouble in cash flow turnover. Bill wanted to lend some money to Jonai. They agreed on 6 024 dollars, and they invited Mize as a witness. Bill asked Jonai to sign a contract using a digital signature scheme which was base on some public-key algorithm and the hash function MD4. The contract encoding by Latin-1 (ISO 8859-1) character set is as follows:

Receipt for a Loan

Bill lent Jonai 6 024 dollars. Witness is Mize.

Num bKMZ#êM) ì]1Q

The characters after "Num" stand for the code number of the contact declared by Bill in advance. When Jonai signed the contract, later, Bill substituted the

Table 8 The collisions M and M' with little-endian in hex denote without padding

表 8 16 进制小端的形式表示无填充的消息 M 和 M' 的碰撞

	Bill lent Frank 4585 dollars. Witness is Mike.Num LkCWxx+7d$r¬ *_
M	6c6c6942 6e656c20 72462074 206b6e61 35383534 6c6f6420 7372616c 6957202e
	73656e74 73692073 6b694d20 754e2e65 6b4c206d 78785743 2464372b 5f2aac72
	Bill lent Frank 5585 dollars. Witness is Mike.Num LkCWxx+7d$r¬ *_
M'	6c6c6942 6e656c20 72462074 206b6e61 35383535 6c6f6420 7372616c 6957202e
	73656e74 73692073 6b694d20 754e2e65 6b4c206d 78785743 2464372b 5f2aac72
H	418ee357 88180328 1f19aa3b 944ea7e4

Table 9 The two-block collisions M' and M with little-endian in hex denote without padding

表 9 16 进制小端的形式表示无填充的两个分组消息 M 和 M' 的碰撞

M	65636552	20747069	20726f66	6f4c2061	200a6e61	20202020	20202020	20202020
	20202020	20202020	20202020	20202020	20202020	20202020	20202020	0a202020
	6c6c6942	6e656c20	6f4a2074	2069616e	34323036	6c6f6420	7372616c	6957202e
	73656e74	73692073	7a694d20	4e0a2e65	62206d75	235a4d4b	29f74dea	51315dec
M'	65636552	20747069	20726f66	6f4c2061	200a6e61	20202020	20202020	20202020
	20202020	20202020	20202020	20202020	20202020	20202020	20202020	0a202020
	6c6c6942	6e656c20	6f4a2074	2069616e	34323037	6c6f6420	7372616c	6957202e
	73656e74	73692073	7a694d20	4e0a2e65	62206d75	235a4d4b	29f74dea	51315dec
H	34718703	8876d717	c5cc3639	fb1ceb5f				

contract file by another which is as follows:

Receipt for a Loan

Bill lent Jonai 7 024 dollars. Witness is Mize.

Num bKMZ#êM) ì]1Q

Replacing 6 024 by 7 024, the MD4 hash value of the contract prepared by Bill in advance does not change. The first 1 024 bits of M and M' is shown in Table 9.

6 Conclusion

This paper shows how to find some meaningful collisions on MD4 in consideration of the encoding, based on the method presented by Wang et al. in Eurocrypt'05. It obtains a meaningful collision with probability $2^{-33.77}$ on the average, and the probability can be improved in accordance with the concrete example. Not only the result is useful to construct meaningful collisions of the character set encoding by one byte, it can be extended to construct meaningful collisions on other character set: Ansi, UTF-8, etc.

Reference:

[1] Rivest R. RFC 1320 The MD4 message-digest algorithm[S]. MIT and RSA Data Security, 1992-04.

[2] Dobbertin H. Cryptanalysis of MD4[C]//LNCS 1039: Proc of the FSE 1996. [S.l.]: Springer-Verlag, 1996: 53-69.

[3] Wang X Y, Lai X J, Feng D G, et al. Cryptanalysis of the hash functions MD4 and RIPEMD[C]//LNCS 3494: Proc of the Eurocrypt 2005. [S.l.]: Springer-Verlag, 2005: 1-18.

[4] Wang X Y, Yu H B. How to break MD5 and other hash functions[C]//LNCS 3494: Proc of the Eurocrypt 2005. [S.l.]: Springer-Verlag, 2005: 19-35.

[5] Wang X Y, Yu H B, Yin Y L. Efficient collision search attacks on SHA-0[C]//LNCS 3621: Proc of the CRYPT 2005. [S.l.]: Springer-Verlag, 2005: 1-16.

[6] Wang X Y, Yin Y L, Yu H B. Finding collisions in the full SHA-1[C]//LNCS 3621: Proc of the CRYPT 2005. [S.l.]: Springer-Verlag, 2005: 17-36.

[7] Yu H B, Wang G L, Zhang G Y, et al. The second-preimage attack on MD4[C]//LNCS 3810: Proc of the CANS 2005. [S.l.]: Springer-Verlag, 2005: 1-12.

[8] Stevens M, Lenstra A, Weger B D. Chosen-prefix collisions for MD5 and colliding x.509 certificates for different identities[C]//LNCS 4515: Proc of the Eurocrypt 2007. [S.l.]: Springer-Verlag, 2007: 1-22.

[9] Daum M, Lucks D. The story of alice and bob[C]//LNCS 3494: The rump session of Eurocrypt 2005. [S.l.]: Springer-Verlag, 2005.

[10] Gebhardt M, Illies G, Schindler W. A note on the practical value of signal hash collisions for special file formats[C]// Proceedings of NIST Cryptographic Hash Workshop, 2005.

[11] Sasaki Y, Wang L, Ohta K, et al. New message difference for MD4[C]//LNCS 4593: Proc of the FSE 2007. [S.l.]: Springer-Verlag, 2007: 329-348.

[12] Schläffer, Oswald M. Searching for differential paths in MD4[C]//LNCS 4047: Proc of the FSE 2006. [S.l.]: Springer-Verlag, 2006: 242-261.

Appendix

In the appendix, give the tables for the message modification approach, ASCII character set and Latin-1 character set.

Table 10 The modification for correcting conditions on $a_{5,18}$

表 10 $a_{5,18}$ 的条件修改

15	m_{14}	11	c_4,d_4,a_4,b_3	
16	m_{15}	19	$b'_4=b_4\oplus 2^{14},c_4,d_4,a_4$	$m_{15}\leftarrow b'_4$ >>>19$-b_3-F(c_4,d_4,a_4)$
17	m_0	3	a'_5,b'_4,c_4,d_4	$c_{4,15}\neq d_{4,15}$

$P(m_{15,32}=1)=1/16$

Table 11 The modification for correcting conditions on $a_{5,29}$

表 11 $a_{5,29}$ 的条件修改

15	m_{14}	11	c_4,d_4,a_4,b_3	
16	m_{15}	19	$b'_4=b_4\oplus 2^{25},c_4,d_4,a_4$	$m_{15}\leftarrow b'_4$ >>>19$-b_3-F(c_4,d_4,a_4)$
17	m_0	3	a'_5,b'_4,c_4,d_4	$c_{4,26}\neq d_{4,26}$

$P(m_{15,32}=1)=1/2$

Table 12 The modification for correcting conditions on $d_{5,6}$

表 12 $d_{5,6}$ 的条件修改

15	m_{14}	11	c_4,d_4,a_4,b_3	
16	m_{15}	19	$b'_4=b_4\oplus 1,c_4,d_4,a_4$	$m_{15}\leftarrow b'_4$ >>>19$-b_3-F(c_4,d_4,a_4)$
17	m_0	3	a_5,b'_4,c_4,d_4	$c_{4,1}=d_{4,1}$
18	m_4	5	d'_5,a_5,b'_4,c_4	$a_{5,1}\neq c_{4,1}$

$P(m_{15,16}=1)=1/4,P(m_{15,24}=1)=1/16$

Table 13 The modification for correcting conditions on $d_{5,13}$

表 13 $d_{5,13}$ 的条件修改

15	m_{14}	11	c_4,d_4,a_4,b_3	
16	m_{15}	19	$b'_4=b_4\oplus 2^7,c_4,d_4,a_4$	$m_{15}\leftarrow b'_4$ >>>19$-b_3-F(c_4,d_4,a_4)$
17	m_0	3	a_5,b'_4,c_4,d_4	$c_{4,8}=d_{4,8}$
18	m_4	5	d'_5,a_5,b'_4,c_4	$a_{5,8}\neq c_{4,8}$

$P(m_{15,24}=1)=1/8,P(m_{15,8}=1)=1/64$

Table 14 The modification for correcting conditions on $c_{5,13}$

表 14 $c_{5,13}$ 的条件修改

15	m_{14}	11	$c'_4=c_4\oplus 8,d_4,a_4,b_3$	$m_{14}\leftarrow c'_4$ >>>11$-c_3-F(d_4,a_4,b_3)$
16	m_{15}	19	b_4,c'_4,d_4,a_4	$m_{15}\leftarrow b_4$ >>>19$-b_3-F(c'_4,d_4,a_4)$
17	m_0	3	a_5,b_4,c'_4,d_4	$b_{4,4}=d_{4,4}$
18	m_4	5	d_5,a_5,b_4,c'_4	$a_{5,4}\neq b_{4,4}$
19	m_8	9	c'_5,d_5,a_5,b_4	

$P(m_{14,32}=1)=1/128,P(m_{15,8}=1)=1/16$

Table 15 The modification for correcting conditions on $c_{5,18}$

表 15 $c_{5,18}$ 的条件修改

15	m_{14}	11	$c'_4=c_4\oplus 2^8,d_4,a_4,b_3$	$m_{14}\leftarrow c'_4$ >>>11$-c_3-F(d_4,a_4,b_3)$
16	m_{15}	19	b_4,c'_4,d_4,a_4	$m_{15}\leftarrow b_4$ >>>19$-b_3-F(c'_4,d_4,a_4)$
17	m_0	3	a_5,b_4,c'_4,d_4	$b_{4,9}=d_{4,9}$
18	m_4	5	d_5,a_5,b_4,c'_4	$a_{5,9}\neq b_{4,9}$
19	m_8	9	c'_5,d_5,a_5,b_4	

$P(m_{14,32}=1)=1/4,P(m_{15,16}=1)=1/128,P(m_{14,28}=1)=1/4$

Table 16 The set of ASCII characters

表 16 ASCII 字符表

Low 4 bits	High 4 bits							
	0000	0001	0010	0011	0100	0101	0110	0111
0000	NUL	DEL	SP	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EQT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	DEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	
1111	SI	US	/	?	O	_	o	DEL

Table 17 Latin-1(ISO 8859-1) character set

表 17 Latin-1(ISO 8859-1)字符表

Low 4 bits	High 4 bits													
	0	1	2	3	4	5	6	7	A	B	C	D	E	F
0	NUL	DEL	SP	0	@	P	‘	p	NBSP	°	À	Ð	à	
1	SOH	DC1	!	1	A	Q	a	q	¡	±	Á	Ñ	á	ñ
2	STX	DC2	"	2	B	R	b	r	¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s	£	³	Ã	Ó	ã	ó
4	EQT	DC4	$	4	D	T	d	t	¤	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u	¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	¦	¶	Æ	Ö	æ	ö
7	DEL	ETB	’	7	G	W	g	w	§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x	¨	¸	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y	©	¹	É	Ù	é	ù
A	LF	SUB	*	:	J	Z	j	z	a	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{	«	»	Ë	Û	ë	û
C	FF	FS	,	<	L	\	l	|	¬	¼	Ì	Ü	ì	ü
D	CR	GS	–	=	M	]	m	}	-	½	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~	®	¾	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL	¯	¿	Ï	ß	ï	ÿ

JIA Keting was born in 1983. She is a Ph.D. candidate at Shandong University. Her research interests include cryptanalysis of Hash functions and MACs, etc.

贾珂婷(1983-),女,山东青岛人,山东大学博士研究生,主要研究领域为 Hash 函数和 MAC 码的分析等。

WANG Xiaoyun was born in 1966. She is a professor and doctoral supervisor at Tsinghua University and Shandong University. Her research interests include cryptographic theory and technologies, etc.

王小云(1966-),女,山东诸城人,清华大学和山东大学教授、博士生导师,主要研究领域为密码理论和技术等。