Redundancy of the Wang-Yu Sufficient Conditions

Yuto Nakano* Hidenori Kuwakado† Masakatu Morii†

November 13, 2006

Abstract

Wang and Yu showed that MD5 was not collision-resistant, but it is known that their sufficient conditions for finding a collision of MD5 includes some mistakes. In this paper, we examine the sufficient conditions by computer simulation. We show that the Wang-Yu conditions include 16 unnecessary conditions for making a collision. Sasaki et al. claimed that modifying one condition made it possible to remove eleven conditions. However, the result of our computer simulation shows that their conditions does not make a collision.

1 Introduction

Hash functions are used in many cryptographic applications such as message authentication codes, digital signatures, and multi-party protocols. In order to offer secure cryptographic applications, it is necessary that hash functions satisfy following three properties. The first one is called one-wayness. This is the property that it is easy to compute a hash value from a message, but it is difficult to compute a message from the hash value. The second one is called second preimage resistance. This is the property that it is difficult to compute another message when a message and its hash values are given. The third one is called collision resistance. This is the property that it is difficult to find two (or more) messages which have the same hash value.

MD5 was designed by Rivest [1], and is currently a widely-used hash function. However, Wang and Yu [2] have shown that MD5 is not a collision-resistant hash function. Namely, they presented sufficient conditions for finding two messages which make a collision. Since the sufficient conditions, however, included some mistakes and redundancies, they have been examined in [3],[4]. Yajima and Shimoyama showed four conditions to be added and one condition to be corrected[3]. Sasaki et al. presented that modifying one condition made it possible to remove eleven conditions and that two conditions are not necessary[4].

In this paper, we report the result of numerical experiments on the sufficient conditions. Our results show that the conditions presented by Sasaki et al. are incorrect, that is, their conditions do not make the collision. Furthermore, our results show that 16 conditions of the Wang-Yu sufficient conditions are unnecessary. We show the theoretical reason that 14 out of 16 conditions are unnecessary.

2 Description of MD5

MD5 is a hash function that generates a 128-bit hash value from any length of a message and the 128-bit initial value. The message is padded so that its length is a multiple of 512 bits, and is divided into 512-bit message blocks. A compression function takes the message block and the initial value as two inputs, and the 128-bit output of the compression function is used as the initial value of the next compression function.

*Graduate School of Science and Technology, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, 657-8501 Japan.
E-mail: yuto-nakano@stu.kobe-u.ac.jp

†Faculty of Engneering, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, 657-8501 Japan.
E-mail: {kuwakado,mmorii}@kobe-u.ac.jp

The initial values are given as follows:

$$A = \texttt{0x67452301},\ B = \texttt{0xefcdab89},\ C = \texttt{0x98badcfe},\ D = \texttt{0x10325476}.$$

The compression function of MD5 has four rounds, and each round has 16 steps. Chaining values a, b, c, d are initialized as

$$a = A,\ b = B,\ c = C,\ d = D.$$

One of the chaining values is updated in each step, and computation is continued in sequence. Each step operation is as follows:

$$\begin{aligned} a &= b + ((a + f(b,c,d) + m + const) \lll s), \\ d &= a + ((d + f(a,b,c) + m + const) \lll s), \\ c &= d + ((c + f(d,a,b) + m + const) \lll s), \\ b &= c + ((b + f(c,d,a) + m + const) \lll s), \end{aligned}$$

where the operation + means addition modulo 2^{32}, *const* and s are step-dependent constants. m is a 32-bit message word where the 512-bit message block is divided into 16 32-bit message words. "$x \lll s$" is left cyclic shift of x by s bit positions. f is a round-dependent function:

$$\begin{aligned} \text{Round 1} : f &= F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z), \\ \text{Round 2} : f &= G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z)), \\ \text{Round 3} : f &= H(X,Y,Z) = X \oplus Y \oplus Z, \\ \text{Round 4} : f &= I(X,Y,Z) = Y \oplus (X \vee (\neg Z)). \end{aligned}$$

We use following symbols in this paper.
$a_i,\ b_i,\ c_i,\ d_i$: the i-th values of a, b, c, d $(1 \le i \le 16)$.
$a_{i,j},\ b_{i,j},\ c_{i,j},\ d_{i,j}$: the j-th bit of a_i, b_i, c_i, d_i $(1 \le j \le 32)$.
$aa_0,\ bb_0,\ cc_0,\ dd_0$: outputs of the compression function in the first message block.
$aa_1,\ bb_1,\ cc_1,\ dd_1$: outputs of the compression function in the second message block.
ϕ_k : an output of function$f$ in step k $(0 \le k \le 63)$.
$\phi_{k,j}$: the j-th bit of ϕ_k $(0 \le k \le 63,\quad 1 \le j \le 32)$.
v_k : a value before the left cyclic shift $(0 \le k \le 63)$.
u_k : a value after the left cyclic shift $(0 \le k \le 63)$.
m_ℓ : the ℓ-th message word which is divided from a message block by 32 bits $(0 \le \ell \le 15)$.
$x_i[j]$: the j-th bit of x changes from 0 to 1.
$x_i[-j]$: the j-th bit of x changes from 1 to 0.
$x \lll s$: a left cyclic shift by s bits.
$x \ggg s$: a right cyclic shift by s bits.

3 Result and Discussion

3.1 Redundant Conditions

Using the program based on [5], we generated 1000 pairs of collision messages. It took about two hours for finding one pair. After then, we checked if they satisfy the sufficient conditions. The result of our experiment shows that 16 conditions are unnecessary for making a collision. Table 1 and Table 2 show 16 unnecessary conditions. For example, the condition of $a_{2,21} = 0$ is satisfied by 56 pairs, but it is not done by 944 pairs. It follows that the ratio is 94.4% (= 944/1000). From Table 1 and Table 2, we see that the ratio depends on the condition. Assuming that the other part of MD5 is ideally random, we theoretically explain the ratio in Table 1 and Table 2.

Conditions in Table 1

Table 1: Unnecessary condition and ratio (the first message block)

No.	Condition	Ratio[%]
1	$a_{2,21} = 0$	94.4
2	$a_{2,22} = 0$	53.8
3	$a_{2,23} = 1$	49.3
4	$b_{1,22} = c_{1,22}$	45.6
5	$b_{1,23} = c_{1,23}$	53.8
6	$d_{2,22} = 1$	48.1
7	$d_{2,23} = 1$	51.4
8	$c_{2,22} = 1$	47.9
9	$c_{2,23} = 1$	48.6
10	$\phi_{34,32} = 0$	50.3
11	$a_{16,27} = 0$	28.1
12	$c_{16,32} = d_{16,32}$	76.4

Table 2: Unnecessary condition and ratio (the second message block)

No.	Condition	Ratio[%]
1	$\phi_{34,32} = 1$	52.4
2	$d_{16,26} = 1$	5.1
3	$c_{16,26} = 1$	27.3
4	$b_{16,26} = 1$	53.2

1. Condition $a_{2,21} = 0$

   This condition is used to convey the bit carry on the 7th bit to the 23rd bit in the computation of $\Delta a_2 = a_2' - a_2$. However, we changed a_2' as $a_2[7, 8, \cdots, 20, -21]$ from $a_2[7, 8, \cdots, 22, -23]$, $a_{2,21} = 0$ is incorrect. Thus, $a_{2,21}$ must be equal to 1. The reason that there are some collision messages which satisfy $a_{2,21} = 0$ is that a carry of a_2 starts on the 7th bit to the 22nd bit. That is, $a_2' = a_2[7, 8, \cdots, 21, -22]$.

2. Condition $a_{2,22} = 0$

   Since we shortened the length of carry of a_2, it is unnecessary to satisfy $a_{2,22} = 0$. Furthermore, since this bit is selected at random, the probability that $a_{2,22}$ equals 0 is 1/2. This agrees well with the ratio in Table 1.

3. Condition $a_{2,23} = 1$

   In a similar way to $a_{2,22} = 0$, the modification $a_2' = a_2[7, 8, \cdots, 20, -21]$ makes it unnecessary to satisfy $a_{2,23} = 1$. Furthermore, since this bit is selected at random, the probability that this bit equals 1 is 1/2.

4. Condition $b_{1,22} = c_{1,22}$

   Since the carry of a_2 can be stopped at the 21st bit, the calculation of $\Delta\phi_{5,22}$ is shown in eq. (1).

$$\begin{aligned} \Delta\phi_{5,22} &= (a_{2,22} \wedge b_{1,22}) \vee (\neg a_{2,22} \wedge c_{1,22}) - (a_{2,22} \wedge b_{1,22}) \vee (\neg a_{2,22} \wedge c_{1,22}) \\ &= 0. \end{aligned} \tag{1}$$

   Therefore, we can get $\Delta\phi_{5,22} = 0$ without $b_{1,22} = c_{1,22}$. Since $b_{1,22}$ and $c_{1,22}$ are chosen at random, the probability that $b_{1,22}$ equals $c_{1,22}$ is 1/2.

5. Condition $b_{1,23} = c_{1,23}$

   From Table 3 in [2], $\Delta d_{2,3} = 0$ must be satisfied. Since Δm_5, Δd_1, and $\Delta a_{2,3}$ are 0, $\Delta\phi_{5,23} = 0$ is necessary to get $\Delta d_{2,3} = 0$. The derivation of $\Delta\phi_{5,23}$ is expressed as follows:

$$\begin{aligned}\Delta\phi_{5,23} &= (a'_{2,23} \wedge b_{1,23}) \vee (\neg a'_{2,23} \wedge c_{1,23}) - (a_{2,23} \wedge b_{1,23}) \vee (\neg a_{2,23} \wedge c_{1,23}) \\ &= 0, \end{aligned} \tag{2}$$

   where $a'_{2,23} = a_{2,23}$. From eq.(2), we can get $\Delta\phi_{5,23} = 0$ without $b_{1,23} = c_{1,23}$. Moreover, $b_{1,23}$ and $c_{1,23}$ are chosen at random, the probability that $b_{1,23}$ equals $c_{1,23}$ is 1/2.

6. Condition $d_{2,22} = 1$

   Since the carry of a_2 can be stopped at the 21st bit, $\Delta a_{2,22}$ is 0 and other inputs have no difference on the 22nd bit. That is, the calculation of $\Delta\phi_{6,22}$ is expressed as:

$$\begin{aligned}\Delta\phi_{6,22} &= (d_{2,22} \wedge a_{2,22}) \vee (\neg d_{2,22} \wedge b_{1,22}) - (d_{2,22} \wedge a_{2,22}) \vee (\neg d_{2,22} \wedge b_{1,22}) \\ &= 0. \end{aligned} \tag{3}$$

   Eq.(3) means that no condition is necessary to get $\Delta\phi_{6,22} = 0$. Since $d_{2,22}$ is chosen at random, the probability that $d_{1,22}$ equals 1 is 1/2.

7. Condition $d_{2,23} = 1$

   In a similar way to $d_{2,22} = 1$, this condition also become unnecessary by stopping the carry of a_2 at the 21st bit. The probability that $d_{2,23}$ equals 1 is 1/2 because this bit is selected at random.

8. Condition $c_{2,22} = 1$

   We have no idea why this condition is unnecessary, and this is future work.

9. Condition $c_{2,23} = 1$

   We have no idea why this condition is unnecessary, and this is future work.

10. Condition $\phi_{34,32} = 0$

    This condition is used in step 34 (in this step, c_9 is calculated) in the first message block. In this step, each difference of chaining value is expressed as $\Delta c_8 = 0$, $\Delta d_9 = 0$, $\Delta a_9 = 0$, $\Delta b_8 = 0$. This means that $\Delta\phi_{34}$ must be 0. The message word used in this step is m_{11}, whose difference Δm_{11} is 2^{15}, and after 16 bits left cyclic shift this difference is in the 32nd bit. Therefore, $\Delta c_9 = 2^{31}$ is achieved without $\phi_{34,32} = 0$. All input values to ϕ_{34} is selected at random, the probability that $\phi_{34,32}$ equals 0 must be 1/2.

11. Condition $a_{16,27} = 0$

    This condition is used to control the output of a non-linear function in step 63 (in this step, b_{16} is calculated). The output difference in this step is $\Delta b_{16} = 2^{31} + 2^{25}$. To achieve the ideal difference, we need $\Delta\phi_{61} = 2^{31}$. This is easily proved by following equation:

$$\begin{aligned}\Delta u_{63} &= \Delta b_{16} - \Delta c_{16} = 2^{31} + 2^{25} - 2^{31} - 2^{25} = 0, \\ \Delta v_{63} &= \Delta u_{63} \ggg 21, \\ \Delta\phi_{63} &= \Delta v_{63} - \Delta m_9 - b_{15} = 2^{31}. \end{aligned}$$

    Furthermore, $\Delta\phi_{63,27}$ is expressed as follows:

$$\Delta\phi_{63,27} = d'_{16,27} \oplus (c'_{16,27} \vee (\neg a'_{16,27})) - d_{16,27} \oplus (c_{16,27} \vee (\neg a_{16,27})).$$

    From Table 3 in [2], there is no difference on $d_{16,27}$ and $a_{16,27}$, that is,

$$\Delta\phi_{63,27} = d_{16,27} \oplus (c'_{16,27} \vee (\neg a_{16,27})) - d_{16,27} \oplus (c_{16,27} \vee (\neg a_{16,27})).$$

If $c'_{16,27}$ does not equal to $c_{16,27}$, $a_{16,27} = 0$ must be satisfied to get $\Delta\phi_{63,27} = 0$. Some collision messages, however, do not have difference on the 27th bit of c_{16}. Those messages do not have to satisfy $a_{16,27} = 0$.

12. Condition $c_{16,32} = d_{16,32}$

    This condition is probably used to control the output of non-linear function in step 63. From the discussion of $a_{16,27} = 0$, we know that $\Delta\phi_{63}$ has to be 2^{31}. $\Delta\phi_{63,32}$ is expressed as follows:

$$\Delta\phi_{63,32} = d'_{16,32} \oplus (c'_{16,32} \vee (\neg a'_{16,32})) - d_{16,32} \oplus (c_{16,32} \vee (\neg a_{16,32})). \quad (4)$$

    From eq.(4), it is easily proved that $c_{16,32} = a_{16,32}$ is necessary to get $\Delta\phi_{61,32} = 1$. However, the condition listed in Table 3 in [2] is $c_{16,32} = d_{16,32}$, and this has to be corrected to $c_{16,32} = a_{16,32}$. The output of a compression function in the first message block is expressed as follows:

$$cc_0 = c_{16} + C,\ dd_0 = d_{16} + D. \quad (5)$$

    We denote the 32nd bit of C and D as C_{32} and D_{32}, respectively. We have three sufficient conditions for cc_0, and one of them is $cc_{0,32} = dd_{0,32}$. If there is no carry to the 32nd bit from lower bits in the addition expressed in eq.(5), $c_{16,32} \neq d_{16,32}$ is needed to achieve $cc_{0,32} = dd_{0,32}$ because of $C_{32} = 1$ and $D_{32} = 0$. The probability which there is carry from lower bit to the 32nd bit in the addition of eq.(5) is calculated as

$$\frac{1}{4}(2^0 + 2^{-1} + 2^{-2} + \cdots + 2^{-30}) = 0.499\ldots \approx \frac{1}{2}.$$

    Only when there is a carry either cc_0 or dd_0, $c_{16,32} = d_{16,32}$ is satisfied. Therefore, the probability that $c_{16,32}$ equals $d_{16,32}$ is 1/4.

Conditions in Table 2

1. Condition $\phi_{34,32} = 1$

    This condition is used in step 34 (in this step, c_9 is calculated) in the second message block. In this step, each difference of chaining value is expressed as $\Delta c_8 = 0,\ \Delta d_9 = 0,\ \Delta a_9 = 0,\ \Delta b_8 = 0$. This means that $\Delta\phi_{34}$ must be 0. The message word used in this step is m_{11} and its difference is -2^{15}. After 16 bits cyclic shift, this difference is on the 32nd bit. Therefore, $\Delta c_9 = -2^{31} = 2^{31}$ is achieved without $\phi_{34,32} = 1$. Since $\phi_{34,32}$ is selected at random, the probability that $\phi_{34,32}$ equals 1 is 1/2.

2. Condition $d_{16,26} = 1$

    In step 61, we have $dd_{0,26} = 0, \Delta dd_{0,26} = 1$ and we need $dd'_{1,26} = dd_{1,26}$. $dd'_{1,26}$ and $dd_{1,26}$ is expressed as follows:

$$dd_1 = dd_0 + d_{16},\ dd'_1 = dd'_0 + d'_{16}.$$

    There are following two cases to achieve $dd_{1,26} = dd'_{1,26}$.

    - There are a bit carry from the 25th bit in one-sided and $d_{16,26}$ equals $d'_{16,26}$.
    - $d_{16,26}$ does not equal $d'_{16,26}$.

    However there cannot be a carry in one-sided, since until the 25th bit dd_0 equals dd'_0 and d_{16} equals d'_{16}. To achieve $\Delta d_{16} = 2^{31} - 2^{25}$, $d_{16,26} \neq d'_{16,26}$ must be satisfied. Since $2^{31} - 2^{25}$, this is the difference on d_{16}, is a subtraction, there are probable differences more than one:

$$2^{31} - 2^{25} \quad = \quad 2^{31} - 2^{26} + 2^{25} = \cdots = 2^{31} - 2^{30} + 2^{29} + \cdots + 2^{25}.$$

    Whichever difference Δd_{16} takes, $d_{16,26} \neq d'_{16,26}$ is always satisfied. When the difference is $2^{31} - 2^{25}$, $d_{16,26}$ must be 1, but when the difference is $2^{31} - 2^{26} + 2^{25}$ and so on, $d_{16,26} = 1$ does not have to be satisfied. In fact, collision messages whose difference is $2^{31} - 2^{26} + 2^{25}$ do not have to satisfy $d_{16,26} = 1$.

3. Condition $c_{16,26} = 1$

   We need $c_{16,26} \neq c'_{16,26}$ to achieve the ideal difference; $\Delta c_{16} = 2^{31} - 2^{25}$, in a similar way to $d_{16,26} = 1$. Δc_{16} can also have some difference that is expressed as follows:

$$2^{31} - 2^{25} = 2^{31} - 2^{26} + 2^{25} = \cdots = 2^{31} - 2^{30} + 2^{29} + \cdots + 2^{25}.$$

   Whichever difference Δc_{16} takes, $c_{16,26} \neq c'_{16,26}$ is always satisfied. When the difference is $2^{31} - 2^{25}$, $c_{16,26}$ must be 1, but when the difference is $2^{31} - 2^{26} + 2^{25}$ and so on, $c_{16,26} = 1$ does not have to be satisfied.

4. Condition $b_{16,26} = 1$

   In a similar way to $d_{16,26}$, We need $b_{16,26} \neq b'_{16,26}$ to achieve the ideal difference; $\Delta b_{16} = 2^{31} - 2^{25}$. Δb_{16} can also have some difference that is expressed as follows:

$$2^{31} - 2^{25} = 2^{31} - 2^{26} + 2^{25} = \cdots = 2^{31} - 2^{30} + 2^{29} + \cdots + 2^{25}.$$

   Whichever difference Δb_{16} takes, $b_{16,26} \neq b'_{16,26}$ is always satisfied. When the difference is $2^{31} - 2^{25}$, $b_{16,26}$ must be 1, but when the difference is $2^{31} - 2^{26} + 2^{25}$ and so on, $b_{16,26} = 1$ does not have to be satisfied.

3.2 Modification by Sasaki et al.

Sasaki et al.[4] claimed that modifying $a_{2,20} = 0$ to $a_{2,20} = 1$ made it possible to remove $b_{1,21} = c_{1,21}$, $b_{1,22} = c_{1,22}$, $b_{1,23} = c_{1,23}$, $a_{2,21} = 0$, $a_{2,22} = 0$, $a_{2,23} = 1$, $d_{2,21} = 1$, $d_{2,22} = 1$, $d_{2,23} = 1$, $c_{2,22} = 1$, and $c_{2,23} = 1$. However we were not able to find a collision using this modification. In contrast, when we remained $a_{2,20}$ as $a_{2,20} = 0$ and removed above eleven conditions, we made collision messages. Since we found collision messages as $a_{2,20}$ equals 0, changing $a_{2,20}$ to $a_{2,20} = 1$ is incorrect. We also found that $b_{1,21} = c_{1,21}$ and $d_{2,21} = 1$ are necessary because all collision messages satisfy both conditions.

Condition $b_{1,21} = c_{1,21}$ is used in step 5 (in this step, d_2 is calculated) in the first message block. The non-linear function in this step is showed as follows:

$$\phi_5 = (a_2 \wedge b_1) \vee (\neg a_2 \wedge c_1).$$

From this equation, $\Delta\phi_5$ is expressed as follows:

$$\Delta\phi_5 = (a'_2 \wedge b'_1) \vee (\neg a'_2 \wedge c'_1) - (a_2 \wedge b_1) \vee (\neg a_2 \wedge c_1). \quad (6)$$

Since $b'_1 = b_1, c'_1 = c_1$, eq.(6) can be simplified:

$$\Delta\phi_5 = (a'_2 \wedge b_1) \vee (\neg a'_2 \wedge c_1) - (a_2 \wedge b_1) \vee (\neg a_2 \wedge c_1). \quad (7)$$

Since the 12 bits left cyclic shift is done in step 5,the difference on $\Delta\phi_{5,21}$ affects $\Delta d_{2,1}$. From Table 3 in [2], $\Delta d_{2,1} = 0$ must be satisfied. The message word used in this step is m_5, whose differential Δm_5 is 0. In addition, both of Δd_1 and $\Delta a_{2,1}$ are 0. Therefore, $\Delta\phi_{5,21}$ must be 0 to get $\Delta d_{2,1} = 0$. Since $a'_{2,21} \neq a_{2,21}$, $b_{1,21} = c_{1,21}$ must be satisfied to ensure $\Delta\phi_{5,21} = 0$.

Condition $d_{2,21} = 1$ is used in step 6 (in this step, c_2 is calculated) in the first message block. Each chaining value is expressed as follows:

$$\begin{cases} c'_1 = c_1 \\ b'_1 = b_1 \\ a'_2 = a_2[7, 8, \ldots, 20, -21] \\ d'_2 = d_2[-7, 24, 32] \end{cases}$$

According to the operations in step 6, we have

$$\begin{aligned}\Delta u_6 &= \Delta c_2 - \Delta d_2 \\ &= -1 - 2^{27} - 2^{31} = -1 - 2^{27} + 2^{31}, \\ \Delta v_6 &= \Delta u \ggg 17, \\ \Delta \phi_6 &= -2^{10} - 2^{14}.\end{aligned}$$

And we also have

$$\phi'_6 = \phi_6[11, \ldots, 14, 16, \ldots 20, -21].$$

The derivation of $\Delta\phi_{6,21}$ is given as following expression:

$$\begin{aligned}\Delta\phi_{6,21} &= (d_{2,21} \wedge a'_{2,21}) \vee (\neg d_{2,21} \wedge b_{1,21}) - (d_{2,21} \wedge a_{2,21}) \vee (\neg d_{2,21} \wedge b_{1,21}) \\ &= -1. \end{aligned} \quad (8)$$

Now that we have a differential on $a_{2,21}$, $d_{2,21} = 1$ is necessary to get a differential on $\phi_{6,21}$. This can be proved from eq.(8).

4 Examples

Table 3 and Table 4 show the collision messages that do not satisfy the conditions discussed in Section 3. The message in Table 3 does not satisfy the following conditions: $a_{2,21} = 0$, $a_{2,23} = 1$, $b_{1,22} = c_{1,22}$, $b_{1,23} = c_{1,23}$, $d_{2,23} = 1$, $c_{2,23} = 1$, $\phi_{34,32} = 0$, $a_{16,27} = 0$, and $c_{16,32} = d_{16,32}$ in the first message block and $\phi_{34,32} = 1$, $c_{16,26} = 1$, and $b_{16,26} = 1$ in the second message block. The message in Table 4 does not satisfy the following conditions: $a_{2,21} = 0$, $a_{2,22} = 0$, $b_{1,23} = c_{1,23}$, $d_{2,22} = 1$, $c_{2,22} = 1$, $a_{16,27} = 0$, and $c_{16,32} = d_{16,32}$ in the first message block and $d_{16,26} = 1$, $c_{16,26} = 1$, and $b_{16,26} = 1$ in the second message block.

5 Conclusion

In this paper, we have examined the sufficient conditions showed by Wang and Yu by computer simulation. Our results show that the Wang-Yu conditions include 16 unnecessary conditions. Hence, the number of conditions for making a collision is reduced to 587 from 599 which is the number of the Wang-Yu sufficient conditions, from 603 which is that of the Yajima-Shimoyama sufficient conditions. Our results also show that the modification of Sasaki et al. does not make a collision. We have also explained why 14 out of 16 conditions are unnecessary.

References

[1] R. Rivest, "The MD5 message-digest algorithm," RFC1321, 1992.

[2] X. Wang and H. Yu,"How to break MD5 and other hash functions," http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf, 2004.

[3] J. Yajima, T. Simoyama, "On the conllision search and the sufficient conditions of MD5," IEICE Technical Report, ISEC2005-78, pp.15–22 2005.

[4] Y. Sasaki, Y. Naito, J. Yajima, T. Shimoyama, N. Kunihiro, and K. Ohta, "How to construct sufficient conditions in searching collisions of MD5," http://eprint.iacr.org/2006/074.pdf, 2006

[5] P. Stach and V. Liu,"MD5 collision generation," http://www.stachliu.com/collisions.html, 2005.

Table 3: Example of collision message1

M_0	0x319266e1 0xa510ab61 0x4d77995a 0x783f70a9
	0x7969e536 0x37b326c0 0x38602559 0x1a805a22
	0x06a3ed35 0x84b287f5 0x9db0b2f3 0x4ee85945
	0x0779dedb 0x96fb6219 0x2d2ebda8 0x138c3808
M_0'	0x319266e1 0xa510ab61 0x4d77995a 0x783f70a9
	0xf969e536 0x37b326c0 0x38602559 0x1a805a22
	0x06a3ed35 0x84b287f5 0x9db0b2f3 0x4ee8d945
	0x0779dedb 0x96fb6219 0xad2ebda8 0x138c3808
M_1	0x298ab32b 0xfa045ee6 0x0cd5aa54 0x8385602e
	0xb42453d0 0xfc45a101 0x07a3c936 0xde2e06d3
	0x24bb1081 0xcc92d309 0xbaf359aa 0xdd72ac7d
	0x878230e9 0xb253c565 0x3ca9267a 0x6e30ead6
M_1'	0x298ab32b 0xfa045ee6 0x0cd5aa54 0x8385602e
	0x342453d0 0xfc45a101 0x07a3c936 0xde2e06d3
	0x24bb1081 0xcc92d309 0xbaf359aa 0xdd722c7d
	0x878230e9 0xb253c565 0xbca9267a 0x6e30ead6
H	aa_1=0x5109de9a bb_1=0xb6a5b8b8
	cc_1=0xff2dc0bc dd_1=0xcf2cb392

Table 4: Example of collision message2

M_0	0x1900cc51 0x4d0723e4 0x3bc88f2d 0x48cef048
	0xc2029e01 0xd6ce03d6 0x122d2978 0xf3865e39
	0x0623ed51 0x63b44608 0x79bee933 0x12614f84
	0xe0a0e0df 0x4f491932 0x0f0aafcc 0x83fc7b52
M_0'	0x1900cc51 0x4d0723e4 0x3bc88f2d 0x48cef048
	0x42029e01 0xd6ce03d6 0x122d2978 0xf3865e39
	0x0623ed51 0x63b44608 0x79bee933 0x1261cf84
	0xe0a0e0df 0x4f491932 0x8f0aafcc 0x83fc7b52
M_1	0x496bd84e 0x17d041c1 0x41b2dcb9 0x072fbeb7
	0x28198e60 0x2d839611 0x16389948 0xc200d954
	0xa1fc7775 0x2b31d7f4 0x5747bd9a 0x7cdde0a2
	0xd540c0e2 0xa30bb88b 0x4b687cb7 0x04eddf27
M_1'	0x496bd84e 0x17d041c1 0x41b2dcb9 0x072fbeb7
	0xa8198e60 0x2d839611 0x16389948 0xc200d954
	0xa1fc7775 0x2b31d7f4 0x5747bd9a 0x7cdd60a2
	0xd540c0e2 0xa30bb88b 0xcb687cb7 0x04eddf27
H	aa_1=0x24ef91f2 bb_1=0x0b8b1b33
	cc_1=0x1e745fd4 dd_1=0xa1757c32