Improved Collision Search for SHA-0

Yusuke Naito[1], Yu Sasaki[1], Takeshi Shimoyama[2],
Jun Yajima[2], Noboru Kunihiro[1], and Kazuo Ohta[1]

[1] The University of Electro-Communications, Japan
{tolucky,yu339,kunihiro,ota}@ice.uec.ac.jp
[2] FUJITSU LABORATORIES LTD
{shimo,jyajima}@labs.fujitsu.com

Abstract. At CRYPTO2005, Xiaoyun Wang, Hongbo Yu and Yiqun Lisa Yin proposed a collision attack on SHA-0 that could generate a collision with complexity 2^{39} SHA-0 hash operations. Although the method of Wang et al. can find messages that satisfy the sufficient conditions in steps 1 to 20 by using message modification, it makes no mention of the message modifications needed to yield satisfaction of the sufficient conditions in steps 21 and onwards.
In this paper, first, we give sufficient conditions for the steps from step 21, and propose submarine modification as the message modification technique that will ensure satisfaction of the sufficient conditions from steps 21 to 24. Submarine modification is an extension of the multi-message modification used in collision attacks on the MD-family. Next, we point out that the sufficient conditions given by Wang et al. are not enough to generate a collision with high probability; we rectify this shortfall by introducing two new sufficient conditions. The combination of our newly found sufficient conditions and submarine modification allows us to generate a collision with complexity 2^{36} SHA-0 hash operations. At the end of this paper, we show the example of a collision generated by applying our proposals.

keywords: SHA-0, Collision Attack, Message Modification, Sufficient Condition

1 Introduction

SHA-0 is the hash function issued by NIST in 1993 [5]. All hash functions must hold 3 properties: Pre-image Resistance, Second Pre-image Resistance and Collision Resistance. Collision Resistance means that it is very hard to find x, y such that $x \neq y$ and $H(x) = H(y)$, where $H(\cdot)$ is any hash function. Collision Resistance is more difficult to keep than any other property. The Collision Resistance of SHA-0 was broken recently [2]. This paper uses the term Collision Attack to refer to attacks that break Collision Resistance.

The first collision attack on SHA-0 was proposed by F. Chabaud and A. Joux in 1998 [3]. They employed differential attack and used XOR as the differential. After that, E. Biham and R. Chen improved [3], and found near collisions [1].

Near collision means x, y such that $x \neq y$ and $H(x), H(y)$ differ only by a small number of bits. At the rump session of CRYPTO2004, the first announcement of finding a collision of SHA-0 was made by A. Joux [4]. Details of this attack were presented in EUROCRYPT2005 by E. Biham, R. Chen, A. Joux, P. Carribault, W. Jalby and C. Lemuet [2]. In 2004, Wang proposed an independent collision attack method on SHA-0 [10,11]. Wang's method uses the differential attack approach in which numerical operations are used as the differential. Subsequently, X. Wang, H. Yu and Y. Lisa Yin proposed an improved version of Wang's attack [14]. This method has complexity of 2^{39} SHA-0 hash operations, and is the most efficient collision attack method proposed so far.

The method of Wang et al. can be divided into 2 phases. In the pre-computation phase, a differential path and conditions that indicate that a collision is possible are constructed. In this paper, we call these conditions "sufficient conditions". Sufficient conditions define the triggers for ending collision search. In the collision search phase, an input message satisfying all sufficient conditions is searched for. If this message is found, a collision can be generated. In this phase, message modification is used to efficiently find a message that satisfies the sufficient conditions.

According to Wang et al., in the case of SHA-0, a message satisfying sufficient conditions from steps 1 to 20 can be located efficiently by using message modification. The specification of SHA-0 states that the messages used in steps 1-16 are input messages, whereas messages used in steps after 16 are determined by message expansion as is defined by the specification of SHA-0. In the method of Wang et al., messages satisfying the sufficient conditions in steps 1-16 can, with probability 1, be generated by using message modification. Since steps 1-16 are not affected by the limitations placed on message expansion, it is possible to choose values of chaining variables to satisfy all sufficient conditions, and then calculate messages that can yield these chaining variables. Regarding the sufficient conditions in steps 17-20, if these conditions are not satisfied and message modification is executed, these sufficient conditions are satisfied with probability of almost 1. Since the steps from 17 are affected by message expansion, the message modification in steps after 16 proposed by Wang et al., is executed by generating the differential in the step not affected by message expansion. Since this differential (We call this differential "transmission differential") is transferred to subsequent steps, sufficient conditions are satisfied by the transferred differential. We call this method "transmission method". Without using these methods, the probability that a sufficient condition is satisfied in 1 time is $\frac{1}{2}$. For example, suppose there exists 1 condition in step i and the complexity to calculate all necessary operations up to step i is j steps. In this case, the number of steps needed to ensure the success of step i is $2j$ (on average). By using these methods, if the complexity of message modification is p steps, the number of steps needed to ensure the success of step i is $j + \frac{1}{2} \cdot p$ (on average). Since we choose message modification such that the complexity is $p < j$, message modification reduces the complexity by $j - \frac{1}{2} \cdot p$ steps. Therefore, we can efficiently locate a collision by using message modification. Note that message modification in the steps after 16 is particularly important in reducing the complexity of collision search.

Our Results

Our paper makes 2 contributions.

1st Result: Wang et al. have not proposed message modification to satisfy the sufficient conditions from step 21; their solution is random search. In this paper, we propose message modification for steps 21-24. We call this proposal "submarine modification". It takes advantage of the ideas of multi-message modification for the MD-family (we call multi-message modification for the MD-family "cancel method") and transmission method (Details are described below). Since the same discussion about the complexity of message modification made with regard to the proposal of Wang et al., discussed above, can be applied to submarine modification, submarine modification can more efficiently satisfy the sufficient conditions than random search. Since the structure of the MD-family or SHA-1 is very similar to that of SHA-0, submarine modification may also be applicable to those hash functions.

2nd Result: We show that the sufficient conditions given by Wang et al. are missing two conditions, and then describe the missing sufficient conditions.

From the second result, even if a message satisfying all sufficient conditions given by Wang et al. is found, collision search does not always succeed. Since their conditions are two short, their method will fail with probability $\frac{3}{4}$. We identify the two missing sufficient conditions and use them with our submarine modification proposal to search for a collision. Considering the fact that the number of sufficient conditions in steps 21-24 is 4 and given the complexity of submarine modification, a computer experiment finds that our method finds a collision with complexity 2^{36} SHA-0 hash operations. The PC used had a Pentium4 3.4GHZ CPU(OS: Linux 2.6.9 (Fedora Core 3, Red Hat 3.4.2), Compiler: gcc 3.4.2-i386). In the fastest case, a collision was found in 8 hours. The average time to find a collision was roughly 100 hours.

Overview of Our Main Idea: Submarine Modification

Submarine modification uses two ideas of message modifications, "transmission method" and "cancel method". We can satisfy sufficient conditions for up to step 24 by using submarine modification.

"Transmission method" is the method that can satisfy sufficient conditions for up to step 21 of SHA-0 (Wang et al. apply transmission method to sufficient condition for steps 17-20. We confirm that transmission method is applicable to satisfy sufficient conditions for steps 17-21). Namely, transmission method can satisfy sufficient conditions for 5 steps from a start step of transmission.

"Cancel method" is the method that uses the idea of the local collision. The local collision is the method where we create a differential and offset the differential in within several. We construct the method that inputs differentials and offsets the effects of these differentials before step 16 such that the differential (we call this differential "latent differential") appears again from step 17 due to message expansion after the differential is offset. Differentials don't appear for steps between the step where the differential offsets and the step where the latent differential appears. We call these steps "latent period". We denote the number of steps in latent period after step 17 as t. Influence of differentials created before the step where the latent differential appears does not occur. Cancel method is the method with which the sufficient condition for the step where the latent differential appears is satisfied by using the latent differential. We use the idea of cancel method in order to allow the start step of transmission to locate between step 17 to step 19. Note that cancel method itself does not use transmission of the latent differential.

The method that we propose in this paper satisfies sufficient conditions for up to step 24. If we use transmission method to satisfy sufficient conditions for up to step 24, we need to extend the range where the transmission differential can be started from step 16 to step 19. We can realize it by using the idea of cancel method. Since maximum number of latent period after step 17 for SHA-0 is $t = 3$, we can extent the range of the start step of transmission from step 16 to step 19 if we adopt the transmission differential as the latent differential. The latent differential can be created by using cancel method. Since there exists no influence for satisfied sufficient conditions in latent period by using cancel method, and we can satisfy sufficient conditions for 5 steps from the start step of transmission by applying transmission method. Since this method takes advantage of the differentials whose local effects are cancelled in the earlier steps, we call this message modification technique "submarine modification".

2 Structure of SHA-0[5]

SHA-0 is a hash function issued by NIST in 1993. SHA-0 has the Merkle-Damgård structure, therefore, it repeatedly applies a compression function. SHA-0 input is an arbitrary length message M, and SHA-0 output is 160 bit data $H(M)$. If the length of the input message is not a multiple of 512, the message is padded to realized a multiple of 512 bits. The padding process is $M^* = M||10...0$. First, 1 is added, and then as many 0's as are needed. Padded message M^* is divided into several messages M_i each 512 bits long ($M^* = (M_1||M_2||...||M_n)$). These divided messages are input to the compression function.

$$h_1 = compress(M_1, IV) \rightarrow h_2 = compress(M_2, h_1) \rightarrow \cdots \rightarrow h_n = compress(M_n, h_{n-1})$$
$$H(M) = h_n$$

In this paper, we call the calculation performed in a single run of the compression function 1 block. IV in the above expression is defined as $(a_0, b_0, c_0, d_0, e_0) =$ (0x67452301,0x$efcdab$89,0x98$badcfe$,0x10325476,0xc3d2e1f0). We next explain the structure of the compression function of SHA-0. All calculations in this are 32-bit. In this paper, we exclude the description of "mod 2^{32}".

Procedure 1. Divide the input message M_j into 32 bit messages $m_0, m_1, ..., m_{15}$.

Procedure 2. Calculate m_{16} to m_{79} by $m_i = m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}$

Procedure 3. Calculate chaining variables a_i, b_i, c_i, d_i, e_i in step i by the following procedures.

$$a_i = (a_{i-1} \lll 5) + f(b_{i-1}, c_{i-1}, d_{i-1}) + e_{i-1} + m_{i-1} + k_{i-1},$$
$$b_i = a_{i-1}, c_i = b_{i-1} \lll 30, d_i = c_{i-1}, e_i = d_{i-1}$$

"$\lll j$" denotes left cyclic shift by j bits. Repeat this process 80 times. Initial values a_0, b_0, c_0, d_0, e_0 for the compression function of the first block are IV. a_0, b_0, c_0, d_0, e_0 for the compression function from the second block are the output values of the previous block. Steps 1-20 are called the first round. Steps 21-40, 41-60, and 61-80 are the second, third, and fourth rounds, respectively, k_i is a constant defined in each round. Function f is a boolean function defined in each round. The specifications of k_i and f are shown in Table 1.

round	function$f$	constant k_i
1	$(b \wedge c) \vee (\neg b \wedge d)$	0x5a827999
2	$b \oplus c \oplus d$	0x6ed9eba1
3	$(b \wedge c) \vee (c \wedge d) \vee (d \wedge b)$	0x8f1$bbcdc$
4	$b \oplus c \oplus d$	0xca62c1d6

Table 1. Function f and Constants k in SHA-0

Procedure 4. $(a_0 + a_{80}, b_0 + b_{80}, c_0 + c_{80}, d_0 + d_{80}, e_0 + e_{80})$ is the output of the compression function.

3 Collision Attack by Wang et al.[8,9,14,15]

The method of Wang et al. is based on differential attack which uses subtraction as the differential. If a collision is found on hash function $H(\cdot)$, that is, M, M' such that $H(M) = H(M'), M \neq M'$ is found, the differential values of M and $H(M)$ become $\Delta M = M' - M \neq 0$, $\Delta H(M, M') = H(M') - H(M) = 0$. Let x and x' be certain values. We write $x' - x$ as Δx, and we call Δx the differential value of x. Since the differential value of input message $\Delta M \neq 0$, differential values of the chaining variables of the hash function are not 0.

The method of Wang et al. first notes differential values. It determines the differential values of the chaining variables and the differential value of the input message so that the output differential value of hash function $\Delta H(M, M')$ becomes $\Delta H(M, M') = 0$ and the differential value of the input message becomes $\Delta M \neq 0$. However, even if we find a pair of messages M, M' that satisfy ΔM, the output differential value is not always $H(M') - H(M) = 0$. This can happen since the differentials of chaining values from M and M' do not always satisfy the differential values of the chaining variables. Therefore, we need to set conditions for satisfying the differential values of the chaining variables. We call these conditions "sufficient conditions". These procedures (deciding the differential value of the input message, differential values of the chaining variables and sufficient conditions) are pre-computations.

We start collision search by using the differential value of input message ΔM and sufficient conditions decided in the pre-computation phase. First, we search for message M satisfying all sufficient conditions. Next, we calculate $M' = M + \Delta M$. M and M' thus become collision messages, that is, $H(M) = H(M')$. In order to efficiently locate a message that satisfies all sufficient conditions, message modification can be used.

3.1 Message modification for SHA-0 and MD-family

First, we explain message modification for SHA-0, and clarify the range wherein message modification can be applied. Next, since we use the idea of cancel method, which is originally proposed for MD-family, as part of the proposed submarine modification, we explain the procedures of cancel method.

Message Modification for SHA-0 [14]

Message modification for SHA-0 can generate messages satisfying all sufficient conditions in steps 1-16 with probability of 1. This procedure is shown below.

- **Message Modification for step** i $(1 \leq i \leq 16)$**:**
  1. Generate a_i satisfying all sufficient conditions for a_i.
  2. Calculate $m_{i-1} \leftarrow a_i - (a_{i-1} \lll 5) - f(b_{i-1}, c_{i-1}, d_{i-1}) - e_{i-1} - k_{i-1}$.

Transmission method was proposed by Wang et al as follows. These modifications are executed when sufficient conditions are checked and found to be not satisfied. In message modification for steps 17-20, differentials are generated in order to create a differential on a bit where the sufficient condition that we want to satisfy exists. From the specification of SHA-0, since we can freely choose messages only for steps 1-16, we input the differential on the message used in up to step 16. We then transfer this differential to step 17, which yields the differentials that impact the targeted bits in the subsequent steps.

Multi-Message Modification for MD-family [8,9]

Multi-message modification for the MD-family (which we call *cancel method*) involves modifying messages to satisfy the sufficient conditions from step 17 of the MD-family. In cancel method, differentials are input in steps which are not affected by message expansion, and then cancel the impact of those differentials. The differentials that are input appear in step 17 and later steps due to message expansion, and this leads to satisfaction of the sufficient conditions. Cancel method does not use the technique where the latent differential transfers.

3.2 Collision Search for SHA-0

Collision search is done to locate a message that satisfies all sufficient conditions; it involves the use of 2 block messages. The sufficient conditions on the first block are set in order to control the differentials of the chaining variables on the second block. Since all conditions are conditions of output values, they cannot be satisfied by message modification. Therefore, we don't execute any message modification when searching for a message that satisfies all sufficient conditions of the first block. Fortunately, since the complexity of message search in the first block (2^{14}SHA-0 operations) is much smaller than that of the second block (2^{39}SHA-0 operations), the complexity of the first block does not impact overall complexity. Collision search on the second block is done by using message modification. Furthermore, the early stopping technique can be used to efficiently find a message that satisfies the sufficient conditions. In the early stopping technique, after step 24 is calculated, the sufficient conditions up to step 24 are checked to determine whether they are satisfied or not. If all conditions are satisfied, steps from 25 are calculated. Otherwise, collision search is repeated from the first procedure. It is important to remember that this method still cannot find a message that is assured of satisfying the sufficient conditions in steps 21-24 with probability of almost 1. Submarine modification, proposed in this paper, can satisfy the sufficient conditions in steps 21-24 with probability of almost 1.

Another problem of the existing method is that it is impossible to execute the algorithm proposed by Wang et al. since their description of it is incomplete. We rectify this omission in Appendix B.

4 New Message Modification Techniques

The method of Wang et al. uses message modification to efficiently locate a collision. Their method can efficiently generate messages that satisfying the sufficient conditions up to step 20. However, Wang et al. did not propose message modification for subsequent steps. This section studies message modification, and proposes message modification so as to satisfy the sufficient conditions in steps 21 to 24. In this paper, we call this modification submarine modification. Since the structure of SHA-0 is very similar to those of the MD-family or SHA-1, submarine modification may also be applicable to those hash functions.

4.1 Main Idea of Submarine Modification

Transmission method can be applied to satisfy sufficient conditions for 5 steps from the start step of transmission[3]. If we use transmission method to satisfy sufficient condition for after step 22, we need to extend the range where the transmission differential can be started after step 17. Therefore, we use the idea of cancel method in order to extend the range where the transmission differential can be started. If we use the latent differential as the transmission differential, we can extend the range where the transmission differential can be started to step 19 followed by the 5 steps. In the case of SHA-0, the maximum number of latent period after step 17 is $t = 3$ [4]. As a result, we can satisfy sufficient conditions

[3] We confirm the number of applicable steps by a computer experiment

[4] By considering a local collision and message expansion, we can find $t = 3$

for up to step 24 by combining ideas of cancel method and transmission method. We use the idea of cancel method to create the latent differential for steps 17-19. Since there is no influence for satisfied sufficient conditions in latent period by using cancel method, we can satisfy sufficient conditions for 5 steps from the start step of transmission by applying transmission method. A brief explanation of submarine modification is shown in Figure 1.

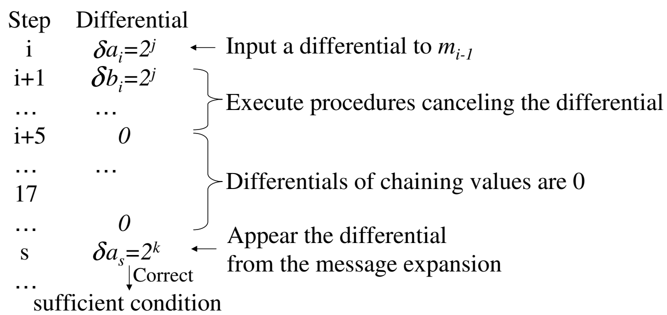

Fig. 1. Outline of Submarine Modification

Remark In this paper, we apply submarine modification to only the case of steps 21-24. However, submarine modification can be also applied to steps 17-20. We want to note that submarine modification is not limited to only the case of steps 21-24.

4.2 How to Construct Submarine Modification

Submarine modification involves inputting and offsetting differentials and transferring differentials. The procedure of submarine modification is as follows:

1. Decide differentials that satisfy a target sufficient condition in step $j (j \geq 17)$ by considering the transfer of differentials.(The idea of transmission method)
2. Decide the method for inputting and offsetting differentials before step 16 to yield the necessary differentials in step j.(The idea of cancel method)

4.3 Proposal of Submarine Modification

There are 4 sufficient conditions from steps 21 to 24: $a_{21,4} = a_{20,4}$ (or $a_{21,4} \neq a_{20,4}$), $a_{22,2} = m_{21,2}, a_{22,4} = a_{21,4}$ (or $a_{22,4} \neq a_{21,4}$), $a_{23,2} = m_{22,2}$. In this section, we propose message modification to satisfy each of these sufficient conditions.

Theorem 1 *Suppose we set following conditions as Extra Conditions.* $a_{6,6} = m_{5,6}, m_{6,11} \neq m_{5,6}, m_{7,6} = m_{5,6}, a_{7,4} = 0, a_{8,4} = 1, m_{10,4} \neq m_{5,6}$. *If we modify* *the message as shown below, the sufficient condition $a_{21,4} = a_{20,4}$ (or $a_{21,4} \neq a_{20,4}$) is satisfied with probability of almost 1.*

$$m_5 \leftarrow m_5 \oplus 2^5, m_6 \leftarrow m_6 \oplus 2^{10}, m_7 \leftarrow m_7 \oplus 2^5, m_{10} \leftarrow m_{10} \oplus 2^3$$

In order to satisfy extra conditions, we generate messages that satisfy these extra conditions in advance by a method similar to that used to satisfy the sufficient conditions.

Proof We explain the change in each chaining variable Theorem 1 is executed in every step.

Step 6. In this step, differential $\delta m_5 = \pm 2^5$ is input. Here, δx is the differential created by message modification on chaining variable x. In this step, a_6 is calculated as follows:

$$a_6 = (a_5 \lll 5) + f(b_5, c_5, d_5) + e_5 + m_5 + k_5.$$

After this equation is calculated, δa_6 becomes $\delta a_6 = \pm 2^5$ because $\delta m_5 = \pm 2^5$. Since $a_{6,6} = m_{5,6}$ is set as the extra condition, $\delta a_6 = \pm 2^5$ does not trigger differential carry. By this condition, since $\delta m_5 = \pm 2^5$ does not cause carry in m_5, and the sign of δa_6 and $\delta m_{5,6}$ are the same, which confirms that no carry occurs.

Step 7. In step 7, a_7 is calculated as follows:

$$a_7 = (a_6 \lll 5) + f(b_6, c_6, d_6) + e_6 + m_6 + k_6.$$

To ensure $\delta a_7 = 0$, we cancel $\delta a_6 = \pm 2^5$ by $\delta m_6 = \pm 2^{10}$. Since $m_{6,11} \neq m_{5,6}$ was set as the extra condition, the sign of $\delta a_6 = \pm 2^5$ and the sign of $\delta m_6 = \pm 2^{10}$ become opposite, and they cancel each other. Due to this condition, in the case of $m_{5,6} = 0$, $m_{6,11}$ becomes $m_{6,11} = 1$. In this situation, $m_{5,6}$ changes from 0 to 1 because of the differential, and $m_{6,11}$ changes from 1 to 0. Since we have ensured that no carry occurs, δm_5 and δm_6 become $\delta m_5 = 2^5$ and $\delta m_6 = -2^{10}$, respectively. Since $\delta m_5 = 2^5$, δa_6 becomes $\delta a_6 = 2^5$. Therefore, $\delta a_7 = 0$ from $\delta a_6 = 2^5 \lll 5 = 2^{10}$ and $\delta m_6 = -2^{10}$. In the case of $m_{5,6} = 0$ and $m_{6,11} = 1$, a similar analysis finds that δa_7 is assured of being 0.

Step 8. In step 8, a_8 is calculated as follows:

$$a_8 = (a_7 \lll 5) + f(b_7, c_7, d_7) + e_7 + m_7 + k_7.$$

To ensure $\delta a_8 = 0$, we cancel $\delta b_7 = \pm 2^5$ by $\delta m_7 = \pm 2^5$. Since $m_{7,6} = m_{5,6}$ was set as the extra condition, $m_{7,6} = 0$ when $m_{5,6} = 0$. In this situation, $m_{5,6}$ changes from 0 to 1, and $m_{7,6}$ changes from 0 to 1. Since we have ensured that no carry occurs, δm_5 and δm_7 become $\delta m_5 = 2^5$ and $\delta m_7 = 2^5$. Since $\delta m_5 = 2^5$, $\delta a_6 = 2^5$, that is, $\delta b_7 = 2^5$, respectively. Since function f is $f(b_7, c_7, d_7) = (b_7 \wedge c_7) \vee (\neg b_7 \wedge d_7)$, and $c_{7,6} = 0, d_{7,8} = 1$ are ensured to be satisfied by the sufficient conditions; the 2nd bit of $f(b_7, c_7, d_7)$ before differential input is 1, and the 2nd bit of $f(b_7, c_7, d_7)$ after differential input is 0. Therefore, $\delta f(b_7, c_7, d_7)$ becomes -2^5 and is canceled by $\delta m_7 = 2^5$. As a result, δa_8 becomes $\delta a_8 = 0$. In the case of $m_{7,6} = 1$ and $m_{5,6} = 1$, a similar analysis confirms that δa_8 is assured of being 0.

Step 9. In step 9, a_9 is calculated as follows:

$$a_9 = (a_8 \lll 5) + f(b_8, c_8, d_8) + e_8 + m_8 + k_8.$$

Since $a_{7,4} = 0$ is set as the extra condition, we can cancel $\delta c_8 = \pm 2^3$ from the property of function f. Since the function f is $f(b_8, c_8, d_8) = (b_8 \wedge c_8) \vee (\neg b_8 \wedge d_8)$, if $b_{8,4} = 0$, the 4-th bit of $f(b_8, c_8, d_8)$ is equal to $d_{8,4}$, and if $b_{8,4} = 1$, the 4-th bit of $f(b_8, c_8, d_8)$ is equal to $c_{8,4}$. Therefore, since $\delta c_8 = \pm 2^3$, $\delta c_8 = \pm 2^3$ is canceled by setting the extra condition $a_{7,4} = 0$, that is, $b_{8,4} = 0$. As a result, δa_9 becomes 0.

Step 10. In step 10, a_{10} is calculated as follows:

$$a_{10} = (a_9 \lll 5) + f(b_9, c_9, d_9) + e_9 + m_9 + k_9.$$

Since $a_{8,4} = 1$ is set as the extra condition, we can cancel $\delta d_9 = \pm 2^3$ from the property of function f. This basically follows Step 9.

Step 11. In step 11, a_{11} is calculated as follows:

$$a_{11} = (a_{10} \lll 5) + f(b_{10}, c_{10}, d_{10}) + e_{10} + m_{10} + k_{10}.$$

To ensure $\delta a_{11} = 0$, we cancel $\delta e_{10} = \pm 2^3$ by $\delta m_{10} = \pm 2^3$. Since $m_{10,4} \neq m_{5,6}$ is set as the extra condition, $m_{10,4}$ becomes $m_{10,4} = 1$ when $m_{5,6} = 0$. In this situation, $m_{5,6}$ changes from 0 to 1, and $m_{10,4}$ changes from 1 to 0. Since we have ensured that no carry is triggered by the differential, δm_5 and δm_{10} become $\delta m_5 = 2^5$ and $\delta m_{10} = -2^3$, respectively. Since $\delta m_5 = 2^5$, δa_6 becomes $\delta a_6 = 2^5$, that is, $\delta e_{10} = 2^3$. Therefore, $\delta e_{10} = 2^3$ is canceled by $\delta m_{10} = -2^3$, and δa_{11} becomes 0. In the case of $m_{5,6} = 1$ and $m_{10,4} = 0$, a similar analysis shows that δa_{11} becomes 0.

From Step 17. Because of input differentials and message expansion, the following message differentials appear from step 19: $\delta m_{18} = \pm 2^3, \delta m_{19} = \pm 2^5$ and $\delta m_{20} = \pm 2^{10}$. $\delta m_{18} = \pm 2^3$ is transferred as shown below, and $a_{21,4} = a_{20,4}$ (or $a_{21,4} \neq a_{20,4}$) is satisfied by $\delta a_{21} = \pm 2^3$.

$$\delta m_{18} = \pm 2^3 \rightarrow \delta a_{19} = \pm 2^3 \rightarrow \delta b_{20} = \pm 2^3 \rightarrow \delta a_{21} = \pm 2^3 \qquad \square$$

Remark We experimentally confirmed that the probability that this message modification can satisfy the target condition without affecting other sufficient conditions is almost 100%. The complexity of this message modification is less than the operations of 2 steps.

Theorem 2 *Suppose we set following conditions as Extra Conditions:* $a_{11,21} = m_{10,21}, m_{11,26} \neq m_{10,21}, a_{10,23} = a_{9,23}, a_{12,19} = 0, a_{13,19} = 1, m_{15,19} \neq m_{10,21}, m_{19,26} \neq$

$m_{18,21}$. *If we modify a message as shown below, the sufficient condition* $a_{22,2} = m_{21,2}$ *is satisfied with probability of almost 1.*

$$m_{10} \leftarrow m_{10} \oplus 2^{20}, m_{11} \leftarrow m_{11} \oplus 2^{25}, m_{15} \leftarrow m_{15} \oplus 2^{18}$$

Proof Since the proof of Theorem 2 is almost the same as the proof of Theorem 1 and due to lack of space, we omit the explanation of this proof.

Remark We experimentally confirmed that the probability that this message modification can satisfy the target condition without affecting the other sufficient conditions is 97.5%. The complexity of this message modification is less than the operations of 3 steps.

Theorem 3 *Suppose we set the following conditions as Extra Conditions:* $a_{11,8} = m_{10,8}, m_{11,13} \neq m_{10,8}, a_{10,10} = a_{9,10}, a_{12,6} = 0, a_{13,6} = 1, m_{15,6} \neq m_{10,8}, m_{19,13} \neq m_{18,8}$. *If we modify the message as shown below, sufficient condition* $a_{22,4} = a_{21,4}$ *(or* $a_{22,4} \neq a_{21,4}$*) is satisfied with probability of almost 1.*

$$m_{10} \leftarrow m_{10} \oplus 2^{7}, m_{11} \leftarrow m_{11} \oplus 2^{12}, m_{15} \leftarrow m_{15} \oplus 2^{5}$$

Proof Since the proof of Theorem 3 is almost same as that of Theorem 1 and due to lack of space, we omit the explanation of this proof.

Remark We experimentally confirmed that the probability that this message modification can satisfy the target condition without affecting the other sufficient conditions is almost 100%. The complexity of this message modification is less than the operations of 3 steps.

Theorem 4 *Suppose we set following conditions as Extra Conditions:* $a_{11,16} = m_{10,16}, m_{11,21} \neq m_{10,16}, m_{12,16} \neq m_{10,16}, a_{12,14} = 0, a_{13,14} = 1, m_{15,14} \neq m_{10,16}, m_{19,21} \neq m_{18,16}$. *If we modify the message as shown below, the sufficient condition* $a_{23,2} = m_{22,2}$ *is satisfied with probability of almost 1.*

$$m_{10} \leftarrow m_{10} \oplus 2^{15}, m_{11} \leftarrow m_{11} \oplus 2^{20}, m_{12} \leftarrow m_{12} \oplus 2^{15}, m_{15} \leftarrow m_{15} \oplus 2^{13}$$

Proof Since the proof of Theorem 4 is almost the same as the proof of Theorem 1 and due to lack of space, we omit the explanation of this proof.

Remark We experimentally confirmed that the probability that this message modification can satisfy the target condition without affecting the other sufficient conditions is 97%. The complexity of this message modification is less than the operations of 4 steps.

4.4 Application to SHA-1

Since a collision attack on SHA-1 [15] is similar to an attack on SHA-0, submarine modification would be applicable to SHA-1. This section considers the application of submarine modification to SHA-1.

Collision search of SHA-1 is done by using message modification as well as collision search of SHA-0. In SHA-1, only message modification for sufficient conditions up to step 22 has been proposed. Therefore, we discuss the possibility of applying submarine modification to realizing the sufficient conditions after step 22 of SHA-1. For example, we discuss message modification to satisfy $a_{23,2} = m_{22,2}$.

Example *Suppose we set following conditions as Extra Conditions:* $a_{11,15} = m_{10,15}, m_{11,20} \neq m_{10,15}, a_{10,17} \neq m_{9,17}, a_{12,13} = 0, a_{13,13} = 1, m_{15,13} \neq m_{10,15}, m_{19,21} \neq m_{18,16}$ *If we modify the message as shown below, the sufficient condition* $a_{23,2} = m_{22,2}$ *is satisfied with probability of almost 1.*

$$m_{10} \leftarrow m_{10} \oplus 2^{14}, m_{11} \leftarrow m_{11} \oplus 2^{19}, m_{15} \leftarrow m_{15} \oplus 2^{12}$$

However, this message modification can impact other sufficient conditions. An analysis of this is a future work.

If we execute this procedure, the following message differentials appear from step 19 due to message expansion: $\delta m_{18} = \pm 2^{13} \pm 2^{15}, \delta m_{19} = \pm 2^{20}, \delta m_{20} = \pm 2^{15}, \delta m_{21} = \pm 2^{14} \pm 2^{16}, \delta m_{22} = \pm 2^{21}$ Since $m_{19,21} \neq m_{18,16}$ is set as the extra condition, we can minimize the probability of breaking the other sufficient conditions. We omit this explanation since it basically follows that of Theorem 2.

$\delta m_{18} = \pm 2^{13}$ is transferred as shown below, and $a_{23,2} = m_{22,2}$ is satisfied by $\delta a_{23} = \pm 2$.

$$\delta m_{18} = \pm 2^{13} \to \delta a_{19} = \pm 2^{13} \to \delta a_{20} = \pm 2^{18} \to \delta a_{21} = \pm 2^{23} \to \delta a_{22} = \pm 2^{28} \to \delta a_{23} = \pm 2$$

Remark Wang et al. announced an improved version of their original attack on SHA-1 [15] at NIST HASH WORKSHOP 2005 and CT-RSA'06 [12,13].

5 Lack of Sufficient Conditions

When we use the sufficient conditions given by Wang et al. [14], a collision attack does not necessarily succeed even if all sufficient conditions are satisfied. This problem occurs because their approach lacks two conditions. Our analysis, detailed below, showed that the missing conditions are $b_{0,9} = 0$ and $b_{0,11} = 1$.

a_3 is calculated as follows:

$$a_3 = (a_2 \lll 5) + f(b_2, c_2, d_2) + e_2 + m_2 + k_2.$$

We transform the above equation for f.

$$f(b_2, c_2, d_2) = a_3 - (a_2 \lll 5) - e_2 - m_2 - k_2$$

Since $\Delta a_3 = 2 - 2^9 - 2^{11} + 2^{16}, \Delta a_2 = -2^4 - 2^6 + 2^{11}, \Delta e_2 = 0$ and $\Delta m_2 = 2 + 2^6 \pm 2^{31}$, $\Delta f(b_2, c_2, d_2)$ is calculated as follows:

$$\begin{aligned}\Delta f(b_2, c_2, d_2) &= \Delta a_3 - (\Delta a_2 \lll 5) - \Delta e_2 - \Delta m_2 \\ &= (2 - 2^9 - 2^{11} + 2^{16}) - ((-2^4 - 2^6 + 2^{11}) \lll 5) - 0 - (2 + 2^6 + 2 \pm 31) \\ &= -2^6 \pm 2^{31}.\end{aligned}$$

Since $\varDelta b_2 = -2+2^6+2^{11}$, $b_{2,2}$ is fixed to change from 1 to 0 due to the differential -2, $b_{2,7}$ is fixed to change from 1 to 0, $b_{2,8}$ is fixed to change from 1 to 0, $b_{2,9}$ is fixed to change from 0 to 1 due to the use of differential 2^6. The sign of the change by differential $\pm 2^{31}$ does not have to be considered since it is MSB. Here, we focus on the 7th and 9th bits.

First, we discuss the 7th bit. Wang et al. takes advantage of the fact that $b_{2,7}$ changes from 1 to 0 in order to make differential -2^6 on $f(b_2, c_2, d_2)$. From the property of $f(b_2, c_2, d_2) = (b_2 \wedge c_2) \vee (\neg b_2 \wedge d_2)$, if we set $c_{2,7} = 1$ and $d_{2,7} = 0$, that is, $a_{0,9} = 1$ and $b_{0,9} = 0$ as sufficient conditions, we can make differential -2^6. However, $b_{0,9} = 0$ was not one of the sufficient conditions described by Wang et al.

We turn now to the 9th bit. $b_{2,9}$ changes from 0 to 1. Wang et al. cancel this influence in function f. From the property of $f(b_2, c_2, d_2) = (b_2 \wedge c_2) \vee (\neg b_2 \wedge d_2)$, if we set $c_{2,9} = d_{2,9}$, that is, $a_{0,11} = b_{0,11}$, we can cancel the influence of the change of $b_{2,9}$. Since $a_{0,11} = 1$ is one of the sufficient conditions given by Wang et al, we need to set $b_{0,11} = 1$ as a sufficient condition. This sufficient condition was not specified by Wang et al.

From the above, we need to use $b_{0,9} = 0$ and $b_{0,11} = 1$ as sufficient conditions in addition to those given by Wang et al.

6 Complexity of Collision Search

Without the additional sufficient conditions the generation of a message that yields a collision will fail with probability $\frac{3}{4}$.

Combining the two additional sufficient conditions with those of Wang et al. and using submarine modification reduces the complexity of collision search to 2^{36} SHA-0 operations. This calculation is given below.

1st block and Step 1-13 of 2nd block. The complexity of generating messages for these steps is insignificant. Refer to the paper written by Wang et al. [14].

Step 14-20 of 2nd block. The complexity of generating messages that satisfy all sufficient conditions in steps 14-20, including message modification, is less than 8 steps.

Step 21 of 2nd block. The complexity of generating messages that satisfy all sufficient conditions up to step 21 including submarine modification is less than,

$$8 + 1 + \frac{1}{2} \cdot 2 = 10.$$

Step 22 of 2nd block. The complexity of generating messages that satisfy all sufficient conditions up to step 22 including submarine modification is calculated as follows: Let the complexity where conditions up to step 22 are satisfied and the number of times m_{14}, m_{15} is chosen is less than i times $x_{22,i}$. In this situation, the following equation below is valid. Here $x_{22,0} = 0$.

$$x_{22,i} = \left(\frac{1}{2} \cdot 0.025\right)^{i-1} \cdot \left(10 + 1 + \frac{1}{2} \cdot 3 + \frac{1}{2} \cdot 3\right) + x_{22,i-1}$$

M_{1block}	f459644c b87cdae1 ed98d4a6 7f5c304b a8606648 073dda8d 9f044c3a 2386c95f 8b611aa4 d66ed3b9 c4854f6e d57662b3 d687ebe0 f61cefe5 6d0252c2 01f298bc
h_{1block}	41f3e784 96831ef3 563e0aa9 d7def7ba 232e8581
M_{2block}	76c21fb3 8a725c5a 13a6039c a23c1950 53e65762 b70bbb88 705ec5b6 079e5dd5 f58793f6 d67d305e 352ee1b8 87c36500 fd012cb5 a51c4269 6a72aabd 7a2449cc
M'_{2block}	f6c21ff1 8a725c5a 93a603de a23c1910 53e65722 b70bbbca f05ec5b4 879e5dd7 f58793b6 567d305e b52ee1f8 07c36502 fd012cb7 251c4229 ea72aabd fa24498c
h_{2block}	cad681a1 354105dc ac31607b 6ccaba44 c76d1948

Table 2. An example of generated collision pair

The complexity is about 15 steps since $\lim_{i \to \infty} x_{22,i} \approx 15$.

Step 23 of 2nd block. The complexity of generating messages that satisfy all sufficient conditions up to step 23 including submarine modification is calculated as follows: Let the complexity where conditions up to step 23 are satisfied and the number of times m_{14}, m_{15} is chosen is less than i times $x_{23,i}$. In this situation, the following equation below is valid. Here $x_{23,0} = 0$.

$$x_{23,i} = \left(\frac{1}{2} \cdot 0.03\right)^{i-1} \cdot \left(15 + 1 + \frac{1}{2} \cdot 4\right) + x_{23,i-1}$$

The complexity is about 18 steps since $\lim_{i \to \infty} x_{23,i} \approx 18$.

Step $i(i = 24 - 80)$ of 2nd block. Let the complexity of generating messages that satisfy all sufficient conditions up to the $i-1$ step be y_{i-1}. If there are n_i sufficient conditions in the i-th step, the probability that all of them are satisfied is 2^{-n_i}. Therefore, y_i, the complexity of generating messages that satisfy all sufficient conditions up to the i-th step, is $y_i = (y_{i-1} + 1) \cdot 2^{n_i}$. From this equation, $y_{80} = 6180766429108$. This is equivalent to 2^{36} SHA-0 operations. From the above consideration, the total complexity of collision search is 2^{36} SHA-0 operations.

Remark There is a possibility the collision attack could be further improved by using another differential path. We discuss this topic in Appendix A.

7 Conclusion

In this paper, we proposed submarine modification, message modification that can satisfy the sufficient conditions in steps 21-24. Moreover, we showed that submarine modification is applicable to SHA-1. We also showed that the sufficient conditions given by Wang et al. are incomplete since they are missing $b_{0,9} = 0$ and $b_{0,11} = 1$. Therefore, even if a message that satisfies all sufficient conditions given by Wang et al. is discovered, a collision generation may fail with probability $\frac{3}{4}$. By utilizing the two additional sufficient conditions and submarine modification, the complexity of collision search is reduced to 2^{36} SHA-0 operations.

Table 2 shows a collision found by using the technique proposed herein. M_{1block} is a message of the 1st block, h_{1block} is the output of the compression function of the 1st block. M_{2block} is a message for the 2nd block, M'_{2block} is a message of 2nd block after the differential is input, h_{2block} is the output of the compression function of 2nd block.

Acknowledgement We would like to thank The Telecommunications Advancement Foundation for supporting our research.

References

1. Eli Biham and Rafi Chen. *Near Collisions of SHA-0*. CRYPTO'04, LNCS 3152, pp290–305, Springer-Verlag, 2004.
2. Eli Biham, Rafi Chen, A. Joux, P. Carribault, W. Jalby and C. Lemuet. *Collisions in SHA-0 and Reduced SHA-1*. EUROCRYPT'05, LNCS 3494, pp36–57, Springer-Verlag, 2005.
3. Florent Chabaud and Antoine Joux. *Differential Collisions in SHA-0*. CRYPTO'98, LNCS 1462, pp56–71, Springer-Verlag, 1998.
4. Antoine Joux. *Collision for SHA-0*. Runm session of CRYPTO2004, August 2004.
5. NIST. *Secure hash standard*. Federal Information Processing Stacdard, FIPS-180, May 1993.
6. NIST. *Secure hash standard*. Federal Information Processing Stacdard, FIPS-180-1, April 1995.
7. Xiaoyun Wang, Dengguo Feng, Hui Chen, Xuejia Lai and Xiuyuan Yu. *Collision for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*. In Rump Session of CRYPTO'04 and Cryptology ePrint Archive, Report 2004/199.
8. Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen and Xiuyuan Yu. *Cryptanalysis of the Hash Functions MD4 and RIPEMD*. EUROCRYPT'05, LNCS 3494, pp1–18, Springer-Verlag, 2005.
9. Xiaoyun Wang and Hongbo Yu. *How to Break MD5 and Other Hash Functions*. EUROCRYPT'05, LNCS 3494, pp19–35, Springer-Verlag, 2005.
10. Xiaoyun Wang. *The Collision Attack on SHA-0*. In Chinese, to appear on www.infosec.edu.cn, 1997.
11. Xiaoyun Wang. *The Improved Collision Attack on SHA-0*. In Chinese, to appear on www.infosec.edu.cn, 1998.
12. Xiaoyun Wang, Andrew C Yao, and Frances Yao. *Cryptanalysis on SHA-1 Hash Function* . Keynote Speech at CRYPTOGRAPHIC HASH WORKSHOP.
13. Xiaoyun Wang. *Cryptanalysis of Hash functions and Potential Dangers*. Invited Talk at CT-RSA 2006.
14. Xiaoyun Wang, Hongbo Yu and Yiqun Lisa Yin. *Efficient Collision Search Attack on SHA-0*. CRYPTO'05, LNCS 3621, pp1–16, Springer-Verlag, 2005.
15. Xiaoyun Wang, Yiqun Lisa Yin and Hongbo Yu. *Finding Collisions in the Full SHA-1*. CRYPTO'05, LNCS 3621, pp17–36, Springer-Verlag, 2005.

A A Study of Other Disturbance Vectors

Wang et al. chose a disturbance vector under the condition that the sufficient conditions up to step 20 can be satisfied by message modification. Therefore, they chose a disturbance vector to minimize the number of sufficient conditions after step 20. However, submarine modification can satisfy sufficient conditions up to step 24 can be satisfied by message modification. Therefore, we expect that if we choose a disturbance vector to minimize the number of sufficient conditions after step 24, we can generate a collision with complexity under 2^{36} SHA-0 operations. If we use the disturbance vector chosen by Wang et al, the number of conditions after step 24 is 38. However, by using the disturbance vector shown in Table 11, the number of conditions after step 24 is 37. Therefore, we expect that the disturbance vector shown in Table 11 enables us to generate a collision with complexity under 2^{36} SHA-0 operations. Additional analysis on this matter is a future task.

i	value
$-5, ..., -1$	0 1 1 1 0
$0, ..., 19$	0 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 1
$20, ..., 39$	0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 0 1 0 1 0
$40, ..., 59$	0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0
$60, ..., 79$	0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0

Table 3. A Disturbance Vector for Reduced Complexity

B Complement of Collision Search by Wang et al.

B.1 2nd Bit and 7th Bit of Messages

The complexity claims of Wang et al. claim address only the sufficient conditions of chaining variables. They don't consider the complexity of satisfying the sufficient conditions of messages. However, when a random message is generated, it must satisfy the sufficient conditions of messages, and this takes a few steps. This raises the complexity of collision search. This increase can be suppressed by fixing the 2nd bit and 7th bit of the messages in advance in order to ensure satisfaction of the sufficient conditions.

B.2 Sufficient Conditions Given by Wang et al.

The sufficient conditions of Wang et al. include those for $a_{13,4}, a_{14,4}, a_{15,4}, a_{16,4}, a_{17,2}$. These values depend on the method used to fix the 2nd and 7th bits of the messages (Discussed in Appendix B.1). That is, if a fixing method different from that of Wang et al. is chosen, the sufficient conditions for $a_{13,4}, a_{14,4}, a_{15,4}, a_{16,4}, a_{17,2}$ are also changed.