Complexity of the Collision and Near-Collision Attack on SHA-0 with Different Message Schedules

Mitsuhiro HATTORI, Shoichi HIROSE, and Susumu YOSHIDA

Graduate School of Informatics, Kyoto University, Kyoto, 606–8501 JAPAN
hattori@hanase.kuee.kyoto-u.ac.jp

Abstract. SHA-0 employs a primitive polynomial of degree 16 over GF(2) in its message schedule. There are 2048 primitive polynomials of degree 16 over GF(2). For each primitive polynomial, a SHA-0 variant can be constructed. In this paper, the security of 2048 variants is analyzed against the Chabaud-Joux attack proposed in CRYPTO'98. The analysis shows that all the variants could be collision-attacked by using near-collisions as a tool and thus the replacement of the primitive polynomial is not a proper way to make SHA-0 secure. However, it is shown that the selection of the variants highly affects the complexity of the attack. Furthermore, a collision in the most vulnerable variant is presented. It is obtained by the original Chabaud-Joux attack without any improvements.

1 Introduction

The cryptographic hash function SHA-0 was issued by NIST (National Institute of Standards and Technology) in 1993[1]. Its structure is mainly based on MD4[2], but there are some differences such as a message schedule. In 1995, an addendum was made on SHA-0 and the new version was called SHA-1[3]. SHA-1 is widely used as a cryptographic tool while SHA-0 is rarely used. However, it is still important and meaningful to invent attacks on SHA-0 because it can be a clue about finding effective attacks on SHA-1 and making clear how to design secure cryptographic hash functions.

A differential attack was proposed by Chabaud and Joux in 1998[4]. Using this attack, they found a collision in the reduced version of SHA-0. They also claimed that a collision could be found with complexity 2^{61} by the attack.

Recently a few significant results of collision attacks have been reported[5, 6, 7]. In 2004, Biham and Chen proposed a method to reduce the complexity of the Chabaud-Joux attack using "neutral bits"[5]. They reduced the complexity to 2^{56}. They also showed that near-collisions can be used as a tool for find collisions. They found near-collisions of 18-bit differences with complexity 2^{40}. Wang et.al. found a collision attack on SHA-0 whose complexity is about 2^{40}[6]. However, the details of the attack are not reported. Collisions in SHA-0 have finally been found in [7].

In spite of the recent success in the attacks on SHA-0, SHA-1 seems still secure. The only addendum to SHA-1 is the one-bit rotate shift operation in the message schedule. It implies that the message schedule is very critical in the design of cryptographic hash functions.

The message schedule process in SHA-0 expands 16 words $\{W_0, \ldots, W_{15}\}$ to 80 words $\{W_0, \ldots, W_{79}\}$ using the equation

$$W_i = W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}, \quad i = 16, \ldots, 79 \ , \tag{1}$$

where $\oplus$ is bitwise exclusive OR. (1) is derived from a primitive polynomial of 16 degrees over GF(2)[8], that is

$$x^{16} + x^{13} + x^8 + x^2 + 1 \ . \tag{2}$$

There are 2048 primitive polynomials of 16 degrees and (2) is one of them. It has not been made public why this polynomial is adopted and it is still unclear how appropriate the selection is. We analyze it in this paper.

We deal with 2048 SHA-0 variants each of which employs one of 2048 primitive polynomials. We estimate their collision and near-collision resistance to the Chabaud-Joux attack. We do not consider the complexity reduction by the Biham-Chen neutral bit. This is because it seems difficult to estimate the complexity of their attack. The complexity depends on the number of neutral bits obtained experimentally.

First we estimate collision resistance of SHA-0 variants. We find 412 variants are totally resistant to the Chabaud-Joux attack and 817 variants are more resistant than the original SHA-0. Notice that the "neutral bits" technique cannot be applied to the totally resistant 412 variants. We also find a collision in the most vulnerable variant using the original Chabaud-Joux attack without any improvements. The complexity is estimated at about 2^{45}.

Second, we estimate near-collision resistance of the SHA-0 variants. We find no variants are totally resistant to the Chabaud-Joux near-collision attack. Therefore all the variants could be collision-attacked by using such near-collisions as a tool. The complexity of the most resistant one is 2^{69}, while that of the most vulnerable one is 2^{35}. It is concluded that the replacement of the primitive polynomial is not a proper way to make SHA-0 secure.

This paper is organized as follows: Section 2 describes the algorithm of SHA-0. Section 3 describes the Chabaud-Joux attack and Section 4 describes our analysis. Section 5 concludes the paper.

2 Description of SHA-0

This section describes the algorithm of SHA-0. Prior to the description, we define a few notations. $\oplus$ is the bitwise addition $\bmod 2$ and $\boxplus$ is the addition $\bmod 2^{32}$. $\mathrm{ROL}_l(W)$ is the l-bit left rotate shift operation of the 32-bit word W. Numbers with a prefix "$0x$" are represented in hexadecimal.

2.1 Algorithm of SHA-0

The hash value of a message M is computed as follows[3, 9].

1. Append a single bit '1' to the end of M, followed by k-bits '0'. k is the smallest non-negative solution to the equation $|M|+1+k = 448 \bmod 512$, where $|M|$ is the bit length of M. Then append the 64-bit binary representation of $|M|$. The length of the padded message is a multiple of 512. Divide them into 512-bit blocks $M_1, \ldots, M_n$.
2. Initialize a 5-word buffer H_0 to
$$H_0 = (0x67452301, 0x\mathrm{efcdab89}, 0x\mathrm{98badcfe}, 0x10325476, 0x\mathrm{c3d2e1f0}) \ .$$
3. For each M_j, proceed a compression function as follows:
$$H_j = \mathrm{compress}(M_j, H_{j-1}) \ .$$
The compression function is described momentarily.
4. Output H_n. This is the hash value of M.

2.2 Algorithm of the Compression Function

The compression function $\mathrm{compress}(M_j, H_{j-1})$ proceeds the following operations.

1. Initialize five registers A, B, C, D, and E by
$$(A_0, B_0, C_0, D_0, E_0) = H_{j-1} \ .$$
2. Divide the 512-bit message block M_j to 16 words $W_0, \ldots, W_{15}$. The length of a word is 32 bits.
3. Expand the 16 words to 80 words by
$$W_i = W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}, \quad i = 16, \ldots, 79 \ . \tag{3}$$
We call this operation the message schedule.
4. For $i = 0$ to $i = 79$, iterate the following operations.
$$\begin{aligned} A_{i+1} &= \mathrm{ROL}_5(A_i) \boxplus f_i(B_i, C_i, D_i) \boxplus E_i \boxplus K_i \boxplus W_i \\ B_{i+1} &= A_i \\ C_{i+1} &= \mathrm{ROL}_{30}(B_i) \\ D_{i+1} &= C_i \\ E_{i+1} &= D_i \ , \end{aligned}$$
where f_i and K_i are defined in Table 1.
5. Output $H_j = (A_0 \boxplus A_{80}, B_0 \boxplus B_{80}, C_0 \boxplus C_{80}, D_0 \boxplus D_{80}, E_0 \boxplus E_{80})$.

Table 1: Definition of f_i and constant K_i

Round i	$f_i(B, C, D)$	K_i
0, ... ,19	$(B \wedge C) \vee (\neg B \wedge D)$	$0x$5a827999
20, ... ,39	$B \oplus C \oplus D$	$0x$6ed9eba1
40, ... ,59	$(B \wedge C) \vee (C \wedge D) \vee (D \wedge B)$	$0x$8f1bbcdc
60, ... ,79	$B \oplus C \oplus D$	$0x$ca62c1d6

3 The Chabaud-Joux Attack

In this section we describe how the Chabaud-Joux attack[4] is intended to work.

The main idea of their attack is to flip several bits in a message and adjust the influence caused by the inversion within 5 succeeding rounds. Let $M = \{W_0, \ldots, W_{79}\}$ and $M' = \{W'_0, \ldots, W'_{79}\}$ be a pair of colliding messages. Suppose that $W_i \oplus W'_i = 0x00000002$. This implies that the bit 1 of W_i is inverted (note that the LSB is bit 0). Then a difference is made in A_{i+1} and A'_{i+1}. If no adjustment is made, the difference triggers other differences in A_{i+2} and A'_{i+2}, A_{i+3} and A'_{i+3}, and so on. These differences also trigger many differences. To prevent this, adjustment is made on W'_{i+1}, W'_{i+2}, W'_{i+3}, W'_{i+4}, and W'_{i+5}. Let $W'_{i+1} = W_{i+1} \oplus 0x00000040$. Then $A_{i+2} = A'_{i+2}$ with some probability. Let $W'_{i+2} = W_{i+2} \oplus 0x00000002$. Then $A_{i+3} = A'_{i+3}$ with some probability. Let $W'_{i+k} = W_{i+k} \oplus 0x80000000$ $(k = 3, 4, 5)$. Then $A_{i+k+1} = A'_{i+k+1}$ with some probability.

In order to maximize the success probability of the adjustment, the first inversions are always on bit 1 of a message. The 80-bit sequence of the inversion bits is called a *mask base* m. Let m_i be the i-th bit of m. Since m is a part of a message, m must follow the following equation:

$$m_i = m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}, \quad i = 11, \ldots, 79 \ , \tag{4}$$

where i starts at 11 because m_{11} is used for calculating the adjustment of W'_{16}, and W'_{16} must follow the equation (3).

A constraint is needed on m. $m_{75}, \ldots, m_{79}$ must be 0 because such inversions cannot be adjusted in the left few rounds. Since each m_i can be represented by a linear combination of $m_0, \ldots, m_{10}$, the constraint on $m_{75}, \ldots, m_{79}$ is also represented by a linear combination of $m_0, \ldots, m_{10}$. In case of the original SHA-0, the constraint is as follows:

$$\begin{aligned}
m_{75} &= m_0 \oplus m_1 \oplus m_4 \oplus m_5 \oplus m_8 \oplus m_9 = 0 \\
m_{76} &= m_1 \oplus m_2 \oplus m_5 \oplus m_6 \oplus m_9 \oplus m_{10} = 0 \\
m_{77} &= m_2 \oplus m_6 \oplus m_7 \oplus m_8 \oplus m_{10} = 0 \\
m_{78} &= m_0 \oplus m_7 \oplus m_9 = 0 \\
m_{79} &= m_1 \oplus m_8 \oplus m_{10} = 0 \ .
\end{aligned}$$

Therefore the number of possible mask bases is $2^6 = 64$. Since $m = 000\ldots0$ is not an appropriate mask base, the number of mask base candidates is 63. For each mask base, the probability of succeeding in finding a collision is calculated, and the most effective mask base is selected for the attack.

Here we show an example of calculation of success probability for

$$m = \texttt{0001 0000 0001 0010 0000 0010 0001 1011 0111 1110}$$
$$\texttt{1101 0010 0001 0101 0010 1010 0010 1110 0110 0000} \;.$$

The calculation is performed as shown in Table 2. Let us take an example of round 3 in Table 2. The probability that A_4 and A'_4 differs only in bit 1 is 1/2. The probabilities that the adjustments in f_5, f_6 and f_7 succeed are $1/2^2$, 1/2 and 1/2, respectively. Therefore the probability that the flip on round 3 is appropriately adjusted is $1/2^5$. For each 1 in the mask base, the probability of the success in adjustment can be evaluated in the similar way. Refer to [4] for details.

4 The Attack on SHA-0 Variants

The message schedule formula (3) is derived from the 16-degree primitive polynomial (2) over GF(2). The number of 16-degree primitive polynomials over GF(2) is 2048. It has not been made public why the polynomial (2) is adopted among these primitive polynomials, and it is still unclear how appropriate the selection is.

We shed light on this question. For each primitive polynomial, a SHA-0 variant is constructed whose message schedule is based on it. Thus, we have 2048 SHA-0 variants (including the original one). We evaluate their resistance to the Chabaud-Joux differential attack. First we evaluate their resistance to the collision attack. Then we evaluate their resistance to the near-collision attack.

4.1 The Collision Attack

In this section we evaluate the resistance of SHA-0 variants to the Chabaud-Joux collision attack.

Method. For each primitive polynomial, the complexity of the attack is calculated as follows.

1. Derive the recurrence formulas on W_i and m_i.
2. Calculate all the mask base candidates, taking a constraint on $m_{75}, \ldots, m_{79}$ into consideration.
3. For each mask base candidate, compute the maximum probability of succeeding in the attack.
4. Select a mask base which has the maximum success probability and compute the computational complexity of the attack.

Table 2: Success probability for mask base m. "No difference in propagation" is the probability that the disturbance in W'_i does not cause the difference in carry propagation between the computation of A'_{i+1} and that of A_{i+1}. The total probability is $1/2^{82}$, but the first $1/2^{14}$ is ignored because it is the probability of succeeding in the initiation of the attack.

Round i	No difference in propagation	f_{i+2}	f_{i+3}	f_{i+4}	Overall probability
3	1/2	$1/2^2$	1/2	1/2	$1/2^5$
11	1/2	$1/2^2$	1/2	1/2	$1/2^5$
14	1/2	$1/2^2$	1/2	1/2	$1/2^4 \times 1/2$
22	1/2	1/2	1	1	$1/2^2$
27	1/2	1/2	1	1	$1/2^2$
28	1/2	1/2	1	1	$1/2^2$
30	1/2	1/2	1	1	$1/2^2$
31	1/2	1/2	1	1	$1/2^2$
33	1/2	1/2	1	1	$1/2^2$
34	1/2	1/2	1	1	$1/2^2$
35	1/2	1/2	1	1	$1/2^2$
36	1/2	1/2	1	1	$1/2^2$
37	1/2	1/2	1	1	$1/2^2$
38	1/2	1/2	1	1/2	$1/2^3$
40	1/2	1/2	1/2	1	$1/2^3$
41	1/2	1/2	1	1/2	$1/2^3$
43	1/2	1/2	1/2	1/2	$1/2^4$
46	1/2	1/2	1/2	1/2	$1/2^4$
51	1/2	1/2	1/2	1/2	$1/2^4$
53	1/2	1/2	1/2	1/2	$1/2^4$
55	1/2	1/2	1/2	1/2	$1/2^4$
58	1/2	1/2	1	1	$1/2^2$
60	1/2	1/2	1	1	$1/2^2$
62	1/2	1/2	1	1	$1/2^2$
66	1/2	1/2	1	1	$1/2^2$
68	1/2	1/2	1	1	$1/2^2$
69	1/2	1/2	1	1	$1/2^2$
70	1/2	1/2	1	1	$1/2^2$
73	1/2	1/2	1	1	$1/2^2$
74	1/2	1/2	1	1	$1/2^2$

Here we show an example of applying the method to $x^{16}+x^{12}+x^{11}+x^7+x^4+x+1$.

1. Based on the polynomial, the recurrence formulas on W_i and m_i are

$$W_i = W_{i-4} \oplus W_{i-5} \oplus W_{i-9} \oplus W_{i-12} \oplus W_{i-15} \oplus W_{i-16},\ i = 16, \ldots, 79 \quad (5)$$
$$m_i = m_{i-4} \oplus m_{i-5} \oplus m_{i-9} \oplus m_{i-12} \oplus m_{i-15} \oplus m_{i-16},\ i = 11, \ldots, 79. \quad (6)$$

2. For this example, m_{79} is always 0. Constraints are

$$\begin{aligned} m_7 &= m_2 \oplus m_3 \\ m_8 &= m_3 \oplus m_4 \\ m_9 &= m_0 \oplus m_4 \oplus m_5 \\ m_{10} &= m_1 \oplus m_5 \oplus m_6 \ . \end{aligned}$$

   The number of mask base candidates is $2^7 - 1 = 127$.
3. For all the 127 mask base candidates, compute the probability of succeeding in the attack. The result is in Table 3.
4. The maximum probability is $1/2^{52}$ and the complexity for the attack is about 2^{52}.

As shown in this example, the number of mask base candidates differs according to polynomials.

Results. We calculated the computational complexity of the attack for all the 2048 primitive polynomials. The result is shown in Table 4. The complexity of the attack on the original SHA-0 is 2^{68}. Actually, in [4], it is stated that the complexity is reduced to 2^{61} for the original SHA-0 by some improvements. However, the result in Table 4 shows computational complexity of the attack without any improvements for each SHA-0 variant.

The maximum complexity represented by 2^{∞} in Table 4 means that no mask base is usable for the attack. Thus, 412 variants are totally resistant. In order for the attack to succeed, a mask base must not have any m_i $(0 \le i \le 15)$ such that $m_i = m_{i+1} = 1$. If it has, a disturbance caused by m_i would never be corrected and a collision would never happen.

The number of variants which are not totally resistant but more resistant to the attack than the original one is 817, and the number of variants which are as resistant as or more vulnerable to the attack is 818. The original one is therefore weaker one among 2048 variants.

The most vulnerable variant employs the polynomial $x^{16}+x^{14}+x^{12}+x^{10}+x^8+x^7+x^6+x^5+x^4+x^2+1$. The complexity is 2^{45}. We found a collision of this variant after two days of computation on 16 PCs;

```
af3bd4fd ada5ead4 5d1da03d ccf3db2e
b3887606 5e068898 40c134dc 263294bf
cbf91ff0 1b997ed2 92118e2f fae0fd89
846f3d48 a53fafd5 cb3a1deb 5f46b7b0
```

Table 3: Probability of success in the attack using each mask base on $x^{16}+x^{12}+x^{11}+x^7+x^4+x+1$. A mask base consists of 11 bits and it is represented as $m_0 \ldots m_{10}$. Probability 0 means the attack cannot work at all.

Mask base	Probability
00000010001	0
00000100011	0
00000110010	0
00001000110	0
00001010111	0
00001100101	0
00001110100	0
00010001100	0
00010011101	0
00010101111	0
00010111110	0
00011001010	0
00011011011	0
00011101001	0
00011111000	0
00100001000	$1/2^{78}$
00100011001	0
00100101011	0
00100111010	0
00101001110	0
00101011111	0
00101101101	0
00101111100	0
00110000100	0
00110010101	0
00110100111	0
00110110110	0
00111000010	0
00111010011	0
00111100001	0
00111110000	0
01000000001	0
01000010000	$1/2^{78}$
01000100010	$1/2^{77}$
01000110011	0
01001000111	0
01001010110	0
01001100100	0
01001110101	0
01010001101	0

Mask base	Probability
01010011100	0
01010101110	0
01010111111	0
01011001011	0
01011011010	0
01011101000	0
01011111001	0
01100001001	0
01100011000	0
01100101010	0
01100111011	0
01101001111	0
01101011110	0
01101101100	0
01101111101	0
01110000101	0
01110010100	0
01110100110	0
01110110111	0
01111000011	0
01111010010	0
01111100000	0
01111110001	0
10000000010	0
10000010011	0
10000100001	$1/2^{78}$
10000110000	0
10001000100	$1/2^{79}$
10001010101	0
10001100111	0
10001110110	0
10010001110	0
10010011111	0
10010101101	0
10010111100	0
10011001000	0
10011011001	0
10011101011	0
10011111010	0
10100001010	$1/2^{52}$

Table 3: Continued

Mask base	Probability
10100011011	0
10100101001	$1/2^{68}$
10100111000	0
10101001100	0
10101011101	0
10101101111	0
10101111110	0
10110000110	0
10110010111	0
10110100101	0
10110110100	0
10111000000	0
10111010001	0
10111100011	0
10111110010	0
11000000011	0
11000010010	0
11000100000	0
11000110001	0
11001000101	0
11001010100	0
11001100110	0
11001110111	0
11010001111	0

Mask base	Probability
11010011110	0
11010101100	0
11010111101	0
11011001001	0
11011011000	0
11011101010	0
11011111011	0
11100001011	0
11100011010	0
11100101000	0
11100111001	0
11101001101	0
11101011100	0
11101101110	0
11101111111	0
11110000111	0
11110010110	0
11110100100	0
11110110101	0
11111000001	0
11111010000	0
11111100010	0
11111110011	0

and

```
af3bd4fd ada5ead4 5d1da03d ccf3db2e
b3887606 5e06889a 40c1349c 263294bd
4bf91ff0 9b997ed0 12118e6f fae0fd8b
046f3d48 253fafd5 4b3a1de9 5f46b7f0
```

give the same hash value

```
e930e9be 464527bf e489b780 a3314c11 fcb6ea88 .
```

The minimum number of terms of primitive polynomials is 5. We call such polynomials *5-term polynomials*, and we call variants employing 5-term polynomials *5-term variants*. The original SHA-0 is a 5-term variant. The fewer the number of terms is, the faster the message schedule proceeds. The result of 5-term variants is extracted in Table 5. There are 10 5-term variants which are totally resistant to the attack. For example, $x^{16} + x^{14} + x^9 + x^4 + 1$ is totally resistant. One of the most vulnerable 5-term variants is with $x^{16}+x^9+x^4+x^2+1$ and the complexity is about 2^{55}.

Table 4: Computational complexity needed for the collision attack on SHA-0 variants

Complexity	Number of variants	
		Summation
2^{∞}	412	412
2^{101}	1	413
2^{100}	0	413
2^{99}	0	413
2^{98}	0	413
2^{97}	0	413
2^{96}	0	413
2^{95}	0	413
2^{94}	0	413
2^{93}	1	414
2^{92}	1	415
2^{91}	1	416
2^{90}	3	419
2^{89}	4	423
2^{88}	7	430
2^{87}	7	437
2^{86}	9	446
2^{85}	10	456
2^{84}	12	468
2^{83}	12	480
2^{82}	20	500
2^{81}	22	522
2^{80}	29	551
2^{79}	34	585
2^{78}	36	621
2^{77}	47	668
2^{76}	44	712
2^{75}	70	782
2^{74}	57	839

Complexity	Number of variants	
		Summation
2^{73}	65	904
2^{72}	78	982
2^{71}	90	1072
2^{70}	79	1151
2^{69}	79	1230
2^{68}	84	1314
2^{67}	81	1395
2^{66}	79	1474
2^{65}	80	1554
2^{64}	68	1622
2^{63}	73	1695
2^{62}	59	1754
2^{61}	52	1806
2^{60}	49	1855
2^{59}	41	1896
2^{58}	31	1927
2^{57}	27	1954
2^{56}	24	1978
2^{55}	16	1994
2^{54}	11	2005
2^{53}	11	2016
2^{52}	11	2027
2^{51}	5	2032
2^{50}	3	2035
2^{49}	5	2040
2^{48}	5	2045
2^{47}	1	2046
2^{46}	1	2047
2^{45}	1	2048

4.2 The Near-Collision Attack

In this section we evaluate the resistance of SHA-0 variants to the Chabaud-Joux near-collision attack.

The Near-Collisions in This Paper. In this paper, the restricted near-collisions are treated. This is because we are concerned with the near-collisions which can be used as a tool for finding collisions.

Biham and Chen stated in their paper[5] that near-collisions can be used as a tool in order to find collisions, but the details have been unpublished. We examined them and found that only such near-collisions can be used for finding collisions as the differences exist only on bit 1 of A_{80} and B_{80} and bit 31 of C_{80}, D_{80} and E_{80}. These differences can be adjusted within several steps at the

Table 5: Computational complexity needed for the collision attack on 5-term variants. "Polynomials" denotes the powers of three terms except for x^{16} and 1. x^{16} and 1 are omitted because each polynomial includes them. As an example, "fc1" stands for $x^{16} + x^{15} + x^{12} + x + 1$.

Complexity	Number of variants		Polynomials (in hexadecimal)
		Summation	
2^{∞}	10	10	fc1, fa4, f42, f41, ed5, e94, dc7, b97, a73, 943
2^{83}	1	11	974
2^{82}	1	12	c72
2^{81}	0	12	
2^{80}	1	13	ba5
2^{79}	0	13	
2^{78}	0	13	
2^{77}	4	17	e83, b98, b32, 952
2^{76}	0	17	
2^{75}	0	17	
2^{74}	0	17	
2^{73}	2	19	a76, 972
2^{72}	1	20	a74
2^{71}	1	21	eb7
2^{70}	5	26	fd4, f97, f96, b65, a53
2^{69}	2	28	f72, c96
2^{68}	5	33	f94, e91, dcb, d82, a71
2^{67}	3	36	c97, c61, 975
2^{66}	1	37	875
2^{65}	1	38	c71
2^{64}	3	41	ec7, c31, 532
2^{63}	4	45	fca, ec1, ca3, a96
2^{62}	1	46	543
2^{61}	1	47	db6
2^{60}	0	47	
2^{59}	1	48	d96
2^{58}	1	49	d64
2^{57}	0	49	
2^{56}	1	50	edb
2^{55}	2	52	942, 641

next compression function in the scheme of the Chabaud-Joux and Biham-Chen attack. Our consideration is supported by the fact that the one-bit difference of the near-collision shown as an example in [10] is on bit 1 of B_{80}.

We estimate the complexity of finding such near-collisions in the scheme of the Chabaud-Joux attack. Since the Biham-Chen attack is an improved version of the Chabaud-Joux attack, the complexity will be reduced by the Biham-Chen attack instead.

Method. The complexity of the near-collision attack is calculated in the same way as that of the collision attack. In the near-collision attack, the constraint on a mask m is cancelled that is $m_{75}, \ldots, m_{79}$ must be 0. If $m_i (i = 75, \ldots, 79)$ is 1, the adjustment of the influences of the flip will not be completed, which results in one-bit difference in one of the 5-bits we mentioned. Thus this near-collision can be used for the collision attack.

Results. We calculated the computational complexity of the near-collision attack for all the 2048 SHA-0 variants. The result is shown in Table 6. In the collision attack there are 412 totally resistant variants. In the near-collision attack, however, all variants can be attacked by the Chabaud-Joux attack. Therefore all variants could be collision-attacked by using the near-collisions as a tool. The complexity of the near-collision attack on the original SHA-0 is 2^{51}. The original one is therefore weaker one among 2048 variants.

The most resistant variant whose complexity is 2^{69} seems secure, but the complexity may be reduced by using the Biham-Chen attack. Therefore, all the variants do not seem acceptable for a replacement of SHA-0.

5 Conclusion

In this paper we evaluated the complexity of the Chabaud-Joux attack on SHA-0 variants which employ 16-degree primitive polynomials over GF(2). We showed that all the variants could be collision-attacked by using near-collisions as a tool. It is revealed that the value of the complexity of the collision attack is distributed from 2^{45} to 2^{∞}, and that of the near-collision attack is distributed from 2^{35} to 2^{69}. We also showed that the most vulnerable SHA-0 variant can be collision-attacked within a few days. These results imply that the message schedule should be designed carefully because it affects heavily the security of customized hash functions.

References

[1] National Institute of Standards and Technology, Secure hash standard, FIPS Publication, no.180, 1993.

Table 6: Computational complexity needed for the near-collision attack on SHA-0 variants

Complexity	Number of variants	Summation
2^{69}	3	3
2^{68}	4	7
2^{67}	8	15
2^{66}	35	50
2^{65}	55	105
2^{64}	71	176
2^{63}	121	297
2^{62}	167	464
2^{61}	192	656
2^{60}	188	844
2^{59}	198	1042
2^{58}	189	1231
2^{57}	189	1420
2^{56}	132	1552
2^{55}	125	1677
2^{54}	87	1764
2^{53}	84	1848
2^{52}	57	1905
2^{51}	38	1943
2^{50}	33	1976
2^{49}	22	1998
2^{48}	20	2018
2^{47}	11	2029
2^{46}	5	2034
2^{45}	5	2039
2^{44}	5	2044
2^{43}	1	2045
2^{42}	1	2046
2^{41}	0	2046
2^{40}	0	2046
2^{39}	0	2046
2^{38}	0	2046
2^{37}	1	2047
2^{36}	0	2047
2^{35}	1	2048

[2] R. Rivest, "The MD4 message digest algorithm," CRYPTO'90, Lecture Notes in Computer Science, vol. 537, pp.303–311, 1991.

[3] National Institute of Standards and Technology, Secure hash standard, FIPS Publication, no.180-1, 1995.

[4] F. Chabaud and A. Joux, "Differential collisions in SHA-0," Crypto'98, Lecture Notes in Computer Science, vol. 1462, pp.56–71, 1998.

[5] E. Biham and R. Chen, "Near-collisions of SHA-0," Cryptology ePrint Archive, Report 2004/146, 2004. `http://eprint.iacr.org/2004/146`

[6] X. Wang, D. Feng, X. Lai, and H. Yu, "Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD," Cryptology ePrint Archive, Report 2004/199, 2004. `http://eprint.iacr.org/2004/199`

[7] A. Joux, "Collisions in SHA-0," Short talk presented at CRYPTO 2004 Rump Session, 2004.

[8] W.W. Peterson, Error-correcting codes, M.I.T. Press, 1961.

[9] National Institute of Standards and Technology, Secure hash standard, FIPS Publication, no.180-2, 2002.

[10] E. Biham and R. Chen, "New results on SHA-0 and SHA-1," Short talk presented at CRYPTO 2004 Rump Session, 2004.