Gröbner Basis Based Cryptanalysis of SHA-1

Makoto Sugita * Mitsuru Kawazoe † Hideki Imai ‡

Abstract— Recently, Wang proposed a new method to cryptanalyze SHA-1 and found collisions of 58-round SHA-1. However many details of Wang's attack are still unpublished, especially, 1) How to find differential paths? 2) How to modify messages properly? For the first issue, some results have already been reported. In our article, we clarify the second issue and give a sophisticated method based on Gröbner basis techniques. We propose two algorithm based on the basic and an improved message modification techniques respectively. The complexity of our algorithm to find a collision for 58-round SHA-1 based on the basic message modification is 2^{29} message modifications and its implementation is equivalent to 2^{31} SHA-1 computation experimentally, whereas Wang's method needs 2^{34} SHA-1 computation. The proposed improved message modification is applied to construct a more sophisticated algorithm to find a collision. The complexity to find a collision for 58-round SHA-1 based on this improved message modification technique is 2^8 message modifications, but our latest implementation is very slow, equivalent to 2^{31} SHA-1 computation experimentally. However we conjecture that our algorithm can be improved by techniques of error correcting code and Gröbner basis. By using our methods, we have found many collisions for 58-round SHA-1.

Keywords: hash function, SHA-1, Gaussian elimination, Gröbner basis

1 Introduction

MD4 is a first dedicated hash function proposed by R. Rivest in 1990, and MD5 was proposed as an improved version of MD4 in 1991 also by R. Rivest. Following the same design paradigm, SHA-0 was published by NIST in 1993 and SHA-1 was issued by NIST in 1995 as a Federal Information Processing Standard. SHA-2 was also proposed by NIST as an improved version of SHA-1 where the length of hash results are 256, 384, 512.

In the first cryptanalysis of these algorithms, Dobbertin [1] has found semi-free start collision of MD5. Later on, Wang [5], [6] has proposed collision attack on SHA-0 whose complexity was estimated to be as 2^{45} SHA-0 computation. Chabaud-Joux [12] independently found differential collision attack against SHA-0 using essentially the same pattern. Introducing a new approach based on the neutral bit, near-collisions and multi-collisions, for SHA-0 and reduced SHA-1 have been reported in [10], [11], [9].

Employing the modular differential attack and message modification technique, Wang [4] has found collisions for the following hash functions MD4, MD5, HAVAL-128, RIPEMD, and in [7], [8], it is proposed how to break MD4, RIPEMD, MD5 and other hash functions, with the attack complexity against MD4 and MD5 proportional to 2^8 and 2^{37}, respectively. In [14] and [15], efficient collision search attacks against SHA-0 and 58-round SHA-1 have been reported as well as a complexity evaluation against full SHA-1 claimed to be 2^{69} SHA-1 computation and in the improved approach to be 2^{63}.

In this article, we give a sophisticated method to analyze SHA-1. Our method is based on the Gaussian elimination and Gröbner basis techniques. Our key ideas are to view a set of sufficient conditions as a system of equations of boolean functions and to consider message modifications as error-correcting procedures for non-linear codes. For 58-round SHA-1, the complexity of our algorithm using only a basic message modification technique to find a collision is 2^{29} message modifications (equivalent

* IT Security Center, Information-technology Promotion Agency, Japan, 2-28-8 Honkomagome, Bunkyo-ku Tokyo, 113-6591, Japan, m-sugita@ipa.go.jp
† Faculty of Liberal Arts and Sciences, Osaka Prefecture University, 1-1 Gakuen-cho Naka-ku Sakai Osaka 599-8531 Japan, kawazoe@las.osakafu-u.ac.jp
‡ Advanced Industrial Science and Technology (AIST), Akihabara Dai Bldg., 1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan; Department of Electrical, Electronic and Communication Engineering, Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga Bunkyo-ku, Tokyo 112-8551 Japan h-imai@aist.go.jp

round	Boolean function f_i	constant k_i
$1-20$	IF: $(x \wedge y) \vee (\neg x \wedge z)$	$0x5a827999$
$21-40$	XOR: $x \oplus y \oplus z$	$0x6ed6eba1$
$41-60$	MAJ: $(x \wedge y) \wedge (x \vee z) \wedge (y \vee z)$	$0x8fabbcdc$
$61-80$	XOR: $x \oplus y \oplus z$	$0xca62c1d6$

Table 1: Definition of function f_i

to 2^{31} SHA-1 computation experimentally), whereas Wang's method needs 2^{34} SHA-1 computation. We propose an improved algorithm using improved message modification whose complexity to find a collision for 58-round SHA-1 is 2^8 message modifications, but our latest implementation is very slow, equivalent to 2^{31} SHA-1 computation experimentally. However we conjecture that our algorithm can be improved by techniques of error correcting code and Gröbner basis. By using our methods, we have found many collisions for 58-round SHA-1 which are different from Wang's result.

2 Description of SHA-1 and Wang's analysis

2.1 SHA-1 algorithm

The hash function SHA-1 generates 160-bit hash result from message of length less than 2^{64} bits. It has Merkle/Damgard structure like other hash functions, and has 160-bit chaining value and 512-bit message block, and initial chaining values (IV) are fixed. From 512-bit block of the padded message, SHA-1 divides it into 16×32-bit words $(m_0, m_1, \cdots, m_{15})$ and expands the message by

$$m_i = (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}) \lll 1$$

for $i = 16, \cdots, 79$, where $x \lll n$ denotes n-bit left rotation of x. Using expanded messages, for $i = 0, 1, \cdots, 79$,

$$a_{i+1} = (a_i \lll 5) + f_i(b_i, c_i, d_i) + e_i + m_i + k_{i+1},$$

$$b_{i+1} = a_i,\ c_{i+1} = b_i \lll 30,\ d_{i+1} = c_i,\ e_{i+1} = d_i$$

where initial chaining value $IV = (a_0, b_0, c_0, d_0, e_0)$ is $(0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1f0)$ and function f_i is defined as in Table 1. In the following, we express 32-bit words as hexadecimal numbers.

2.2 Wang's attack

Wang's attack is summarized as follows.

- Find disturbance vector with low Hamming weight (difference for subtractions modulo 2^{32}).
- Construct differential paths by specifying conditions so that the differential path will occur with high probabilities.
- Generate a message randomly, modify it using message modification techniques, and find a collision.

By this method, Wang et al. has succeeded in finding collisions of MD4, MD5, RIPEMD, SHA-0 and 58-round SHA-1.

In the case of full-round SHA-1, Wang's attack need to use two iteration. They found collision with two iteration, i.e. each message in the collision includes two message blocks (1024-bit). They gives a set of sufficient conditions so that the differential occurs. Use a message modification technique they greatly improve the collision probability. In [15], they claimed that complexity to find a collision of full-round SHA-1 is 2^{69} and in CRYPTO'05 Rump Session, they claimed that they have improved complexity into 2^{63}. In the Rump Session, they claimed that they found new collision path of SHA-1 and described strategies for message modification. This strategy is: First they determine which message bits are possible candidates for modification. The message modification process must respect all chaining variable conditions and message conditions may require adding extra chaining variable conditions in round 1-16 and message conditions. Message modification follow certain topological order coming from correlations among chaining variable conditions.

Despite they have proposed new method, many details are still unpublished. Not all information are published about their attack, especially, 1) How to find differential paths? 2) How to modify messages properly?

In our analysis, we shall clarify and improve the second issue in the above, and show the effectiveness of our approach via computer experiment.

3 Definition and Notation

We take a complete set of representatives of $\mathbb{Z}/2^{32}\mathbb{Z}$ as $\{0, 1, 2, \ldots, 2^{32}-1\}$. So we identifies the ring $\mathbb{Z}/2^{32}\mathbb{Z}$ as the set $\{0, 1, 2, \ldots, 2^{32}-1\}$. When we ignore carry effects in the arithmetic of $\mathbb{Z}/2^{32}\mathbb{Z}$, we consider the ring $\mathbb{Z}/2^{32}\mathbb{Z}$

as the vector space $\mathbb{F}_2^{32}$ by using a set theoretical identification mapping
$\mathbb{F}_2^{32} \ni (x_0, x_1, \ldots, x_{31}) \mapsto x_0 2^{31} + x_1 2^{30} + \cdots + x_{30} 2^1 + x_{31} 2^0 \in \mathbb{Z}/2^{32}\mathbb{Z}$.

Definition 1 *Let $m = (m_0, m_1, \ldots, m_{31})$, $m' = (m'_0, m'_1, \ldots, m'_{31})$ be vectors of $\mathbb{F}_2^{32}$. For a pair m and m', we define the following notation.*

$$\Delta^+ m_j = \begin{cases} 1 & \text{if } m'_j = 1 \text{ and } m_j = 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$\Delta^- m_j = \begin{cases} 1 & \text{if } m'_j = 0 \text{ and } m_j = 1 \\ 0 & \text{otherwise,} \end{cases}$$

We define $\Delta^{\pm} m_j$ by $\Delta^{\pm} m_j = \Delta^+ m_j \oplus \Delta^- m_j$. Moreover, we define
$\Delta^+ m = (\Delta^+ m_0, \Delta^+ m_1, \ldots, \Delta^+ m_{31})$,
$\Delta^- m = (\Delta^- m_0, \Delta^- m_1, \ldots, \Delta^- m_{31})$
and $\Delta^{\pm} m = \Delta^+ m \oplus \Delta^- m$.

It is obvious that $\Delta^{\pm} m_j = m'_j + m_j \in \mathbb{F}_2$ and $\Delta^{\pm} m = m' + m \in \mathbb{F}_2^{32}$.

Using the above definition, a "disturbance vector" and a "differential without carry" are defined as follows.

Definition 2 *Let $m_i, a_i, b_i, c_i, d_i, e_i$ be as in the definition of SHA-1 and $m'_i, a'_i, b'_i, c'_i, d'_i, e'_i$ another message and its variables. They can be considered as vectors of $\mathbb{F}_2^{32}$. Then, following Wang's notation, we call a vector in the form $(\Delta^{\pm} m_i, \Delta^{\pm} a_i, \Delta^{\pm} b_i, \Delta^{\pm} c_i, \Delta^{\pm} d_i, \Delta^{\pm} e_i)_{i=0,1,\ldots,79}$ a "disturbance vector", and*
$(\Delta^+ m_i, \Delta^- m_i, \Delta^+ a_i, \Delta^- a_i, \ldots, \Delta^+ e_i, \Delta^- e_i)_{i=0,1,\ldots,79}$
a "differential without carry".

Since a disturbance vector ignores the sign '$\pm$', there are many different vectors $(\Delta^+ m_{i,j}, \Delta^- m_{i,j}, \ldots)$ corresponding to the same disturbance vector. So, the choice of a representative $(\Delta^+ m_{i,j}, \Delta^- m_{i,j}, \ldots)$, that is, the choice of a differential without carry is important in an analysis of SHA-1.

It is convenient to use the following definition to consider the ambiguity of the choice of a differential without carry.

Definition 3 *For a message space $M = \mathbb{Z}/2^{32}\mathbb{Z}$, we define function $f : (M \times M) \to M : (x_1, x_2) \mapsto (x_1 - x_2)$ where we consider '$-$' as subtraction of $\mathbb{Z}/2^{32}\mathbb{Z}$. We define differential δM by $\delta M = (M \times M)/\sim$ where for $\delta m_1, \delta m_2 \in \delta M$, $\delta m_1 \sim \delta m_2$ is satisfied if and only if $f(\delta m_1) = f(\delta m_2)$.*

Proposition 1 $\delta M \cong M$

Proof *This is obvious from the definition of δM.*

We define operator $+$ in δM as follows. For $\delta m_1 = (m_1^+, m_1^-) \in \delta M$, $\delta m_2 = (m_2^+, m_2^-) \in \delta M$,

$$\delta m_1 + \delta m_2 = (m_1^+ + m_2^+, m_1^- + m_2^-)$$

Same as the case of disturbance vectors, a choice of a representative (m, m') for a given class δm is very important. When δm is given as a part of a disturbance vector, we call a representative (m, m') for it a "message differential". The important problem is to find a good message differential. Heuristically, a good message differential has low Hamming weight. To find such good message differential, we use the following calculation.

- Calculate $\delta m_3 = (m_3^+, m_3^-) = \delta m_1 + \delta m_2 = (m_1^+ + m_2^+, m_1^- + m_2^-)$.
- Cancel the bit of (m_3^+, m_3^-): If $m_{3,j}^+ = m_{3,j}^- = 1$, change $m_{3,j}^+ = m_{3,j}^- = 0$.

We define operator $-$ in δM as follows. For $\delta m_1 = (m_1^+, m_1^-)$, $\delta m_2 = (m_2^+, m_2^-)$,

$$\delta m_1 - \delta m_2 = (m_1^+ + m_2^-, m_1^- + m_2^+)$$

In calculation, we also use the steps given below.

- Calculate $\delta m_3 = (m_3^-, m_3^-) = \delta m_1 - \delta m_2 = (m_1^+ + m_2^-, m_1^- + m_2^+)$
- Cancel the bit of (m_3^+, m_3^-): If $m_{3,j}^+ = m_{3,j}^- = 1$, change $m_{3,j}^+ = m_{3,j}^- = 0$.

In order to check whether $\delta m_1 = \delta m_2$ or not, we only have to calculate $\delta m_1 - \delta m_2$ and check $\delta m_1 - \delta m_2 = (0, 0)$.

4 Our method

Our method to cryptanalyze for SHA-1 is as follows.

1. Find disturbance vector with low Hamming weight from 21-round to final round (in Wang's example of SHA-1, 58 or 80-round). In this calculation we approximate MAJ function as XOR which holds with probability 3/4 per round.

2. From first round to 20-round, find differential (difference for subtractions modulo 2^{32}) so that $\delta a_{-4}(= \delta e_0 \lll 2)$, $\delta a_{-3}(= \delta d_0 \lll 2)$, $\delta a_{-2}(\ \delta c_0 \lll 2)$, $\delta a_{-1}(= \delta b_0)$, δa_0 is a local collision. We ignore carry effects here.

3. Calculate *sufficient conditions* on $\{a_i\}_{i=0,1,\ldots,20}$ considering carry effect by our semi-automatic method.

4. Determine *advanced sufficient conditions* on m_i by the Gaussian elimination based method.

5. Determine our *advanced sufficient conditions*. (Obtained conditions are essentially Wang's sufficient conditions combined with information for message modification technique.)

6. Generate a message randomly, and modify it using message modification techniques and find collisions.

In the above, Step 4, 5 and 6 are based on our new idea. In Step 4, we use the Gaussian elimination and in Step 5, we use an idea from Gröbner basis techniques. A method used in Step 6 is based on an idea analogous to error-correcting for non-linear codes. The method of Step 1 and 2 is based on the essentially same idea of Wang's attack. So we omit the details of Step 1 and 2 and only describe steps after from Step 3.

4.1 Sufficient conditions for collisions

For a given disturbance vector (or a given differential without carry) we can determine sufficient conditions for collisions on m_i and a_i such that if m_i'' (and a_i'') satisfies these conditions, we can obtain a pair of messages whose differential coincides with a disturbance vector and gives a SHA-1 collision. By the construction, sufficient conditions depend on a choice of a disturbance vector and its differential without carry.

4.2 How to calculate sufficient conditions on a_i?

In this step, we may only consider expanded messages by ignoring relations arising from message expansion.

For a given disturbance vector, we calculate sufficient conditions of chaining variables by adjusting b_i, c_i, d_i so that

$$\delta f(i, b_i, c_i, d_i) = \delta a_{i+1} - (\delta a_i \lll 5) - \delta e_i - \delta m_i.$$

In this calculation, we must adjust carry effects by hand. Although it is difficult to calculate full-automatically, our method is semi-automatic one.

4.3 Gaussian elimination and advanced sufficient conditions

Here we consider to analyze n-round SHA-1 $(58 \leq n \leq 80)$. In order to calculate the sufficient condition on $\{m_{i,j}\}_{i=0,1,\cdots,n;j=0,1,\ldots,31}$, we must take into account that $\Delta^+ m_{i,j} = 1$ implies $m_{i,j} = 0$ and $\Delta^- m_{i,j} = 1$ implies $m_{i,j} = 1$. This is done manually.

Moreover we also consider the relations derived from the key expansion

$$m_i = (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}) \lll 1$$

and we can rewrite all conditions on $0-58$-round by relations of $0-15$-round using the Gaussian elimination. Here all relations are considered as equations over $\mathbb{F}_2$ and an elimination order of $\{m_{i,j}\}_{i=0,1,\ldots,15;j=0,1,\ldots,31}$ is given by

$$m'_{i',j'} \leq m_{i,j} \text{ if } i' \leq i \text{ or } (i' = i \text{ and } j' \leq j).$$

Execute the Gaussian elimination for the system of equations which consists of all conditions on $0-58$ round, we obtain a reduced conditions only on $0-15$-round.

The important thing is that $m_{i,j}$ can be viewed as a polynomial on $a_{k,l}$, $(k \leq i+1)$, because $m_{i,j}$ can be viewed as a boolean function on $a_{k,l}$, $(k \leq i+1)$ by the definition of SHA-1. So it is useful to consider an elimination order of $\{a_{i,j}\}$. We can consider an elimination order of $\{a_{i,j}\}_{i=0,1,\ldots,15;j=0,1,\ldots,31}$ by

$$a'_{i',j'} \leq a_{i,j} \text{ if } i' \leq i \text{ or } (i' = i \text{ and } j' \leq j).$$

These two orders are different but approximately similar because transformation between them is not so complicated.

Experimentally, the best choice of the order is combination of these two orders. Hereafter, we adopt the order of $\{a_{i,j}\}$ when $i = 0, 1, 15, 16$, and the order of $\{m_{i,j}\}$ when $1 < i < 15$. By using the Gaussian elimination with this order, we reduced a system of equations consists of original sufficient conditions to a reduced row echelon form. Then in spite of original sufficient conditions, we use the obtained system of equations in reduced row echelon form as new sufficient conditions. We call them advanced sufficient conditions. On the other hand, for conditions on $\{a_{i,j}\}$, we construct advanced sufficient conditions by adding the information on "control bits" defined in the next section to original sufficient conditions.

4.4 Message modification techniques of m_i

In our procedure we use technique of modifying $\{a_{i,j}\}$ instead of $\{m_{i,j}\}$. We note that in [6] and [5], this technique has been explained but not in detail.

When $(a_0, b_0, c_0, d_0, e_0)$ is fixed, it is clear that $(m_0, m_1, \cdots, m_{15})$ corresponds to $(a_1, a_2, \cdots, a_{16})$ bijectively, which implies that modification of $\{a_{i,j}\}$ is theoretically equivalent to modification of $\{m_{i,j}\}$ in the case of SHA-1.

To find a collision, we start from a random message and then modify it to satisfy sufficient conditions. Message modification technique is used to find a collision for the first 23 rounds.

First we compile a list of controlled relations and control bits associated to first 23-rounds. The set of controlled relations consists of advanced sufficient conditions containing $\{m_{i,j}\}$ and $\{a_{i,j}\}$, $(i = 0, 1, \ldots, 15;\ j = 0, 1, \ldots, 31)$. Control bits are determined for each controlled relation. Control bits are chosen among $a_{i,j}$ which appears in a leading term or a term 'near' leading term in $m_{i,j}$, where $m_{i,j}$ is considered as a boolean function on $a_{i,j}$'s.

If a controlled relation is not satisfied by a current message, we adjust the message by changing values of control bits associated to the controlled relation. In the list, controlled relations are listed following the elimination order used in the Gaussian elimination. Each controlled relation with control bits associated to it is labeled by s_i where i denotes the order in the list.

By using the above setting, a basic procedure for the message modification is given as follows.

Algorithm 1 (Basic Message Modification) *Procedures for message modification: Preset the maximal number of trials M.*

1. *Set $r = 0$.*
2. *Generate $(a_1, a_2, \cdots, a_{16})$ randomly.*
3. *Set $i = 0$.*
4. *Increment i until the controlled relation r_i of s_i is not satisfied. If all relations are satisfied go to final step. If $r > M$, give up and return to Step 2.*
5. *Adjust control bits $a_{i,j}$ of s_i so that corresponding controlled relation and sufficient condition on $\{a_{i,j}\}$ hold. After adjusting, set $i = 0$ and $r = r + 1$ and go to Step 3 and repeat the process until all controlled relations hold.*
6. *If all controlled relations are satisfied, check whether modified message yields collision or not. If it does not generate collision, return to Step 2. If it generates collision, finish.*

The most important issue is that changing the control bit $a_{i,j}$ may effect the controlled relation $r_k(k < i)$ of previous step. In such situation, we have to go back to $i = k$ and correct controlled relations again.

By the proposed method, we can modify a message so that all sufficient conditions on the message $\{m_{i,j}\}$ and all sufficient conditions on the chaining variable $\{a_{ij}\}$ of first 23 rounds hold.

As we show later, Algorithm 1 improves the complexity of attack on 58-round SHA-1 comparing to Wang's method, but we need further improvement. In the following sections, we propose a more effective algorithm.

4.5 Neutral bit, semi-neutral bit and adjuster

By using semi-neutral bits defined below, we can make Algorithm 1 more efficient.

Assume that message conditions and some chaining variable conditions are satisfied. If changing some bit of chaining variable does not affect these conditions, the bit is called a neutral bit, following Wang's terminology. To adjust a message to satisfy remaining conditions, it is useful to use neutral bits. But in the case of SHA-1, there are not enough neutral bits. Here we introduce a notion of semi-neutral bits, a generalization of neutral bits. Assume again that message conditions and some chaining variable conditions are satisfied. If an effect of changing a bit of chaining variable can be easily eliminated so that all conditions previously satisfied are satisfied, we call the bit as a *semi-neutral bit*. Effects of changing semi-neutral bits can be eliminated by controlling a little number of bits. We call such bit an *adjuster*.

4.6 Improved algorithm to find collisions of SHA-1

Using semi-neutral bits and adjusters, we construct a more efficient algorithm to find collisions of SHA-1.

A new procedure to find collisions of SHA-1 is as follows.

Algorithm 2 (Improved Message Modification) *Procedures for message:*

1. *Generate* $(a_1, a_2, \cdots, a_{16})$ *randomly.*
2. *Using the basic message modification described in Algorithm 1, modify* $(a_1, a_2, \cdots, a_{16})$ *so that all message conditions and some chaining variable conditions from the 17-th round to the 23-rd round hold. If this step fails, return to Step 1.*
3. *If remaining changing variable conditions from the 17-th round to the 23-th round are not satisfied, return to Step 1 and repair until all conditions are satisfied (It can be satisfied probabilistically).*
4. *Change values of semi-neutral bits and modify chaining variables using our control sequence, and check whether chaining variable conditions from the 24-th round to the final round are satisfied.*
5. *Repeat all procedure above until all chaining variable conditions are satisfied.*

Remark 1 *(1) In round 17-23, there are uncontrolled relations. In the case of our experiment on 58-round SHA-1(see Section 6), there are 5 uncontrolled relations. So, in Algorithm 2, the probability that output of Step 2 pass the test in Step 3 is* $1/2^5$.

(2) As we show in Section 6, in the case of our experiment on 58-round SHA-1, we use 21 semi-neutral bits and 16 adjusters.

The above proposed algorithm is based on our idea that message modification is analogous to error-correcting procedure for non-linear codes. (See the next section for more details.) For Step 4 in Algorithm 2, we take a naive trial-and-error method in our latest implementation. We think that if we assemble a list of relations and their control bits for after the 23-rd round, and if we use more techniques from Gröbner basis and error-correcting codes, we can make our algorithm more effective.

5 Algebraic Description of Message Modification and the Relation to Error-Correcting Codes

Here we give another point of view which may be useful for further improvements.

5.1 Algebraic Description of message modification.

We can explain Algorithm 2 in terms of ideals of a polynomial ring and Gröbner basis. Here we consider n-round SHA-1 ($58 \leq n \leq 80$).

Let $\mathbb{F}_2[\mathbf{X}]$ be a polynomial ring over $\mathbb{F}_2$ with variables $X_{i,j}$, $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, 31$. Let J be an ideal in $\mathbb{F}_2[\mathbf{X}]$ generated by $\{X_{i,j}^2 + X_{i,j}\}_{i=0,1,\ldots,n;j=0,1,\ldots,31}$ and R a quotient ring $\mathbb{F}_2[\mathbf{X}]/J$. Note that R represents the set of all boolean functions with variables $X_{i,j}$, $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, 31$. For the simplicity of notation, we write an element in R as $f(\mathbf{X})$.

For a randomly taken $(a_1, a_2, \cdots, a_{16}) \in (\mathbb{F}_2^{32})^{16}$, $\mathbf{a} = \{a_{i,j}\}_{i=0,1,\ldots,n;j=0,1,\ldots,31}$ are determined. We associate this $\mathbf{a}$ to the ideal in R generated by $\{X_{i,j} + a_{i,j}\}_{i=0,1,\ldots,n;j=0,1,\ldots,31}$. controlled relations are polynomials in $a_{i,j}$'s and $m_{i,j}$'s. Since $m_{i,j}$ is determined by $a_{i,j}$'s, we may consider those relations as functions on $a_{i,j}$'s. Moreover, since controlled relations are equations via boolean functions, they can be expressed as polynomials on $a_{i,j}$'s. So by replacing $a_{i,j}$ by the variable $X_{i,j}$, we may consider controlled relations are equations in the form $f(\{X_{i,j}\}) = 0$ where $f \in R$. Put $g_{i,j} = X_{i,j} + a_{i,j}$ for each i, j, let I be an ideal generated by $g_{i,j}$'s and let $(f_1, f_2, \ldots)$ an ordered set of polynomials associated to the list of controlled relations. controlled relation and control bits in the list are replaced by f_i's and $g_{i,j}$. We call f_i a control equation and we call $g_{i,j}$ corresponding a control bit a control polynomial.

Let $T := \{f_j\}$ be the set of all conditions in a table of advanced sufficient conditions on which changing semi-neutral bits affect. Let

N be the set of all semi-neutral bits and adjusters. Put $P := \{(i,j) \mid a_{i,j} \in N\}$ and let I_2 be the ideal generated by all polynomials $g_{i,j} = X_{i,j} + a_{i,j}$ for $(i,j) \notin P$ and let R_2 a quotient ring R/I_2. For each f_j in T, let $\bar{f}_j$ be an equation $f_j \mod I_2$ and let $\mathcal{T}$ a system of equations which consists of all $\bar{f}_j$.

Then, Algorithm 2 is described as follows.

Algorithm 3 *Procedures for message modification: Preset the maximal number of trials M.*

1. *Set $r = 0$.*
2. *Generate $(a_1, a_2, \cdots, a_{16}) \in (\mathbb{F}_2^{32})^{16}$ randomly.*
3. *Set $i = 0$.*
4. *Increment i until $f_i \not\equiv 0 \mod I$. If all f_i are contained in I, go to the final step. If $r > M$, give up and return to Step 2.*
5. *For control polynomials $\{g_{j,l}\}$ associated to f_i, replace appropriate $g_{j,l}(X_{j,l})$ by $g_{j,l}(X_{j,l}+1)$ in I to satisfy $f_i \equiv 0 \mod I$. After adjusting, set $r = r + 1$ and go to Step 3.*
6. *Solve a system of equations $\mathcal{T}$ in R_2 by using Gröbner basis algorithm.*
7. *Check whether modified message yields collision or not. If it does not generate collision, return to Step 2. If it generates collision, finish.*

We remark that in a system of polynomial equation considered in Step 6 in the above algorithm, most of equations coming from controlled relations are trivial, that is, $\bar{f}_i \equiv 0$ in R_2.

5.2 Relation between message modification and decoding of error-correcting codes.

Let S be the set of all points in $F = \left(\mathbb{F}_2^{32}\right)^{16}$ satisfying advanced sufficient conditions on $\{a_{i,j}\}$. Note that S is a non-linear subset of F because there are non-linear conditions. Then, for a given $\mathbf{a} \in F$ which is not necessarily contained in S, to find an element in S by modifying $\mathbf{a}$ is analogous to a decoding problem in error-correcting codes. Hence, a basic message modification and a proposed improved message modification including changing semi-neutral bits can be viewed as an error-correcting process for a non-linear code S in F. More precisely, for a non-linear code S in F, an error-correction can be achieved by manipulating control bits and semi-neutral bits.

6 Analysis of 58-round SHA-1 based on our method

Now we show the effectiveness of our method by analyzing 58-round SHA-1.

6.1 Disturbance vector and Message differential pattern

We start from the disturbance vector which is the same as the one Wang gave. (Of course, our method is applicable to other disturbance vectors.) Then we construct differential without carry associated to the disturbance vector. Constructed one is the same one as Wang obtained in [15]. Explicit form of the differential without carry is as in Table 6.1.

We take $\{(\Delta^+ m_i, \Delta^- m_i)\}_{i=0,1,2,\ldots,57}$ as a message-differential. It is a message-differential without continuous 5-bits.

6.2 Sufficient conditions on $\{m_i\}$ and $\{a_i\}$

For the disturbance vector, the differential without carry and the message differential given in the previous step, we give sufficient conditions on 58-round SHA-1. Since it is not written in [15], conditions we give here in Table 3 is the first one which is written in an explicit form.

In Table 3, '`a`' means $a_{i,j} = a_{i-1,j}$, '`A`' means $a_{i,j} = a_{i-1,j}+1$, '`b`' means $a_{i,j} = a_{i-1,(j+2 \mod 32)}$, '`B`' means $a_{i,j} = a_{i-1,(j+2 \mod 32)}+1$, '`c`' means $a_{i,j} = a_{i-2,(j+2 \mod 32)}$ and '`C`' means $a_{i,j} = a_{i-2,(j+2 \mod 32)} + 1$.

By the Gaussian elimination, we rewrite all conditions on $0-57$-round by relations of $0-15$-round. An elimination order of $\{m_{i,j}\}_{i=0,1,\ldots,15;j=0,1,\ldots,31}$ we use here is

$$m'_{i',j'} \leq m_{i,j} \text{ if } i' \leq i \text{ or } (i' = i \text{ and } j' \leq j).$$

The result of Gaussian elimination is as follows.

$m_{15,31} = 1, m_{15,30} = 1, m_{15,29} = 0, m_{15,28} +$

i	$\Delta^+ m_i$	$\Delta^- m_i$	$\Delta^+ a_i$	$\Delta^- a_i$
58	4	0	0	0
57	0	0	0	0
56	0	0	0	0
55	0	0	0	0
54	0	0	0	0
53	0	0	0	0
52	0	0	0	0
51	0	0	0	0
50	0	0	0	0
49	0	0	0	0
48	0	0	0	0
47	80000000	0	0	0
46	0	80000000	0	0
45	0	0	0	0
44	0	80000002	0	0
43	0	40	2	0
42	0	80000000	0	0
41	0	40	2	0
40	0	80000000	0	0
39	80000000	40	2	0
38	0	0	0	0
37	40	80000000	0	2
36	0	80000002	0	0
35	80000000	0	0	0
34	80000000	2	0	0
33	40	0	0	2
32	0	2	0	0
31	2	40000000	0	0
30	40000002	40	2	0
29	2	40000040	2	0
28	1	80000000	0	0
27	42	40000020	0	1
26	40000041	80000002	0	2
25	0	40000002	0	0
24	1	0	0	0
23	2	c0000020	1	0
22	80000041	40000002	0	2
21	40000040	2	0	2
20	0	3	0	0
19	40000000	22	1	0
18	c0000002	41	2	0
17	40000002	40	2	0
16	80000001	0	0	0
15	20000000	60	1	0
14	20000001	0	0	0
13	80000040	0	0	2
12	0	a0000000	0	0
11	40000000	a0000052	102	80000000
10	40000040	0	0	0
9	40000040	12	8003ff00	40002
8	3	0	1fe0000	2000000
7	0	20	209	100180
6	80000001	0	1008000	4000
5	0	60000002	10100600	08080801
4	e0000040	2	8012	4024
3	20000000	40	201	0
2	20000000	40000043	80000014	60000002
1	40000020	20000012	40000000	20000000
0	20000000	0	0	0

Table 2: $\{m_i\}$ and $\{a_i\}$ of differential without carry of 58-round SHA-1

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	-10-----	--------	--------	-1----11
m_3	--0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	-11-----	--------	--------	------1-
m_6	0-------	--------	--------	-------0
m_7	--------	--------	--------	--1-----
m_8	--------	--------	--------	------00
m_9	-0------	--------	--------	-0-1--1-
m_{10}	-0------	--------	--------	-0------
m_{11}	101-----	--------	--------	-1-1--1-
m_{12}	1-1-----	--------	--------	--------
m_{13}	0-------	--------	--------	-0------
m_{14}	--0-----	--------	--------	-------0
m_{15}	--0-----	--------	--------	-11-----
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10
m_{23}	11------	--------	--------	--1---0-
m_{24}	--------	--------	--------	-------0
m_{25}	-1------	--------	--------	------1-
m_{26}	10------	--------	--------	-0----10
m_{27}	-1------	--------	--------	-01---0-
m_{28}	1-------	--------	--------	-------0
m_{29}	-1------	--------	--------	-1----0-
m_{30}	-0------	--------	--------	-1----0-
m_{31}	-1------	--------	--------	------0-
m_{32}	--------	--------	--------	------1-
m_{33}	--------	--------	--------	-0------
m_{34}	0-------	--------	--------	------1-
m_{35}	0-------	--------	--------	--------
m_{36}	1-------	--------	--------	------1-
m_{37}	1-------	--------	--------	-0------
m_{38}	--------	--------	--------	--------
m_{39}	0-------	--------	--------	-1------
m_{40}	1-------	--------	--------	--------
m_{41}	--------	--------	--------	-1------
m_{42}	1-------	--------	--------	--------
m_{43}	--------	--------	--------	-1------
m_{44}	1-------	--------	--------	------1-
m_{45}	--------	--------	--------	--------
m_{46}	1-------	--------	--------	--------
m_{47}	0-------	--------	--------	--------
m_i $(i \geq 48)$	--------	--------	--------	--------

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101-----	--------	--------	-1-a10aa
a_2	01100---	------0-	----a---	1--00010
a_3	0010----	-10---1a	------0-	0a-1a0-0
a_4	11010---	-01-----	01aaa---	0-10-100
a_5	10-01a--	-1-01-aa	--00100-	0---01-1
a_6	11--0110	-a-1001-	01100010	1-a111-1
a_7	-1--1110	a1a1111-	-101-001	1---0-10
a_8	-0----10	0000000a	a001a1--	100-0-1-
a_9	00------	11000100	00000000	101-1-1-
a_{10}	0-1-----	11111011	11100000	00--0-1-
a_{11}	1-0-----	-------1	01111110	11----0-
a_{12}	0-1-----	--------	--------	-1--a---
a_{13}	1-0-----	--------	--------	-1---01-
a_{14}	1-------	--------	--------	-1---1--
a_{15}	0-------	--------	--------	----0--0
a_{16}	-1------	--------	--------	----a---
a_{17}	-0------	--------	--------	----100-
a_{18}	1-1-----	--------	--------	-----00-
a_{19}	--------	--------	--------	-------0
a_{20}	-C------	--------	--------	----A---
a_{21}	-b------	--------	--------	----a-1-
a_{22}	--------	--------	--------	-----A1-
a_{23}	--------	--------	--------	-------0
a_{24}	-c------	--------	--------	--------
a_{25}	-B------	--------	--------	----a---
a_{26}	--------	--------	--------	-----A1-
a_{27}	--------	--------	--------	-------1
a_{28}	-c------	--------	--------	----A---
a_{29}	-B------	--------	--------	----A-0-
a_{30}	--------	--------	--------	------0-
a_{31}	--------	--------	--------	--------
a_{32}	--------	--------	--------	----A---
a_{33}	--------	--------	--------	------1-
a_{34}	--------	--------	--------	--------
a_{35}	--------	--------	--------	--------
a_{36}	--------	--------	--------	----A---
a_{37}	--------	--------	--------	------1-
a_{38}	--------	--------	--------	----A---
a_{39}	B-------	--------	--------	------0-
a_{40}	C-------	--------	--------	----A---
a_{41}	B-------	--------	--------	------0-
a_{42}	C-------	--------	--------	----A---
a_{43}	B-------	--------	--------	------0-
a_{44}	C-------	--------	--------	--------
a_{45}	B-------	--------	--------	--------
a_i $(i \geq 46)$	--------	--------	--------	--------

Table 3: Sufficient condition on $\{m_{ij}\}$ and $\{a_{i,j}\}$ of 58-round SHA-1

$m_{10,28}+m_{8,29}+m_{7,29}+m_{4,28}+m_{2,28}=1, m_{15,27}+m_{14,25}+m_{12,28}+m_{12,26}+m_{10,28}+m_{9,27}+m_{9,25}+m_{8,29}+m_{8,28}+m_{7,28}+m_{7,27}+m_{6,26}+m_{5,28}+m_{4,26}+m_{3,25}+m_{2,28}+m_{1,25}+m_{0,28}=1, m_{15,26}+m_{10,28}+m_{10,26}+m_{8,28}+m_{8,27}+m_{7,27}+m_{6,29}+m_{5,27}+m_{4,26}+m_{2,27}+m_{2,26}+m_{0,27}=1, m_{15,25}+m_{11,28}+m_{10,27}+m_{10,25}+m_{9,28}+m_{8,27}+m_{8,26}+m_{7,26}+m_{6,29}+m_{6,28}+m_{5,26}+m_{4,25}+m_{3,28}+m_{2,28}+m_{2,26}+m_{2,25}+m_{1,28}+m_{0,28}+m_{0,26}=0, m_{15,24}+m_{12,28}+m_{11,27}+m_{10,26}+m_{10,24}+m_{9,28}+m_{9,27}+m_{8,29}+m_{8,26}+m_{8,25}+m_{7,25}+m_{6,29}+m_{6,28}+m_{6,27}+m_{5,25}+m_{4,28}+m_{4,24}+m_{3,28}+m_{3,27}+m_{2,27}+m_{2,25}+m_{2,24}+m_{1,28}+m_{1,27}+m_{0,27}+m_{0,25}=1, m_{15,23}+m_{12,28}+m_{12,27}+m_{11,26}+m_{10,25}+m_{10,23}+m_{9,27}+m_{9,26}+m_{8,28}+m_{8,25}+m_{8,24}+m_{7,29}+m_{7,24}+m_{6,28}+m_{6,27}+m_{6,26}+m_{5,24}+m_{4,27}+m_{4,23}+m_{3,27}+m_{3,26}+m_{2,26}+m_{2,24}+m_{2,23}+m_{1,27}+m_{1,26}+m_{0,26}+m_{0,24}=1, m_{15,22}+m_{14,25}+m_{12,28}+m_{12,27}+m_{11,25}+m_{10,27}+m_{10,24}+m_{10,22}+m_{9,28}+m_{9,27}+m_{9,26}+m_{8,27}+m_{8,24}+m_{8,23}+m_{7,28}+m_{7,27}+m_{7,23}+m_{6,27}+m_{6,25}+m_{5,23}+m_{4,28}+m_{4,27}+m_{4,22}+m_{3,26}+m_{2,28}+m_{2,27}+m_{2,25}+m_{2,23}+m_{2,22}+m_{1,26}+m_{0,25}+m_{0,23}=0, m_{15,6}=1, m_{15,5}=1, m_{15,4}+m_{12,5}+m_{10,4}+m_{4,5}+m_{4,4}+m_{2,5}+m_{2,4}=1, m_{15,3}+m_{12,2}+m_{10,2}+m_{8,3}+m_{7,3}+m_{7,2}+m_{5,3}+m_{4,2}+m_{3,4}+m_{3,2}+m_{2,3}+m_{2,2}+m_{1,2}+m_{0,3}=0, m_{15,2}+m_{12,5}+m_{11,5}+m_{10,4}+m_{10,2}+m_{8,4}+m_{8,3}+m_{7,3}+m_{5,5}+m_{5,3}+m_{4,5}+m_{4,2}+m_{2,5}+m_{2,3}+m_{2,2}+m_{0,3}=1, m_{15,1}+m_{12,5}+m_{11,3}+m_{11,2}+m_{10,4}+m_{10,2}+m_{9,2}+m_{8,3}+m_{8,2}+m_{5,4}+m_{4,5}+m_{4,4}+m_{4,0}+m_{3,31}+m_{3,4}+m_{3,2}+m_{2,5}+m_{2,4}+m_{2,3}+m_{1,31}+m_{0,3}=0, m_{15,0}+m_{1,0}=1, m_{14,31}=0, m_{14,30}=1, m_{14,29}=0, m_{14,28}+m_{9,28}+m_{6,29}+m_{3,28}+m_{1,28}=0, m_{14,27}+m_{12,28}+m_{9,27}+m_{7,29}+m_{6,28}+m_{4,28}+m_{3,27}+m_{1,27}=0, m_{14,26}+m_{12,27}+m_{10,28}+m_{9,28}+m_{9,26}+m_{7,28}+m_{6,27}+m_{4,28}+m_{4,27}+m_{3,26}+m_{2,28}+m_{1,26}=1, m_{14,24}+m_{12,27}+m_{12,25}+m_{11,28}+m_{10,27}+m_{10,26}+m_{9,26}+m_{9,24}+m_{8,29}+m_{7,26}+m_{6,29}+m_{6,25}+m_{5,28}+m_{4,28}+m_{4,26}+m_{4,25}+m_{3,28}+m_{3,24}+m_{2,26}+m_{1,24}+m_{0,28}=0, m_{14,23}+m_{12,26}+m_{12,24}+m_{11,27}+m_{10,26}+m_{10,25}+m_{9,28}+m_{9,25}+m_{9,23}+m_{8,28}+m_{7,25}+m_{6,28}+m_{6,24}+m_{5,27}+m_{4,27}+m_{4,25}+m_{4,24}+m_{3,28}+m_{3,27}+m_{3,23}+m_{2,25}+m_{1,28}+m_{1,23}+m_{0,27}=1, m_{14,22}+m_{13,20}+m_{12,25}+m_{12,24}+m_{12,23}+m_{11,28}+m_{11,23}+m_{11,21}+m_{10,27}+m_{9,26}+m_{9,24}+m_{9,23}+m_{8,29}+m_{8,27}+m_{8,26}+m_{8,25}+m_{8,22}+m_{8,20}+m_{7,26}+m_{7,25}+m_{6,29}+m_{6,23}+m_{6,22}+m_{5,28}+m_{5,25}+m_{5,21}+m_{4,28}+m_{4,26}+m_{4,25}+m_{4,23}+m_{3,28}+m_{3,24}+$

$m_{3,21}+m_{2,26}+m_{2,20}+m_{1,24}+m_{0,28}+m_{0,25}+m_{0,20}=1, m_{14,21}+m_{12,27}+m_{12,24}+m_{12,22}+m_{11,25}+m_{10,28}+m_{10,27}+m_{10,24}+m_{10,23}+m_{9,28}+m_{9,26}+m_{9,23}+m_{9,21}+m_{8,29}+m_{8,26}+m_{7,29}+m_{7,28}+m_{7,23}+m_{6,29}+m_{6,26}+m_{6,22}+m_{5,25}+m_{4,28}+m_{4,27}+m_{4,25}+m_{4,23}+m_{4,22}+m_{3,26}+m_{3,25}+m_{3,21}+m_{2,28}+m_{2,23}+m_{1,26}+m_{1,21}+m_{0,25}=0, m_{14,20}+m_{12,26}+m_{12,23}+m_{12,21}+m_{11,28}+m_{11,24}+m_{10,28}+m_{10,27}+m_{10,26}+m_{10,23}+m_{10,22}+m_{9,27}+m_{9,25}+m_{9,22}+m_{9,20}+m_{8,28}+m_{8,25}+m_{7,28}+m_{7,27}+m_{7,22}+m_{6,29}+m_{6,28}+m_{6,25}+m_{6,21}+m_{5,24}+m_{4,27}+m_{4,26}+m_{4,24}+m_{4,22}+m_{4,21}+m_{3,28}+m_{3,25}+m_{3,24}+m_{3,20}+m_{2,27}+m_{2,22}+m_{1,25}+m_{1,20}+m_{0,28}+m_{0,24}+m_{47,31}=1, m_{14,5}+m_{8,5}+m_{6,5}=1, m_{14,4}+m_{12,5}+m_{11,3}+m_{11,2}+m_{10,4}+m_{10,3}+m_{10,2}+m_{10,1}+m_{9,2}+m_{8,5}+m_{7,2}+m_{6,5}+m_{6,4}+m_{5,4}+m_{5,2}+m_{4,5}+m_{4,4}+m_{4,0}+m_{3,31}+m_{3,4}+m_{3,2}+m_{2,5}+m_{2,3}+m_{2,2}+m_{1,31}+m_{0,4}+m_{0,3}+m_{0,2}=1, m_{14,3}+m_{11,3}+m_{11,2}+m_{8,2}+m_{7,4}+m_{7,2}+m_{7,1}+m_{6,2}+m_{5,3}+m_{4,0}+m_{3,3}+m_{2,2}+m_{1,31}+m_{1,3}=0, m_{14,2}+m_{12,5}+m_{12,3}+m_{10,4}+m_{9,2}+m_{7,4}+m_{6,3}+m_{4,5}+m_{4,4}+m_{4,3}+m_{3,2}+m_{2,5}+m_{2,4}+m_{1,2}=1, m_{14,1}+m_{12,4}+m_{11,2}+m_{10,2}+m_{9,3}+m_{8,3}+m_{7,2}+m_{6,2}+m_{5,5}+m_{5,2}+m_{4,4}+m_{3,31}+m_{3,4}+m_{3,2}+m_{3,1}+m_{2,4}+m_{2,3}+m_{0,3}=0, m_{14,0}=0, m_{13,31}=0, m_{13,30}=0, m_{13,29}+m_{8,29}=0, m_{13,28}+m_{8,28}+m_{2,28}+m_{0,28}=0, m_{13,27}+m_{11,28}+m_{8,29}+m_{8,27}+m_{6,29}+m_{5,28}+m_{3,28}+m_{2,27}+m_{0,27}=1, m_{13,26}+m_{11,27}+m_{9,28}+m_{8,28}+m_{8,26}+m_{6,28}+m_{5,27}+m_{3,28}+m_{3,27}+m_{2,26}+m_{1,28}+m_{0,26}=1, m_{13,24}+m_{12,28}+m_{11,27}+m_{11,25}+m_{10,28}+m_{9,27}+m_{9,26}+m_{8,29}+m_{8,26}+m_{8,24}+m_{7,29}+m_{7,28}+m_{6,26}+m_{5,25}+m_{4,28}+m_{3,28}+m_{3,26}+m_{3,25}+m_{2,28}+m_{2,24}+m_{1,28}+m_{1,26}+m_{0,24}=0, m_{13,23}+m_{12,27}+m_{11,26}+m_{11,24}+m_{10,28}+m_{10,27}+m_{9,26}+m_{9,25}+m_{8,29}+m_{8,28}+m_{8,25}+m_{8,23}+m_{7,29}+m_{7,28}+m_{7,27}+m_{6,25}+m_{5,28}+m_{5,24}+m_{4,28}+m_{4,27}+m_{3,27}+m_{3,25}+m_{3,24}+m_{2,27}+m_{2,23}+m_{1,27}+m_{1,25}+m_{0,28}+m_{0,23}=0, m_{13,22}+m_{12,26}+m_{11,28}+m_{11,25}+m_{11,23}+m_{10,27}+m_{10,26}+m_{9,28}+m_{9,25}+m_{9,24}+m_{8,28}+m_{8,27}+m_{8,24}+m_{8,22}+m_{7,28}+m_{7,27}+m_{7,26}+m_{6,29}+m_{6,24}+m_{5,28}+m_{5,27}+m_{5,23}+m_{4,27}+m_{4,26}+m_{3,28}+m_{3,26}+m_{3,24}+m_{3,23}+m_{2,28}+m_{2,26}+m_{2,22}+m_{1,26}+m_{1,24}+m_{0,28}+m_{0,27}+m_{0,22}=1, m_{13,6}=0, m_{13,5}+m_{12,5}+m_{5,5}+m_{4,5}+m_{2,5}=0, m_{13,4}+m_{12,5}+m_{11,2}+m_{10,4}+m_{7,4}+m_{5,4}+m_{5,3}+m_{5,2}+m_{4,5}+m_{4,4}+m_{3,31}+m_{2,5}+m_{2,4}+m_{2,2}+m_{1,2}=0, m_{13,3}+m_{8,3}+m_{5,4}+m_{3,4}+m_{2,3}+m_{0,3}=0, m_{13,2}+m_{10,3}+m_{10,2}+m_{10,1}+$

$m_{9,2}+m_{8,2}+m_{7,4}+m_{7,2}+m_{4,0}+m_{3,4}+m_{3,3}+m_{3,2}+m_{2,3}+m_{2,2}+m_{1,31}+m_{1,2}+m_{0,3}=0, m_{13,1}+m_{10,2}+m_{9,3}+m_{8,3}+m_{7,4}+m_{7,2}+m_{6,2}+m_{5,3}+m_{5,2}+m_{4,0}+m_{3,4}+m_{3,2}+m_{2,3}+m_{2,2}+m_{1,31}+m_{0,3}=0, m_{13,0}+m_{1,31}=1, m_{12,31}=1, m_{12,30}=0, m_{12,29}=1, m_{12,0}+m_{4,0}+m_{3,0}+m_{1,31}+m_{1,0}=0, m_{11,31}=1, m_{11,30}=0, m_{11,29}=1, m_{11,6}=1, m_{11,4}=1, m_{11,1}=1, m_{11,0}+m_{1,31}=0, m_{10,31}=0, m_{10,30}=0, m_{10,29}=0, m_{10,6}=0, m_{10,5}+m_{4,5}+m_{2,5}=0, m_{10,0}+m_{4,0}+m_{1,0}=0, m_{9,31}+m_{3,31}+m_{3,0}+m_{1,0}=1, m_{9,30}=0, m_{9,29}=1, m_{9,6}=0, m_{9,5}+m_{8,5}+m_{6,5}+m_{3,5}=0, m_{9,4}=1, m_{9,1}=1, m_{9,0}+m_{3,0}+m_{1,0}=0, m_{8,31}=0, m_{8,30}=1, m_{8,1}=0, m_{8,0}=0, m_{7,31}+m_{3,31}+m_{1,31}+m_{1,0}=0, m_{7,30}=1, m_{7,5}=1, m_{7,0}+m_{3,0}=0, m_{6,31}=0, m_{6,30}=0, m_{6,0}=0, m_{5,31}+m_{3,31}=0, m_{5,30}=1, m_{5,29}=1, m_{5,1}=1, m_{5,0}+m_{3,0}+m_{1,31}=1, m_{4,31}=0, m_{4,30}=0, m_{4,29}=0, m_{4,6}=0, m_{4,1}=1, m_{3,30}=1, m_{3,29}=0, m_{3,6}=1, m_{2,31}=0, m_{2,30}=1, m_{2,29}=0, m_{2,6}=1, m_{2,1}=1, m_{2,0}=1, m_{1,30}=0, m_{1,29}=1, m_{1,5}=0, m_{1,4}=1, m_{1,1}=1, m_{0,31}=0, m_{0,30}=0, m_{0,29}=0$

From derived equations, we obtain advanced sufficient conditions on $\{m_{i,j}\}$.

6.3 control bits and controlled relations

We determine control bits and controlled relations as in Table 4, and Table 5, where a control sequence denotes a pair of a control bit and a controlled relation.

Now we summarize our advanced sufficient conditions on $\{m_{i,j}\}$and $\{a_{i,j}\}$ by showing two tables (Table 6) which illustrate advanced sufficient conditions, controlled relations, control bits and semi-neutral bits.

Symbols in Table 6 mean:

- 'a', 'A', 'b', 'B', 'c', 'C': as in Section 6.2.
- 'L' means that it is the leading term of controlled relation of Table 4.
- 'w': adjust $a_{i,j}$ so that $m_{i+1,j}=0$.
- 'W': adjust $a_{i,j}$ so that $m_{i+1,j}=1$.
- 'v': adjust $a_{i,j}$ so that $m_{i,(j+27 \mod 32)}=0$.
- 'V': adjust $a_{i,j}$ so that $m_{i,(j+27 \mod 32)}=1$.
- 'h': adjust $a_{i,j}$ so that corresponding controlled relation including $m_{i+1,j}$ as leading term holds.

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	L10-----	--------	--------	-1----11
m_3	-L0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	L11-----	--------	--------	------1L
m_6	0L------	--------	--------	-------0
m_7	LL------	--------	--------	--1----L
m_8	LL------	--------	--------	------00
m_9	L0L-----	--------	--------	-0L1--1L
m_{10}	L0L-----	--------	--------	-0L----L
m_{11}	101-----	--------	--------	-1-1--1L
m_{12}	1L1-----	--------	--------	-------L
m_{13}	0LLLLL-L	LL------	--------	-0LLLLLL
m_{14}	LL0LLL-L	LLLL----	--------	--LLLLL0
m_{15}	LL0LLLLL	LL------	--------	-11LLLLL
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10
m_{23}	11------	--------	--------	--1---0-
m_{24}	--------	--------	--------	-------0
m_{25}	-1------	--------	--------	------1-
m_{26}	10------	--------	--------	-0----10
m_{27}	-1------	--------	--------	-01---0-
m_{28}	1-------	--------	--------	-------0
m_{29}	-1------	--------	--------	-1----0-
m_{30}	-0------	--------	--------	-1----0-
m_{31}	-1------	--------	--------	------0-
m_{32}	--------	--------	--------	------1-
m_{33}	--------	--------	--------	-0------
m_{34}	0-------	--------	--------	------1-
m_{35}	0-------	--------	--------	--------
m_{36}	1-------	--------	--------	------1-
m_{37}	1-------	--------	--------	-0------
m_{38}	--------	--------	--------	--------
m_{39}	0-------	--------	--------	-1------
m_{40}	1-------	--------	--------	--------
m_{41}	--------	--------	--------	-1------
m_{42}	1-------	--------	--------	--------
m_{43}	--------	--------	--------	-1------
m_{44}	1-------	--------	--------	------1-
m_{45}	--------	--------	--------	--------
m_{46}	1-------	--------	--------	--------
m_{47}	0-------	--------	--------	--------
m_i $(i \geq 48)$	--------	--------	--------	--------

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101V--vV	Y-------	--------	-1-a10aa
a_2	01100vVv	------0-	----a---	1-w00010
a_3	0010--Vv	-10---1a	------0-	0aX1a0W0
a_4	11010vv-	-01-----	01aaa---	0W10-100
a_5	10w01aV-	-1-01-aa	--00100-	0w--01W1
a_6	11W-0110	-a-1001-	01100010	1-a111W1
a_7	w1x-1110	a1a1111-	-101-001	1---0-10
a_8	h0Xvvv10	0000000a	a001a1--	100X0-1h
a_9	00XVrr-V	11000100	00000000	101-1-1y
a_{10}	0w1-rv-v	11111011	11100000	00hW0-1h
a_{11}	1w0--V-V	-------1	01111110	11x---0Y
a_{12}	0w1-rV-V	--------	--------	-1XWa-Wh
a_{13}	1w0--vv-	-rr-----	--------	-1-qq01y
a_{14}	1rhhvvVh	hh------	qNNNNNqN	N1hhh1hh
a_{15}	0rwhhhVh	hhhh---N	qNNqqNqN	NNhh0hh0
a_{16}	W1whhhhh	hhqNqNqN	NNqNNqqq	qWWhahhh
a_{17}	-0------	--------	--------	----100-
a_{18}	1-1-----	--------	--------	-----00-
a_{19}	--------	--------	--------	-------0
a_{20}	-C------	--------	--------	----A---
a_{21}	-b------	--------	--------	----a-1-
a_{22}	--------	--------	--------	-----A1-
a_{23}	--------	--------	--------	-------0
a_{24}	-c------	--------	--------	--------
a_{25}	-B------	--------	--------	----a---
a_{26}	--------	--------	--------	-----A1-
a_{27}	--------	--------	--------	-------1
a_{28}	-c------	--------	--------	----A---
a_{29}	-B------	--------	--------	----A-0-
a_{30}	--------	--------	--------	------0-
a_{31}	--------	--------	--------	--------
a_{32}	--------	--------	--------	----A---
a_{33}	--------	--------	--------	------1-
a_{34}	--------	--------	--------	--------
a_{35}	--------	--------	--------	--------
a_{36}	--------	--------	--------	----A---
a_{37}	--------	--------	--------	------1-
a_{38}	--------	--------	--------	----A---
a_{39}	B-------	--------	--------	------0-
a_{40}	C-------	--------	--------	----A---
a_{41}	B-------	--------	--------	------0-
a_{42}	C-------	--------	--------	----A---
a_{43}	B-------	--------	--------	------0-
a_{44}	C-------	--------	--------	--------
a_{45}	B-------	--------	--------	--------
a_i $(i \geq 46)$	--------	--------	--------	--------

Table 6: 'Advanced' sufficient condition on $\{m_{i,j}\}$ and $\{a_{i,j}\}$

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{124}	$a_{16,7}, a_{15,9}, a_{14,9}$	$a_{23,0} = 0$
s_{123}	$a_{16,9}$	$a_{22,2} + a_{21,2} = 1$
s_{122}	$a_{16,13}, a_{15,15}, a_{15,12}, a_{15,11}$	$a_{22,1} = 1$
s_{121}	$a_{16,10}$	$a_{21,3} + m_{20,3} = 0$
s_{120}	$a_{16,8}$	$a_{21,1} = 1$
s_{119}	$a_{16,15}, a_{16,20}$	$a_{20,3} + m_{19,3} = 1$
s_{118}	$a_{16,17}$	$a_{19,0} = 0$
s_{117}	$a_{16,21}$	$a_{18,31} = 1$
s_{116}	$a_{16,19}$	$a_{18,29} = 1$
s_{115}	$a_{13,4}$	$a_{18,2} = 0$
s_{114}	$a_{13,3}$	$a_{18,1} = 0$
s_{113}	$a_{14,15}$	$a_{17,30} = 0$
s_{112}	$a_{16,31}$	$m_{15,31} = 1$
s_{111}	$a_{16,29}$	$m_{15,29} = 0$
s_{110}	$a_{16,28}$	$m_{15,28} + m_{10,28} + m_{8,29} + m_{7,29} + m_{4,28} + m_{2,28} = 1$
s_{109}	$a_{16,27}, a_{13,28}$	$m_{15,27} + m_{14,25} + m_{12,28} + m_{12,26} + m_{10,28} + m_{9,27} + m_{9,25} + m_{8,29} + m_{8,28}$ $+m_{7,28} + m_{7,27} + m_{6,26} + m_{5,28} + m_{4,26} + m_{3,25} + m_{2,28} + m_{1,25} + m_{0,28} = 1$
s_{108}	$a_{16,26}$	$m_{15,26} + m_{10,28} + m_{10,26} + m_{8,28} + m_{8,27} + m_{7,27} + m_{6,29} + m_{5,27} + m_{4,26}$ $+m_{2,27} + m_{2,26} + m_{0,27} = 1$
s_{107}	$a_{16,25}$	$m_{15,25} + m_{11,28} + m_{10,27} + m_{10,25} + m_{9,28} + m_{8,27} + m_{8,26} + m_{7,26} + m_{6,29}$ $+m_{6,28} + m_{5,26} + m_{4,25} + m_{3,28} + m_{2,28} + m_{2,26} + m_{2,25} + m_{1,28} + m_{0,28} + m_{0,26} = 0$
s_{106}	$a_{16,24}$	$m_{15,24} + m_{12,28} + m_{11,27} + m_{10,26} + m_{10,24} + m_{9,28} + m_{9,27} + m_{8,29} + m_{8,26}$ $+m_{8,25} + m_{7,25} + m_{6,29} + m_{6,28} + m_{6,27} + m_{5,25} + m_{4,28} + m_{4,24} + m_{3,28} + m_{3,27}$ $+m_{2,27} + m_{2,25} + m_{2,24} + m_{1,28} + m_{1,27} + m_{0,27} + m_{0,25} = 1$
s_{105}	$a_{16,23}$	$m_{15,23} + m_{12,28} + m_{12,27} + m_{11,26} + m_{10,25} + m_{10,23} + m_{9,27} + m_{9,26} + m_{8,28}$ $+m_{8,25} + m_{8,24} + m_{7,29} + m_{7,24} + m_{6,28} + m_{6,27} + m_{6,26} + m_{5,24} + m_{4,27} + m_{4,23}$ $+m_{3,27} + m_{3,26} + m_{2,26} + m_{2,24} + m_{2,23} + m_{1,27} + m_{1,26} + m_{0,26} + m_{0,24} = 1$
s_{104}	$a_{16,22}$	$m_{15,22} + m_{14,25} + m_{12,28} + m_{12,27} + m_{11,25} + m_{10,27} + m_{10,24} + m_{10,22} + m_{9,28}$ $+m_{9,27} + m_{9,26} + m_{8,27} + m_{8,24} + m_{8,23} + m_{7,28} + m_{7,27} + m_{7,23} + m_{6,27}$ $+m_{6,25} + m_{5,23} + m_{4,28} + m_{4,27} + m_{4,22} + m_{3,26} + m_{2,28} + m_{2,27} + m_{2,25}$ $+m_{2,23} + m_{2,22} + m_{1,26} + m_{0,25} + m_{0,23} = 0$
s_{103}	$a_{16,6}$	$m_{15,6} = 1$
s_{102}	$a_{16,5}$	$m_{15,5} = 1$
s_{101}	$a_{16,4}$	$m_{15,4} + m_{12,5} + m_{10,4} + m_{4,5} + m_{4,4} + m_{2,5} + m_{2,4} = 1$
s_{100}	$a_{16,2}$	$m_{15,2} + m_{12,5} + m_{11,5} + m_{10,4} + m_{10,2} + m_{8,4} + m_{8,3} + m_{7,3} + m_{5,5}$ $+m_{5,3} + m_{4,5} + m_{4,2} + m_{2,5} + m_{2,3} + m_{2,2} + m_{0,3} = 1$
s_{99}	$a_{16,1}$	$m_{15,1} + m_{12,5} + m_{11,3} + m_{11,2} + m_{10,4} + m_{10,2} + m_{9,2} + m_{8,3} + m_{8,2}$ $+m_{5,4} + m_{4,5} + m_{4,4} + m_{4,0} + m_{3,31} + m_{3,4} + m_{3,2} + m_{2,5} + m_{2,4} + m_{2,3}$ $+m_{1,31} + m_{0,3} = 0$
s_{98}	$a_{16,0}$	$m_{15,0} + m_{1,0} = 1$
s_{97}	$a_{15,30}$	$m_{15,3} + m_{12,2} + m_{10,2} + m_{8,3} + m_{7,3} + m_{7,2} + m_{5,3} + m_{4,2} + m_{3,4} + m_{3,2}$ $+m_{2,3} + m_{2,2} + m_{1,2} + m_{0,3} = 0$
s_{96}	$a_{15,25}$	$m_{15,30} = 1$
s_{95}	$a_{14,26}$	$m_{14,31} = 0$
s_{94}	$a_{14,25}$	$m_{14,30} = 1$
s_{93}	$a_{15,29}$	$m_{14,29} = 0$
s_{92}	$a_{15,28}$	$m_{14,28} + m_{9,28} + m_{6,29} + m_{3,28} + m_{1,28} = 0$
s_{91}	$a_{15,27}$	$m_{14,27} + m_{12,28} + m_{9,27} + m_{7,29} + m_{6,28} + m_{4,28} + m_{3,27} + m_{1,27} = 0$
s_{90}	$a_{15,26}$	$m_{14,26} + m_{12,27} + m_{10,28} + m_{9,28} + m_{9,26} + m_{7,28} + m_{6,27} + m_{4,28} + m_{4,27}$ $+m_{3,26} + m_{2,28} + m_{1,26} = 1$
s_{89}	$a_{15,24}$	$m_{14,24} + m_{12,27} + m_{12,25} + m_{11,28} + m_{10,27} + m_{10,26} + m_{9,26} + m_{9,24} + m_{8,29}$ $+m_{7,26} + m_{6,29} + m_{6,25} + m_{5,28} + m_{4,28} + m_{4,26} + m_{4,25} + m_{3,28} + m_{3,24} + m_{2,26}$ $+m_{1,24} + m_{0,28} = 0$
s_{88}	$a_{15,23}$	$m_{14,23} + m_{12,26} + m_{12,24} + m_{11,27} + m_{10,26} + m_{10,25} + m_{9,28} + m_{9,25} + m_{9,23}$ $+m_{8,28} + m_{7,25} + m_{6,28} + m_{6,24} + m_{5,27} + m_{4,27} + m_{4,25} + m_{4,24} + m_{3,28} + m_{3,27}$ $+m_{3,23} + m_{2,25} + m_{1,28} + m_{1,23} + m_{0,27} = 1$
s_{87}	$a_{15,22}$	$m_{14,22} + m_{13,20} + m_{12,25} + m_{12,24} + m_{12,23} + m_{11,28} + m_{11,23} + m_{11,21} + m_{10,27}$ $+m_{9,26} + m_{9,24} + m_{9,23} + m_{8,29} + m_{8,27} + m_{8,26} + m_{8,25} + m_{8,22} + m_{8,20}$ $+m_{7,26} + m_{7,25} + m_{6,29} + m_{6,23} + m_{6,22} + m_{5,28} + m_{5,25} + m_{5,21} + m_{4,28}$ $+m_{4,26} + m_{4,25} + m_{4,23} + m_{3,28} + m_{3,24} + m_{3,21} + m_{2,26} + m_{2,20} + m_{1,24}$ $+m_{0,28} + m_{0,25} + m_{0,20} = 1$
s_{86}	$a_{15,21}$	$m_{14,21} + m_{12,27} + m_{12,24} + m_{12,22} + m_{11,25} + m_{10,28} + m_{10,27} + m_{10,24}$ $+m_{10,23} + m_{9,28} + m_{9,26} + m_{9,23} + m_{9,21} + m_{8,29} + m_{8,26} + m_{7,29} + m_{7,28}$ $+m_{7,23} + m_{6,29} + m_{6,26} + m_{6,22} + m_{5,25} + m_{4,28} + m_{4,27} + m_{4,25} + m_{4,23}$ $+m_{4,22} + m_{3,26} + m_{3,25} + m_{3,21} + m_{2,28} + m_{2,23} + m_{1,26} + m_{1,21} + m_{0,25} = 0$
s_{85}	$a_{15,20}$	$m_{14,20} + m_{12,26} + m_{12,23} + m_{12,21} + m_{11,28} + m_{11,24} + m_{10,28} + m_{10,27}$ $+m_{10,26} + m_{10,23} + m_{10,22} + m_{9,27} + m_{9,25} + m_{9,22} + m_{9,20} + m_{8,28} + m_{8,25}$ $+m_{7,28} + m_{7,27} + m_{7,22} + m_{6,29} + m_{6,28} + m_{6,25} + m_{6,21} + m_{5,24} + m_{4,27} + m_{4,26}$ $+m_{4,24} + m_{4,22} + m_{4,21} + m_{3,28} + m_{3,25} + m_{3,24} + m_{3,20} + m_{2,27} + m_{2,22} + m_{1,25}$ $+m_{1,20} + m_{0,28} + m_{0,24} + m_{47,31} = 1$

Table 4: Control bit and controlled relations of 58-round SHA-1 (I)

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{84}	$a_{15,5}$	$m_{14,5} + m_{8,5} + m_{6,5} = 1$
s_{83}	$a_{15,4}$	$m_{14,4} + m_{12,5} + m_{11,3} + m_{11,2} + m_{10,4} + m_{10,3} + m_{10,2} + m_{10,1} + m_{9,2}$ $+m_{8,5} + m_{7,2} + m_{6,5} + m_{6,4} + m_{5,4} + m_{5,2} + m_{4,5} + m_{4,4} + m_{4,0} + m_{3,31}$ $+m_{3,4} + m_{3,2} + m_{2,5} + m_{2,3} + m_{2,2} + m_{1,31} + m_{0,4} + m_{0,3} + m_{0,2} = 1$
s_{82}	$a_{14,30}$	$m_{14,3} + m_{11,3} + m_{11,2} + m_{8,2} + m_{7,4} + m_{7,2} + m_{7,1} + m_{6,2} + m_{5,3}$ $+m_{4,0} + m_{3,3} + m_{2,2} + m_{1,31} + m_{1,3} = 0$
s_{81}	$a_{15,2}$	$m_{14,2} + m_{12,5} + m_{12,3} + m_{10,4} + m_{9,2} + m_{7,4} + m_{6,3} + m_{4,5} + m_{4,4}$ $+m_{4,3} + m_{3,2} + m_{2,5} + m_{2,4} + m_{1,2} = 1$
s_{80}	$a_{15,1}$	$m_{14,1} + m_{12,4} + m_{11,2} + m_{10,2} + m_{9,3} + m_{8,3} + m_{7,2} + m_{6,2} + m_{5,5}+$ $m_{5,2} + m_{4,4} + m_{3,31} + m_{3,4} + m_{3,2} + m_{3,1} + m_{2,4} + m_{2,3} + m_{0,3} = 0$
s_{79}	$a_{14,27}$	$m_{14,0} = 0$
s_{78}	$a_{13,26}$	$m_{13,31} = 0$
s_{77}	$a_{13,25}$	$m_{13,30} = 0$
s_{76}	$a_{14,29}$	$m_{13,29} + m_{8,29} = 0$
s_{75}	$a_{14,28}$	$m_{13,28} + m_{8,28} + m_{2,28} + m_{0,28} = 0$
s_{74}	$a_{13,22}$	$m_{13,27} + m_{11,28} + m_{8,29} + m_{8,27} + m_{6,29} + m_{5,28} + m_{3,28} + m_{2,27} + m_{0,27} = 1$
s_{73}	$a_{13,21}$	$m_{13,26} + m_{11,27} + m_{9,28} + m_{8,28} + m_{8,26} + m_{6,28} + m_{5,27} + m_{3,28} + m_{3,27}$ $+m_{2,26} + m_{1,28} + m_{0,26} = 1$
s_{72}	$a_{14,24}$	$m_{13,24} + m_{12,28} + m_{11,27} + m_{11,25} + m_{10,28} + m_{9,27} + m_{9,26} + m_{8,29} + m_{8,26}$ $+m_{8,24} + m_{7,29} + m_{7,28} + m_{6,26} + m_{5,25} + m_{4,28} + m_{3,28} + m_{3,26} + m_{3,25} + m_{2,28}$ $+m_{2,24} + m_{1,28} + m_{1,26} + m_{0,24} = 0$
s_{71}	$a_{14,23}$	$m_{13,23} + m_{12,27} + m_{11,26} + m_{11,24} + m_{10,28} + m_{10,27} + m_{9,26} + m_{9,25} + m_{8,29}+$ $m_{8,28} + m_{8,25} + m_{8,23} + m_{7,29} + m_{7,28} + m_{7,27} + m_{6,25} + m_{5,28} + m_{5,24} + m_{4,28}$ $+m_{4,27} + m_{3,27} + m_{3,25} + m_{3,24} + m_{2,27} + m_{2,23} + m_{1,27} + m_{1,25} + m_{0,28} + m_{0,23} = 0$
s_{70}	$a_{14,22}$	$m_{13,22} + m_{12,26} + m_{11,28} + m_{11,25} + m_{11,23} + m_{10,27} + m_{10,26} + m_{9,28} + m_{9,25}$ $+m_{9,24} + m_{8,28} + m_{8,27} + m_{8,24} + m_{8,22} + m_{7,28} + m_{7,27} + m_{7,26} + m_{6,29} + m_{6,24}$ $+m_{5,28} + m_{5,27} + m_{5,23} + m_{4,27} + m_{4,26} + m_{3,28} + m_{3,26} + m_{3,24} + m_{3,23} + m_{2,28}$ $+m_{2,26} + m_{2,22} + m_{1,26} + m_{1,24} + m_{0,28} + m_{0,27} + m_{0,22} = 1$
s_{69}	$a_{13,0}$	$m_{13,6} = 0$
s_{68}	$a_{14,5}$	$m_{13,5} + m_{12,5} + m_{5,5} + m_{4,5} + m_{2,5} = 0$
s_{67}	$a_{14,4}$	$m_{13,4} + m_{12,5} + m_{11,2} + m_{10,4} + m_{7,4} + m_{5,4} + m_{5,3} + m_{5,2} + m_{4,5} + m_{4,4} + m_{3,31}$ $+m_{2,5} + m_{2,4} + m_{2,2} + m_{1,2} = 0$
s_{66}	$a_{14,3}$	$m_{13,3} + m_{8,3} + m_{5,4} + m_{3,4} + m_{2,3} + m_{0,3} = 0$
s_{65}	$a_{13,28}$	$m_{13,2} + m_{10,3} + m_{10,2} + m_{10,1} + m_{9,2} + m_{8,2} + m_{7,4} + m_{7,2} + m_{4,0} + m_{3,4} + m_{3,3}$ $+m_{3,2} + m_{2,3} + m_{2,2} + m_{1,31} + m_{1,2} + m_{0,3} = 0$
s_{64}	$a_{14,1}$	$m_{13,1} + m_{10,2} + m_{9,3} + m_{8,3} + m_{7,4} + m_{7,2} + m_{6,2} + m_{5,3} + m_{5,2} + m_{4,0} + m_{3,4}$ $+m_{3,2} + m_{2,3} + m_{2,2} + m_{1,31} + m_{0,3} = 0$

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{63}	$a_{14,0}$	$m_{13,0} + m_{1,31} = 1$
s_{62}	$a_{12,26}$	$m_{12,31} = 1$
s_{61}	$a_{13,30}$	$m_{12,30} = 0$
s_{60}	$a_{12,24}$	$m_{12,29} = 1$
s_{59}	$a_{12,27}$	$m_{12,0} + m_{4,0} + m_{3,0} + m_{1,31} + m_{1,0} = 0$
s_{58}	$a_{11,26}$	$m_{11,31} = 1$
s_{57}	$a_{12,30}$	$m_{11,30} = 0$
s_{56}	$a_{11,24}$	$m_{11,29} = 1$
s_{55}	$a_{12,5}$	$m_{11,6} = 1$
s_{54}	$a_{11,0}$	$m_{11,6} = 1$
s_{53}	$a_{12,4}$	$m_{11,4} = 1$
s_{52}	$a_{12,1}$	$m_{11,1} = 1$
s_{51}	$a_{12,0}$	$m_{11,0} + m_{1,31} = 0$
s_{50}	$a_{10,26}$	$m_{10,31} = 0$
s_{49}	$a_{11,30}$	$m_{10,30} = 0$
s_{48}	$a_{10,24}$	$m_{10,29} = 0$
s_{47}	$a_{11,5}$	$m_{10,6} = 0$
s_{46}	$a_{10,0}$	$m_{10,5} + m_{4,5} + m_{2,5} = 0$
s_{45}	$a_{10,27}$	$m_{10,0} + m_{4,0} + m_{1,0} = 0$
s_{44}	$a_{9,26}$	$m_{9,31} + m_{3,31} + m_{3,0} + m_{1,0} = 1$
s_{43}	$a_{9,25}$	$m_{9,30} = 0$
s_{42}	$a_{10,30}$	$m_{9,30} = 0$
s_{41}	$a_{9,24}$	$m_{9,29} = 1$
s_{40}	$a_{9,0}$	$m_{9,6} = 0$
s_{39}	$a_{4,8}$	$m_{9,6} = 0$
s_{38}	$a_{10,5}$	$m_{9,5} + m_{8,5} + m_{6,5} + m_{3,5} = 0$
s_{37}	$a_{10,4}$	$m_{9,4} = 1$
s_{36}	$a_{9,28}$	$m_{9,1} = 1$
s_{35}	$a_{9,27}$	$m_{9,0} + m_{3,0} + m_{1,0} = 0$
s_{34}	$a_{8,26}$	$m_{8,31} = 0$
s_{33}	$a_{9,29}$	$m_{8,30} = 1$
s_{32}	$a_{8,28}$	$m_{8,1} = 0$
s_{31}	$a_{8,27}$	$m_{8,0} = 0$
s_{30}	$a_{8,31}$	$m_{7,31} + m_{3,31} + m_{1,31} + m_{1,0} = 0$
s_{29}	$a_{8,29}$	$m_{7,30} = 1$
s_{28}	$a_{8,4}$	$m_{7,5} = 1$
s_{27}	$a_{6,6}$	$m_{7,5} = 1$

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{26}	$a_{8,0}$	$m_{7,0} + m_{3,0} = 0$
s_{25}	$a_{7,31}$	$m_{6,31} = 0$
s_{24}	$a_{7,29}$	$m_{6,30} = 0$
s_{23}	$a_{3,26}$	$m_{5,31} + m_{3,31} = 0$
s_{22}	$a_{5,25}$	$m_{5,30} = 1$
s_{21}	$a_{6,29}$	$m_{5,29} = 1$
s_{20}	$a_{6,1}$	$m_{5,1} = 1$
s_{19}	$a_{3,27}$	$m_{5,0} + m_{3,0} + m_{1,31} = 1$
s_{18}	$a_{4,26}$	$m_{4,31} = 0$
s_{17}	$a_{4,25}$	$m_{4,30} = 0$
s_{16}	$a_{5,29}$	$m_{4,29} = 0$
s_{15}	$a_{5,6}$	$m_{4,6} = 0$
s_{14}	$a_{5,1}$	$m_{4,1} = 1$
s_{13}	$a_{3,25}$	$m_{3,30} = 1$
s_{12}	$a_{3,24}$	$m_{3,29} = 0$
s_{11}	$a_{4,6}$	$m_{3,6} = 1$
s_{10}	$a_{2,26}$	$m_{2,31} = 0$
s_9	$a_{2,25}$	$m_{2,30} = 1$
s_8	$a_{2,24}$	$m_{2,29} = 0$
s_7	$a_{3,5}$	$m_{2,6} = 1$
s_6	$a_{2,6}$	$m_{2,6} = 1$
s_5	$a_{3,1}$	$m_{2,1} = 1$
s_4	$a_{2,5}$	$m_{1,5} = 0$
s_3	$a_{1,28}$	$m_{1,1} = 1$
s_2	$a_{1,25}$	$m_{1,30} = 0$
s_1	$a_{1,24}$	$m_{1,29} = 1$
s_0	$a_{1,23}$	$m_{1,29} = 1$

Table 5: Control bit and controlled relations of 58-round SHA-1 (II)(III)(IV)

- 'r' means to adjust $a_{i,j}$ so that corresponding controlled relation including $m_{i,(j+27 \mod 32)}$ as leading term holds.
- 'x', 'y': adjust $a_{i+1,j-1}$, $a_{i,j-1}$ so that $m_{i,j} = 0$, respectively.
- 'X', 'Y': adjust $a_{i+1,j-1}$, $a_{i,j-1}$ so that $m_{i,j} = 1$, respectively.
- 'N': semi-neutral bit.
- 'q' : adjust $a_{i,j}$ so that relations after 17-round hold.

In this case, the set of bits corresponding to 'q' is exactly same to the set of *adjuster*s.

By using our advanced sufficient conditions on $\{a_{i,j}\}$ and Algorithm 1 which is used as Step 2 in Algorithm 2, we can adjust the value of $\{m_{i,j}\}_{i=0,1,\cdots,15;j=0,1,\cdots,31}$ according to the order defined as $m'_{i',j'} \leq m_{i,j}$ if $i' \leq i$ or ($i' = i$ and $j' \leq j$). By the proposed method we have succeeded in modifying message so that all sufficient conditions on message $\{m_{i,j}\}$ and some sufficient conditions on chaining variable $\{a_{ij}\}$ of first 23 rounds. Still 34 conditions remain as listed below: $a_{17,3} = 1, a_{17,2} = 0, a_{17,1} = 0, a_{26,1} = 1, a_{27,0} = 1, a_{29,1} = 0, a_{30,1} = 0, a_{33,1} = 1, a_{37,1} = 1, a_{39,1} = 0, a_{41,1} = 0, a_{43,1} = 0, a_{20,30} + a_{18,0} = 1, a_{21,30} + a_{20,0} = 0, a_{24,30} + a_{22,0} = 0, a_{25,30} + a_{24,0} = 1, a_{25,3} + a_{24,3} = 0, a_{26,2} + a_{25,2} = 1, a_{28,30} + a_{26,0} = 0, a_{28,3} + a_{27,3} = 1, a_{29,30} + a_{28,0} = 1, a_{29,3} + a_{28,3} = 1, a_{32,3} + a_{31,3} = 1, a_{36,3} + a_{35,3} = 1, a_{38,3} + a_{37,3} = 1, a_{39,31} + a_{38,1} = 1, a_{40,3} + a_{39,3} = 1, a_{40,31} + a_{38,1} = 1, a_{41,31} + a_{40,1} = 1, a_{42,31} + a_{40,1} = 1, a_{43,31} + a_{42,1} = 1, a_{42,3} + a_{41,3} = 1, a_{44,31} + a_{42,1} = 1, a_{45,31} + a_{44,1} = 1$.

Among the above conditions, there are five conditions $a_{17,3} = 1$, $a_{17,2} = 0$, $a_{17,1=0}$, $a_{20,30} + a_{18,0} = 1$, $a_{21,30} + a_{20,0} = 0$ which are related to only first 23 rounds. The probability that these five conditions are satisfied after the basic message modification (used in Step 2 of Algorithm 2) is $1/2^5$.

To adjust other 29 conditions, we use semi-neutral bits as we described in Algorithm 2.

6.4 New Collisions

Using Algorithm 2 (essentially, using semi-neutral bits showed in Table 6 to adjust the above remaining 29 conditions), we found many collisions of 58-round SHA-1 as follows. As we show in Table 6, we have 21 semi-neutral bits and 16 adusters.

Here we show some of new collisions we found. They are new collisions different from Wang's result. For other examples of new collisions, see [13].

$$
\begin{aligned}
m &= 0x1ead6636319fe59e4ea7ddcbc7961642\\
& 0ad9523af98f28db0ad135d0e4d62aec\\
& 6c2da52c3c7160b606ec74b2b02d545e\\
& bdd9e4663f1563194f497592dd1506f9\\
m' &= 0x3ead6636519fe5ac2ea7dd88e7961602\\
& ead95278998f28d98ad135d1e4d62acc\\
& 6c2da52f7c7160e446ec74f2502d540c\\
& 1dd9e466bf1563596f497593fd150699
\end{aligned}
$$

$$
\begin{aligned}
m &= 0x16507a963da18c5f4195d14bd55695ea\\
& 0cb08092f79649bb0717a22658c119fc\\
& 5a36c1f8b960383b08929187ae9842fa\\
& b690d8710452419d585d012edcaf0278\\
m' &= 0x36507a965da18c6d2195d108f55695aa\\
& ecb080d0979649b98717a22758c119dc\\
& 5a36c1fbf9603869489291c74e9842a8\\
& 1690d871845241dd785d012ffcaf0218
\end{aligned}
$$

6.5 Complexity

When we use the basic message modification which we described in Algorithm 1, the complexity to find a collision for 58-round SHA-1 is 2^{29} message modifications (equivalent to 2^{31} SHA-1 computation experimentally) because there 29 remaining conditions after message modifications, whereas Wang's method needs 2^{34} message modifications and 2^{34} SHA-1 computation.

Now we consider the complexity when we use the improved message modification proposed as Algorithm 2. Since there are 5 remaining conditions which should be tested in Step 3, the probability that the output of Step 2 pass the test of Step 3 is $1/2^5$. And since there are 29 remaining conditions after Step 3 and we have 21 semi-neutral bits, the probability that the modified message in Step 4 pass the final test of Step 4 is $1/2^8$. Hence when we use Algorithm 2, we have the complexity to find a collision for 58-round SHA-1 is 2^8 message modifications experimentally, because Step 4 is a dominant part of the algorithm. However, the real complexity to find a collision for 58-round SHA-1 in our latest implementation is 2^{31} SHA-1 computation, i.e. one improved message modification is 2^{23} heavier than the one of Algorithm 1. However, using sophisticated techniques of error correcting code (list decoding, iterative decoding, etc.) and Gröbner basis, it can be faster. Similarly, in the case of full-round SHA-1, we can use the same technique. The problem is that number of semi-neutral bits is much smaller than the case 58-round SHA-1. In this case one message modification is much heavier than the case of 58-round SHA-1. Implementation of such sophisticated technique is the future problem.

7 A concluding note

This paper yields an improved method for cryptanalysis of SHA-1 which originates from an explanation of the mathematical basis for Wang's attack and its improvement. We provide the detailed procedures which are based on a novel message modification technique. Particularly, via the computer experiments employing 58-round SHA-1 we have shown, by finding new collisions, that our algorithm is a very efficient one. The proposed method improves the complexity of finding collision for 58-round SHA-1 from 2^{34} SHA-1 computation to 2^{31}. The complexity can be reduced to 2^{8} message modification by using our improved message modification technique, even though complexity of one message modification appears as a high one implying a request for employment of the more sophisticated methods for error-correcting and Gröbner basis.

Acknowledgement The authors would like to thank Prof. Adi Shamir and Prof. Miodrag Mihaljevic for giving many useful comments on our manuscript.

References

[1] Hans Dobbertin, "Cryptanalysis of MD4." Fast Software Encryption 1996: 53-69

[2] X. Y. Wang etc, "An Attack on Hash Function HAVAL-128," Science in China Series E.

[3] L. C. K. Hui, X. Y. Wang etc, The Differential Analysis of Skipjack Variants from the first Round, Advance in Cryptography–CHINACRYPT'2002, Science Publishing House.

[4] Xiaoyun Wang, "Collisions for Some Hash Functions MD4, MD5,HAVAL-128,RIPEMD," Rump Session in Crypto'04, E-print.

[5] X. Y. Wang, "The Improved Collision attack on SHA-0", 1998.

[6] X. Y. Wang, "The Collision attack on SHA-0," 1997.

[7] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, Xiuyuan Yu "Cryptanalysis of the Hash Functions MD4 and RIPEMD." EUROCRYPT 2005: 1-18

[8] Xiaoyun Wang, Hongbo Yu: How to Break MD5 and Other Hash Functions. EUROCRYPT 2005: 19-35

[9] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, William Jalby "Collisions of SHA-0 and Reduced SHA-1." EUROCRYPT 2005: 36-57

[10] Eli Biham, Rafi Chen "Near-Collisions of SHA-0." CRYPTO 2004: 290-305

[11] Antoine Joux, "Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions." CRYPTO 2004: 306-316

[12] Florent Chabaud, Antoine Joux "Differential Collisions in SHA-0." CRYPTO 1998: 56-71

[13] M. Sugita, M. Kawazoe and H. Imai "Gröbner basis based cryptoanalysis of SHA-1", IACR Cryptology ePrint Archive 2006/098, http://eprint.iacr.org/2006/098,

[14] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin, "Efficient Collision Search Attacks on SHA-0," CRYPTO2005 1-16

[15] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, "Finding Collisions in the Full SHA-1," CRYPTO 2005, 17-36

Appendix: How to find good message differential

As we stated in Section 2, a choice of "differential without carry" is very important. Here we show how to find good "differential without carry" and good "message differential" with low Hamming weight.

Our strategy is as follows.

- Find a message differential in which difference appears only on continuing 4-bits. There are a few message differential patterns which have values only on 3- or 4- bits.
- Find another message-differentials of continuing 4-bit by shifting the one obtained in the previous step.
- Substitute message-differentials into each round and combine them (adding a disturbance vector) and obtain a 'better' message differential.

If we start from Wang's message-differential with continuing 4-bit, we have the results as in Fig. 1, Fig. 2. By our experiments, Wang's disturbance vector seems a best possible one.

i	Δ^+m	Δ^-m	i	Δ^+m	Δ^-m
0	20000000	0	29	2	40000040
1	40000020	20000012	30	40000002	40
2	20000000	40000043	31	2	40000000
3	20000000	40	32	0	2
4	e0000040	2	33	40	0
5	0	60000002	34	80000000	2
6	80000001	0	35	80000000	0
7	0	20	36	0	80000002
8	3	0	37	40	80000000
9	40000040	12	38	0	0
10	40000040	0	39	80000000	40
11	40000000	a0000052	40	0	80000000
12	0	a0000000	41	0	40
13	80000040	0	42	0	80000000
14	20000001	0	43	0	40
15	20000000	60	44	0	80000002
16	80000001	0	45	0	0
17	40000002	40	46	0	80000000
18	c0000002	41	47	80000000	0
19	40000000	22	48	0	0
20	0	3	49	0	0
21	40000040	2	50	0	0
22	80000041	40000002	51	0	0
23	2	c0000020	52	0	0
24	1	0	53	0	0
25	0	40000002	54	0	0
26	40000041	80000002	55	0	0
27	42	40000020	56	0	0
28	1	80000000	57	0	0

Table. A message-differential of continuous 4-round

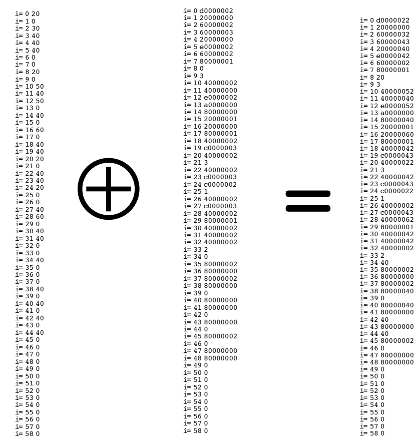

Figure 1: Finding good disturbance vector (I)

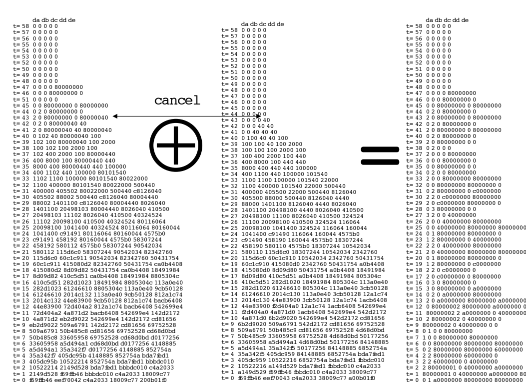

Figure 2: Finding good disturbance vector (II)