Algebraic Cryptanalysis of 58-round SHA-1

Makoto Sugita[1], Mitsuru Kawazoe[2], Ludovic Perret[3], and Hideki Imai[4]

[1] IT Security Center, Information-technology Promotion Agency, Japan
2-28-8 Honkomagome, Bunkyo-ku Tokyo, 113-6591, Japan
m-sugita@ipa.go.jp
[2] 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
[3] SPIRAL/SALSA
Site Passy-Kennedy
LIP6 – Paris 6 University
104 avenue du Président Kennedy
75016 Paris France
ludovic.perret@lip6.fr
[4] National Institute of 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

Abstract. In 2004, a new attack against SHA-1 has been proposed by a team leaded by Wang [15]. The aim of this article[5] is to sophisticate and improve Wang's attack by using algebraic techniques. We introduce new notions, namely semi-neutral bit and adjuster and propose then an improved message modification technique based on algebraic techniques. In the case of the 58-round SHA-1, the experimental complexity of our improved attack is 2^{31} SHA-1 computations, whereas Wang's method needs 2^{34} SHA-1 computations. We have found many new collisions for the 58-round SHA-1. We also study the complexity of our attack for the full SHA-1.

Keywords: SHA-1, Gröbner basis, differential attack

[5] A part of this work has been done when the third author was invited at Research Center for Information Security (RCIS) at Tokyo (Japan).

1 Introduction

Conceptually, we can split Wang's attack in four different steps.

Step 1. Choose a suitable 80×32-bit vector Γ.
Step 2. Choose a differential characteristic.
Step 3. Find a set of sufficient conditions on a message m and the chaining variables which guarantees with high probability that the message pair $(m, m+\Gamma)$ follows the differential characteristic. This implies that the two messages do collide.
Step 4. Choose a message m randomly and modify it until all sufficient conditions hold.

Using this method, Wang's team succeeded in finding collisions on the most popular hash functions, namely MD4, MD5, RIPEMD, SHA-0 and 58-round SHA-1 [11, 17, 15]. The attack is conceptually simple, but its implementation turns out to be very laborious in practice. To fill this gap between theory and practice, several teams decided to compensate their lack of intuition by the power of a computer, that is to say they tried to automatize the different steps of the attack.

For instance, coding theory can be used for finding a suitable difference in Step 1[8] of the attack. Recently, De Cannière and Rechberger [3] presented an algorithm allowing to find optimal differential characteristics in Step 2. The third step is tightly coupled with the previous one; most sufficient conditions follow from the choice of the differential characteristic.

We use algebraic techniques for actually finding collisions on 58-round SHA-1. In this case, the complexity of our method for finding a collision is equivalent to 2^{31} SHA-1 computations (experimentally), whereas Wang's method needs 2^{34} SHA-1 computations. As a proof of concept, we have found many new collisions for 58-round SHA-1, which have never been reported so far. We also apply our method for the case of the full SHA-1, and study the complexity of our approach.

The key idea is to describe the message modification technique into an algebraic framework. This is done by viewing the set of sufficient conditions as a non-linear system of Boolean equations. We hope that this will be a first step towards the use of algebraic tools (such as Gröbner bases) in the cryptanalysis of hash functions.

We will focus our attention on the last step of Wang's et al attack [15]. Namely, find a message satisfying a set of sufficient conditions depending on a disturbance vector and a differential path. This message can be then use to produce a collision. We shall call *conventional message modification* the process [15] permitting to construct such a suitable message. Here, we will present an improved message modification technique. To do so, we introduce the concepts *semi-neutral bits* and *adjusters*.

This paper is organized as follows. In Section 2, we give a description of SHA-1. Along the way, we introduce the notations and definitions that will be used throughout this paper. In Section 3, we describe our improved message modification technique. We explain how to use Gaussian elimination to construct a set controlled relations from the sufficient conditions. We also introduce a new notion, that we called *semi-neutral bit*, and describe then our improved message modification. We also give an algebraic descriptions of our improved message modification. This permits to give an interesting connection between the cryptanalysis of hash functions and the use of Gröbner bases. In Section 4, we present the details of our method on 58-round SHA-1. In the appendix, we provide the details for the full SHA-1.

2 Preliminaries

2.1 Description of SHA-1

The hash function SHA-1 generates a 160-bit hash value (or digest) from a message of length less than 2^{64} bits. The input message is padded and then processed in 512-bit message blocks through the Merkle/Damgard iterative structure.
A 80-step compression function is then applied to each of these 512-bit message blocks. It has two types of inputs: a chaining input of 160 bits and a message input of 512 bits. The initial chaining value (called IV) is a set of fixed constants, and the result of the last call to the compression function is the hash of the message.

In SHA-1, the message expansion is defined as follows: each 512-bit block of the padded message is divided into a 16×32-bit word $(m_0, m_1, \ldots, m_{15})$, and then expanded according to the following linear relation :

$$m_i \leftarrow (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}) \lll 1, \text{ for all } i, 16 \leq i \leq 79,$$

$x \lll n$, denoting the n-bit left rotation of a 32-bit word x. The compression function is defined for all $i, 1, \leq i \leq 80$ as follows:

$$\begin{aligned}
a_i &\leftarrow (a_{i-1} \lll 5) + f_i(b_{i-1}, c_{i-1}, d_{i-1}) + e_{i-1} + m_{i-1} + k_i \\
b_i &\leftarrow a_{i-1} \\
c_i &\leftarrow b_{i-1} \lll 30 \\
d_i &\leftarrow c_{i-1} \\
e_i &\leftarrow d_{i-1}
\end{aligned}$$

The initial chaining value IV=$(a_0, b_0, c_0, d_0, e_0)$ being equal to:

$$\big(0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1f0\big).$$

Note that we express as usual 32-bit words as hexadecimal numbers.
For $n, 58 \leq n \leq 80$, we call n-round SHA-1, the restriction of SHA-1 to the first n rounds. The Boolean function f_i and constant k_i employed at each step are defined as in Table 1.

Step	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: $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$	$0x8fabbcdc$
$61-80$	XOR: $x \oplus y \oplus z$	$0xca62c1d6$

Table 1. Definition of f_i and k_i w.r.t. the step

2.2 Definition and notation

We will identify the ring $\mathbb{Z}/2^{32}\mathbb{Z}$ with $\{0, 1, 2, \dots, 2^{32}-1\}$. If we ignore carry effects in the arithmetic of $\mathbb{Z}/2^{32}\mathbb{Z}$, we can identify the ring $\mathbb{Z}/2^{32}\mathbb{Z}$ with the vector space $\mathbb{F}_2^{32}$ by using the canonical bijective mapping ϱ:

$$\varrho : x_{31}2^{31} + x_{30}2^{30} + \dots + x_1 2^1 + x_0 2^0 \in \mathbb{Z}/2^{32}\mathbb{Z} \longmapsto (x_{31}, x_{30}, \dots, x_0) \in \mathbb{F}_2^{32}.$$

Here and in the rest of the paper, we try to find a collision between two messages $m = (m_0, m_1, \dots, m_{79})$ and $m' = (m'_0, m'_1, \dots, m'_{79})$ of $\left(\mathbb{F}_2^{32}\right)^{80}$. The corresponding chaining variables will be denoted by a_i, b_i, c_i, d_i, e_i and $a'_i, b'_i, c'_i, d'_i, e'_i$ respectively. For each $m_i = (m_{i,31}, m_{i,30}, \dots, m_{i,0})$ and $m'_i = (m'_{i,31}, m'_{i,30}, \dots, m'_{i,0})$, we define:

$$\begin{aligned} \Delta m_{i,j} &= m_{i,j} \oplus m'_{i,j} \in \mathbb{F}_2, \\ \Delta m_i &= m_i \oplus m'_i \in \mathbb{F}_2^{32}, \\ \delta m_i &= \varrho^{-1}(m_i) - \varrho^{-1}(m'_i) \in \mathbb{Z}/2^{32}\mathbb{Z}. \end{aligned}$$

Moreover, we set :

$$\Delta^+ m_{i,j} = \begin{cases} 1 & \text{if } (m_{i,j}, m'_{i,j}) = (0,1) \\ 0 & \text{otherwise,} \end{cases} \qquad \Delta^- m_{i,j} = \begin{cases} 1 & \text{if } (m_{i,j}, m'_{i,j}) = (1,0) \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\Delta^+ m_i = \left(\Delta^+ m_{i,31}, \dots, \Delta^+ m_{i,1}, \Delta^+ m_{i,0}\right), \quad \Delta^- m_i = \left(\Delta^- m_{i,31}, \dots, \Delta^- m_{i,1}, \Delta^- m_{i,0}\right).$$

Note that $\Delta m_i = \Delta^+ m_i \oplus \Delta^- m'_i \in \mathbb{F}_2^{32}$. Similarly, we define $\Delta, \Delta^+, \Delta^-, \delta$ for the chaining variables b_i, c_i, d_i and e_i (resp. b'_i, c'_i, d'_i and e'_i). Using the above definition, a *differential characteristic* and a *differential* are defined as follows.

Definition 1. *We call* **differential characteristic** *the sequence :*

$$\left(\Delta m_i, \Delta a_i, \Delta b_i, \Delta c_i, \Delta d_i, \Delta e_i\right)_{0 \le i \le 79},$$

and **differential** *:*

$$\left(\Delta^+ m_i, \Delta^- m_i, \Delta^+ a_i, \Delta^- a_i, \dots, \Delta^+ e_i, \Delta^- e_i\right)_{0 \le i \le 79}.$$

3 An Improved Message Modification Technique

Here, we will consider a n-round SHA-1, with $n, 58 \leq n \leq 80$. We will focus our attention on the last step of Wang's attack. Thus, we will suppose that a disturbance vector is fixed, as well as a suitable differential. We can then determine sufficient conditions on the messages permitting to produce collisions. Remark that sufficient conditions depend on the choice of a disturbance vector and its differential.

3.1 How to calculate sufficient conditions on the a_i ?

In this step, we only consider expanded messages by ignoring relations arising from the message expansion. We compute the sufficient conditions on chaining variables by adjusting b_i, c_i and d_i such that for all $i, 0 \leq i \leq n-1$:

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

In this calculation, we must adjust carry effects by "hand". It is indeed difficult to calculate this full-automatically. In Table 2 and Table 6, we present the sufficient conditions that we have obtained on the chaining variables for 58-round and the full SHA-1, respectively. Note that sufficient conditions on the messages are also quoted in this table.

3.2 Gaussian Elimination and Controlled Relations

To calculate sufficient conditions on the $\{m_{i,j}\}_{0 \leq i \leq n-1}^{0 \leq j \leq 31}$, we take into account that $\Delta^+ m_{i,j} = 1$ implies $m_{i,j} = 0$ and $\Delta^- m_{i,j} = 0$ implies $m_{i,j} = 1$. We also consider the relations derived from the key expansion :

$$m_{i,j} \leftarrow (m_{i-3,j} \oplus m_{i-8,j} \oplus m_{i-14,j} \oplus m_{i-16,j}) \lll 1, \text{ for all } i, 16 \leq i \leq n-1.$$

We shall call *controlled relations* a particular set of $\mathbb{F}_2$-linear equations on the $m_{i,j}$ on one hand, and on the chaining variables $a_{i,j}$ on the other. For the $m_{i,j}$, we consider the relations obtained by performing a Gaussian elimination on the linear equations defined by the key expansion, and the equations derived from the sufficient conditions. To perform this Gaussian elimination, we have considered the following order on the $m_{i,j}$:

$$m_{i',j'} < m_{i,j}, \text{ if } i' < i \text{ or } i' = i \text{ and } j' < j.$$

For $n = 58$, we obtain for instance the following controlled relations :

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

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

The controlled relations also include a subset of the sufficient conditions on the chaining variables. Precisely, we will only consider the conditions involving $a_{i,j}$, with $i \leq R$. The bound R is a positive integer that will be defined later. We will call *uncontrolled relations*, the sufficient conditions which are not a controlled relation.

We define now the notions of *semi-neutral bit*, *control bit* and *adjuster*. The concept of semi-neutral bit is closely related to Biham and Chen's "neutral bit" [2] and Klima's "tunnels" [22]. Namely, if the effect of flipping a bit corresponding to a chaining variable can be "easily" eliminated (i.e. such that all conditions previously satisfied can be satisfied by modifying few bits), then we shall call this bit a *semi-neutral bit*. Thus, the effect of changing a semi-neutral bit can be eliminated by controlling a little number of bits. We shall call these particular bits *adjusters*. Note that the choice of semi-neutral bits and adjusters is not unique. Thus, we have to choose it heuristically.

We emphasize that each $m_{i,j}$ can be viewed as a polynomial on the $a_{k,\ell}$'s, with $k \leq i+1$. Indeed, each $m_{i,j}$ can be viewed as a Boolean function on the $a_{k,\ell}$'s, with $k \leq i+1$, by the definition of SHA-1. Note that when we view $m_{i,j}$ as a Boolean function, we do not approximate (based on approximating MAJ by XOR, ignoring carry effect, etc.), but consider it as exact polynomial on the $a_{k,\ell}$. *Control bits* are determined for each controlled relation. Control bits are chosen among the $a_{k,\ell}$ which appear as a leading term or a term 'near' leading term in $m_{i,j}$, where $m_{i,j}$ is considered as a Boolean function on the $a_{k,\ell}$. The notion of leading term being related to a term ordering, we mention that we have considered here the following order on the $a_{k,\ell}$:

$$a_{k,\ell} < a_{k',\ell'} \text{ if } k' < k \text{ or } k' = k \text{ and } \ell' < \ell.$$

3.3 Conventional/Advanced Message Modification Techniques

The last step of Wang's attack consists of randomly choosing a message and modify some of its bits until all sufficient conditions are satisfied. To our knowledge, this technique has been described for the first time in [18, 19]. We shall call this method *conventional message modification* technique.

Here, we introduce an *improved message modification*. The conventional message modification will be used to obtain a "pre-collision", i.e. a collision from the first round to a given round R. This bound R will depend on the number n of rounds considered. We take $R = 23$ in the case of 58-round SHA-1 and $R = 26$ for the full SHA-1. The improved message modification will then allow to extend the pre-collision into a real collision on n-round SHA-1.

We would like to emphasize that our procedure will modify the chaining variable and not the message. Since IV=$(a_0, b_0, c_0, d_0, e_0)$ is fixed, it is clear that SHA-1 induces a bijection between $(m_0, m_1, \ldots, m_{15})$ and $(a_1, a_2, \ldots, a_{16})$. This implies that a modification on the $a_{i,j}$ can be mapped into a modification on the $m_{i,j}$.

Using our new terminology, we describe the conventional message modification. For this, we use a list of controlled relations C_R, and a list of control bits C_B.

Algorithm 1 Conventional Message Modification
CMM
Input : *A positive integer* R, *a list* C_R *of controlled relations, and a list* C_B *of control bits*
Output : $\mathbf{a} = (a_1, a_2, \ldots, a_{16}) \in \left(\mathbb{F}_2^{32}\right)^{16}$ *satisfying all the controlled relations*
Randomly choose $(a_1, a_2, \ldots, a_{16}) \in \left(\mathbb{F}_2^{32}\right)^{16}$
$\mathbf{a} \leftarrow (a_1, a_2, \ldots, a_{16})$
While *all the controlled relations* C_B *are not satisfied* **do**
Perform an exhaustive search on the bits $a_{i,j} \in C_B$
If *all the relations* C_B *are satisfied* **then** *return the updaded* $\mathbf{a}$
Else $\mathbf{a} \leftarrow$ *Randomly choose* $(a_1, a_2, \ldots, a_{16}) \in \left(\mathbb{F}_2^{32}\right)^{16}$

The CMM algorithm permits then to find a collision on R-round SHA-1. Using semi-neutral bits and adjusters, we present an improved algorithm permitting to find a collision on a n-round SHA-1 (with $n > R$). The new procedure is as follows.

Algorithm 2 Improved Message Modification
IMM
Input : *Positive integers* n, R, *two lists* (SNB, Ad) *of semi-neutral bits and adjusters, a list* C_R *of controlled relations, a list* C_B *of control bits, and a list* S_C *of sufficient conditions*
Output : $\mathbf{a} = (a_1, a_2, \ldots, a_{16}) \in \left(\mathbb{F}_2^{32}\right)^{16}$ *satisfying all the sufficient conditions*
$\mathbf{a} = (a_1, a_2, \ldots, a_{16}) \leftarrow$ *CMM*(R, C_R, C_B)
While *all the sufficient conditions of* S_C *are not satisfied* **do**
Adjust the $a_{i,j}$ *of* $\mathbf{a}$ *corresponding to a semi-neutral bit of* SNB *or an adjuster of* Ad
EndWhile
Return $\mathbf{a}$

Remark 1. We mention that a different version of the CMM and IMM algorithms can be found in [9, 10]. These versions could be more suitable for those wishing to actually implement these two algorithms.

We now analyze the complexity of the IMM algorithm for finding a collision on 58-round SHA-1. In this case, we choose $R = 23$, i.e. the CMM algorithm will be used to find a collision on 23-round SHA-1. It will remain 5 uncontrolled relations in rounds 17–23. Therefore, the CMM algorithm needs at most 2^5 iterations for returning a collision on 23-round SHA-1. There are 29 remaining conditions from rounds 23–58. To adjust these 29 conditions, we use 21 semi-neutral bits and 16 adjusters. Experimentally, the total complexity is improved to 2^{31} SHA-1 computation – with our latest implementation – whereas Wang's method needs theoretically 2^{34} SHA-1 computations. Note that the cost of the IMM algorithm is dominated by the exhaustive search among 21 semi-neutral bits, which means that we could neglect the cost of the CMM algorithm.
As a proof of concept, we give here a new collision on 58-round SHA-1.

$$\begin{aligned} m &= 0x1ead6636319fe59e4ea7ddcbc79616420ad9523af98f28db0ad135d0e4d62aec \\ &\quad 6c2da52c3c7160b606ec74b2b02d545ebdd9e4663f1563194f497592dd1506f9 \\ m' &= 0x3ead6636519fe5ac2ea7dd88e7961602ead95278998f28d98ad135d1e4d62acc \\ &\quad 6c2da52f7c7160e446ec74f2502d540c1dd9e466bf1563596f497593fd150699 \end{aligned}$$

3.4 An Algebraic Description of the Improved Message Modification

We present here an algebraic description of the IMM and CMM algorithms which could be useful for further improvements. For this, we remark that the CMM algorithm is equivalent to the solving of a polynomial system of equations via controlled relations with control bits as unknown variables. Similarly, the while-loop of the IMM algorithm is equivalent to the solving of an algebraic system of equations via sufficient conditions with semi-neutral bits and adjusters as unknown variables.

In other words, let $\mathbf{X} = \{X_{i,j}\}_{1 \le i \le n}^{0 \le j \le 31}$ and let $\mathbb{F}_2[\mathbf{X}]$ be the polynomial ring over $\mathbb{F}_2$ whose variables are $\mathbf{X}$. Remark that sufficient conditions can be considered as polynomial equations via Boolean functions. Thus, they can be expressed as algebraic polynomials on the $a_{i,j}$. Therefore – by replacing each $a_{i,j}$ by the variable $X_{i,j}$ – we can associate a set of polynomials on $\mathbb{F}_2[\mathbf{X}]$ to the set of sufficient conditions. With an obvious notation, we shall call *controlled polynomial* (resp. *uncontrolled polynomial*) the polynomial associated to a controlled relation (resp. uncontrolled relation).

Let J be an ideal in $\mathbb{F}_2[\mathbf{X}]$ generated by $\{X_{i,j}^2 + X_{i,j}\}_{1 \le i \le n}^{0 \le j \le 31}$, i.e. $J = \langle X_{i,j}^2 + X_{i,j}\rangle_{1 \le i \le n}^{0 \le j \le 31}$. Let then B_n be a quotient ring $\mathbb{F}_2[\mathbf{X}]/J$. Note that B_n represents the set of all Boolean functions on the variables $X_{i,j}$.

Let $\mathbf{f} = (f_1, f_2, \dots)$ be the set of controlled polynomials. Note that all controlled polynomials of $\mathbf{f}$ are in the subring $\mathbb{F}_2[\{X_{i,j}\}_{1 \le i \le R}^{0 \le j \le 31}]$, where R is determined by n (for instance, $R = 23$ when $n = 58$ and $R = 26$ when $n = 80$).
For a randomly taken $\mathbf{a} = (a_1, a_2, \dots, a_{16}) \in (\mathbb{F}_2^{32})^{16}$, let $C_R(\mathbf{a})$ be the system

obtain from the sufficient conditions by replacing each variables $X_{i,j}$ – not corresponding to a control bit – by $a_{i,j}$. Similarly, let $S_C(\mathbf{a})$ be the system obtain from the sufficient conditions by replacing each variables $X_{i,j}$ – not corresponding to a semi-neutral bit or adjuster – by $a_{i,j}$. In this setting, the CMM and IMM algorithms can roughly be described as follows:

Algorithm 3 *Algebraic Message Modification*
Randomly choose $(a_1, a_2, \ldots, a_{16}) \in \left(\mathbb{F}_2^{32}\right)^{16}$
$\mathbf{a} \leftarrow (a_1, a_2, \ldots, a_{16})$
Solve the algebraic system of equations $C_R(\mathbf{a})$*:* `The solutions correspond to affectations of control bits verifying all controlled polynomials`
Solve the algebraic system of equations $S_C(\mathbf{a})$*:* `The solutions correspond to affectations of semi-neutral bits and adjusters verifying all uncontrolled polynomials`
Update $\mathbf{a}$ *according to the solutions of the two previous systems*
Return $\mathbf{a}$

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 conventional 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.

4 Analysis of the 58-round SHA-1 using the Improved Message Modification

In this part, we detail the different steps of our technique for finding a collision on 58-round SHA-1. In Table 2, we give the sufficient conditions. Note that we have only quoted the sufficient conditions for the first 20 rounds due to space limitation. For the complete list of the conditions, see [9, 10]. The control bits and controlled relations are given in Table 3. Semi-neutral bits and adjusters are given in Table 4.

- '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 $a_{i,j} = a_{i-2,(j+2 \mod 32)}$
- 'C' means $a_{i,j} = a_{i-2,(j+2 \mod 32)} + 1$
- 'L' means the leading term of controlled relation of Table 3

– 'w', 'W': adjust $a_{i,j}$ so that $m_{i+1,j} = 0, 1$, respectively
– 'v', 'V': adjust $a_{i,j}$ so that $m_{i,(j+27 \mod 32)} = 0, 1$, respectively
– 'h': adjust $a_{i,j}$ so that corresponding controlled relation including $m_{i+1,j}$ as leading term holds
– '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.

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

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---

Table 2. Sufficient condition on the $m_{i,j}$ (resp. $a_{i,j}$)

The conditions remaining after the conventional message modification are listed below: $a_{17,3} = 1, a_{17,2} = 0, a_{17,1} = 0, a_{26,1} = 1, a_{27,0} = 1, a_{29,1} = 0, a_{30,1} = 0, a_{33,1} = 1, a_{37,1} = 1, a_{39,1} = 0, a_{41,1} = 0, a_{43,1} = 0, a_{20,30} + a_{18,0} = 1, a_{21,30} + a_{20,0} = 0, a_{24,30} + a_{22,0} = 0, a_{25,30} + a_{24,0} = 1, a_{25,3} + a_{24,3} = 0, a_{26,2} + a_{25,2} = 1, a_{28,30} + a_{26,0} = 0, a_{28,3} + a_{27,3} = 1, a_{29,30} + a_{28,0} = 1, a_{29,3} + a_{28,3} = 1, a_{32,3} + a_{31,3} = 1, a_{36,3} + a_{35,3} = 1, a_{38,3} + a_{37,3} = 1, a_{39,31} + a_{38,1} = 1, a_{40,3} + a_{39,3} = 1, a_{40,31} + a_{38,1} = 1, a_{41,31} + a_{40,1} = 1, a_{42,31} + a_{40,1} = 1, a_{43,31} + a_{42,1} = 1, a_{42,3} + a_{41,3} = 1, a_{44,31} + a_{42,1} = 1, a_{45,31} + a_{44,1} = 1$.
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 only related to first 23 rounds. For other 29 conditions, we adjust by using 21 semi-neutral bits and 11 adjusters as explained in the improved message modification algorithm (see Algorithm 2).

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}	$a_{16,4}$	$m_{15,4} + m_{12,5} + m_{10,4} + m_{4,5} + m_{4,4} + m_{2,5} + m_{2,4} = 1$
$\vdots$		
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 3. Control bits and controlled relations

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 4. Semi-neutral bits and adjusters

5 A concluding Note

This paper present an improved method for finding collision on SHA-1. To do so, we use algebraic techniques for describing the message modification technique and propose an improvement. The details of our attack can be found in the appendices. The proposed method improves the complexity of an attack against 58-round SHA-1 and we found many new collisions.

References

1. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet, W. Jalby, *Collisions of SHA-0 and Reduced SHA-1.* Advances in Cryptology – EUROCRYPT 2005, LNCS, vol. 3494, pp. 36–57, Springer–Verlag, 2005.
2. E. Biham, R. Chen. *Near-Collisions of SHA-0.* Advances in Cryptology – CRYPTO 2004, LNCS vol. 3152, pp. 290-305, Springer–Verlag, 2004.
3. C. De Cannière, C. Rechberger. *Finding SHA-1 Characteristics.* Advances in Cryptology – ASIACRYPT 2006, LNCS vol. 4284, pp. 1–20, Springer–Verlag, 2006.
4. F. Chabaud, A. Joux. *Differential Collisions in SHA-0.* Advances in Cryptology – CRYPTO '98, LNCS vol. 1462, pp. 56–71, Springer–Verlag, 1998.
5. H. Dobbertin. *Cryptanalysis of MD4.* Fast Software Encryption, Third International Workshop, LNCS vol. 1039, pp. 53–69, Springer–Verlag, 1996.
6. L. C. K. Hui, X. Wang, K. P. Chow, W. W. Tsang, C. F. Chong, H. W. Chan. *The Differential Analysis of Skipjack Variants from the first Round.* Advance in Cryptography – CHINACRYPT2002, Science Publishing House.
7. A. Joux. *Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions.* Advances in Cryptology – CRYPTO 2004, LNCS vol. 3152, pp. 306–316, Springer-Verlag, 2004.
8. N. Pramstaller, C. Rechberger, V. Rijmen. *Exploiting Coding Theory for Collision Attacks on SHA-1.* Proceedings of 10th IMA International Conference on Cryptography and Coding, Lecture Notes in Computer Science, vol. 3796, pp. 78-95, Springer-Verlag, 2005.
9. M. Sugita, M. Kawazoe, H. Imai. *Gröbner Basis Based Cryptanalysis of SHA-1.* IACR Cryptology ePrint Archive 2006/098, http://eprint.iacr.org/2006/098.pdf, 2006
10. M. Sugita, M. Kawazoe, H. Imai. *Gröbner Basis Based Cryptanalysis of SHA-1.* In Proc. of SECOND NIST CRYPTOGRAPHIC HASH WORKSHOP, 2006.
11. X. Wang, X. Lai, D. Feng, H. Chen, X. Yu. *Cryptanalysis of the Hash Functions MD4 and RIPEMD.* Advances in Cryptology – EUROCRYPT 2005, LNCS vol. 3494, pp. 1–18, Springer-Verlag, 2005.
12. X. Wang, D. Feng, X. Yu. *An Attack on Hash Function HAVAL-128.* Science in China Series F **48**(2005), 545–556.
13. X. Wang, A. C. Yao, F. Yao. *Cryptanalysis on SHA-1.* In Proc. of NIST Cryptographic Hash Workshop, 2005.
14. X. Wang, Y. L. Yin, H. Yu. *New Collision Search for SHA-1.* Rump Session of CRYPTO 2005.
15. X. Wang, Y. L. Yin, H. Yu. *Finding Collisions in the Full SHA-1.* Advances in Cryptology - CRYPTO 2005, LNCS vol. 3621, pp. 17–36, Springer-Verlag, 2005.

16. X. Wang, H. Yu. *How to Break MD5 and Other Hash Functions.* Advances in Cryptology – EUROCRYPT 2005, LNCS vol. 3494, pp. 19–35, Springer-Verlag, 2005.
17. X. Wang, H. Yu, Y. L. Yin. *Efficient Collision Search Attacks on SHA-0.* Advances in Cryptology – CRYPTO 2005, LNCS vol. 3621, pp. 1-16, Springer-Verlag, 2005.
18. X. Wang. *The Collision attack on SHA-0*, 1997.
19. X. Wang. *The Improved Collision attack on SHA-0*, 1998.
20. X. Wang. *Collisions for Some Hash Functions MD4, MD5, HAVAL-128, RIPEMD.* Rump Session of Crypto'04.
21. X. Wang. *Cryptanalysis of Hash Functions and Potential Dangers.* RSA Conference 2006, San Jose, USA.
22. V. Klima. *Tunnels in Hash Functions: MD5 Collisions Within a Minute.* IACR Cryptology ePrint Archive 2006/105, http://eprint.iacr.org/2006/098.pdf.

Appendix

A Analysis of the full SHA-1

We present here the details of our method on the full SHA-1.

A.1 Disturbance vector and Differential

We start from the disturbance vector and differential given by Wang et al [13]. From these, we construct a new differential. We modify Wang's differential by changing $\Delta a^{+}_{14,31}$ and $\Delta a^{-}_{14,31}$. For our method, it is important to choose $\Delta^{+}m$, $\Delta^{-}m$, $\Delta^{+}a$ and $\Delta^{-}a$ carefully. A bad choice of plus/minus could lead to an underdefined system of equations (as result of the Gaussian elimination). For this reason, we have to correct Wang's differential. We present in Table 5 the new differential constructed.

	Δm	$\Delta^{+}m$	$\Delta^{-}m$
$i=0$	a0000003	00000001	a0000002
$i=1$	20000030	20000020	00000010
$i=2$	60000000	60000000	00000000
$i=3$	e000002a	40000000	a000002a
$i=4$	20000043	20000042	00000001
$i=5$	b0000040	a0000000	10000040
$i=6$	d0000053	d0000042	00000011
$i=7$	d0000022	d0000000	00000022
$i=8$	20000000	00000000	20000000
$i=9$	60000032	20000030	40000002
$i=10$	60000043	60000041	00000002
$i=11$	20000040	00000000	20000040
$i=12$	e0000042	c0000000	20000042
$i=13$	60000002	00000002	60000000
$i=14$	80000001	00000001	80000000
$i=15$	00000020	00000020	00000000
$i=16$	00000003	00000002	00000001
$i=17$	40000052	00000002	40000050
$i=18$	40000040	00000000	40000040
$i=19$	e0000052	00000002	e0000050
$i=20$	a0000000	00000000	a0000000
⋮			
$i=78$	0000004a	00000002	00000048
$i=79$	0000080a	00000808	00000002
$i=80$	00000000	00000000	00000000

	Δa	$\Delta^{+}a$	$\Delta^{-}a$
$i=0$	00000000	00000000	00000000
$i=1$	e0000001	a0000000	40000001
$i=2$	20000004	20000000	00000004
$i=3$	c07fff84	803fff84	40400000
$i=4$	800030e2	800010a0	00002042
$i=5$	084080b0	08008020	00400090
$i=6$	80003a00	00001a00	80002000
$i=7$	0fff8001	08000001	07ff8000
$i=8$	00000008	00000008	00000000
$i=9$	80000101	80000100	00000001
$i=10$	00000002	00000002	00000000
$i=11$	00000100	00000000	00000100
$i=12$	00000002	00000002	00000000
$i=13$	00000000	00000000	00000000
$i=14$	00000000	00000000	00000000
$i=15$	00000001	00000001	00000000
$i=16$	00000000	00000000	00000000
$i=17$	80000002	80000002	00000000
$i=18$	00000002	00000002	00000000
$i=19$	80000002	80000002	00000000
$i=20$	00000000	00000000	00000000
⋮			
$i=78$	00000000	00000000	00000000
$i=79$	00000040	00000000	00000040
$i=80$	00000000	00000000	00000000

Table 5. A differential for the full SHA-1

A.2 Sufficient Conditions

For the disturbance vector, and the differential given in the previous step, we give in Table 6 the sufficient conditions for the full SHA-1. In Table 6: '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 $a_{i,j} = a_{i-2,(j+2 \mod 32)} + 1$.

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	1-1-----	--------	--------	------10
m_1	--0-----	--------	--------	--01----
m_2	-00-----	--------	--------	--------
m_3	101-----	--------	--------	--1-1-1-
m_4	--0-----	--------	--------	-0----01
m_5	0-01----	--------	--------	-1------
m_6	00-0----	--------	--------	-0-1--01
m_7	00-0----	--------	--------	--1---1-
m_8	--1-----	--------	--------	--------
m_9	-10-----	--------	--------	--00--1-
m_{10}	-00-----	--------	--------	-0----10
m_{11}	--1-----	--------	--------	-1------
m_{12}	001-----	--------	--------	-1----1-
m_{13}	-11-----	--------	--------	------0-
m_{14}	1-------	--------	--------	-------0
m_{15}	--------	--------	--------	--0-----
m_{16}	--------	--------	--------	------01
m_{17}	-1------	--------	--------	-1-1--0-
m_{18}	-1------	--------	--------	-1------
m_{19}	111-----	--------	--------	-1-1--0-
m_{20}	1-1-----	--------	--------	--------
⋮				
m_{78}	--------	--------	--------	-1--1-0-
m_{79}	--------	--------	----0---	----0-1-
m_{80}	--------	--------	--------	--------

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	010----0	-0-01-0-	10-0-10-	---a0101
a_2	-100---1	0aa10a1a	01a1a011	1--a11a1
a_3	01011---	-1000000	00000000	01--a0a1
a_4	0-101--a	---10000	00101000	010---10
a_5	0-0101-1	-1-11110	00111-00	10010100
a_6	1-0a1a0a	a0a1aaa-	--10010-	--01-0--
a_7	--0-0111	11111111	111-010-	0-0-0110
a_8	-10---01	11110000	010-111-	1---000-
a_9	00----11	11111111	111----0	----1-01
a_{10}	-11-----	--------	-----a--	-1--1-0-
a_{11}	100-----	--------	-------1	-1--0---
a_{12}	--------	--------	--------	-1----0-
a_{13}	0-------	--------	--------	-1---0--
a_{14}	1-------	--------	--------	-----1--
a_{15}	--------	--------	--------	----0--0
a_{16}	-1------	--------	--------	----1-A-
a_{17}	00------	--------	--------	----0-0-
a_{18}	1-1-----	--------	--------	----a-0-
a_{19}	0-b-----	--------	--------	------0-
a_{20}	--0-----	--------	--------	----a---
⋮				
a_{78}	--------	--------	--------	----b---
a_{79}	--------	--------	--------	-1------
a_{80}	--------	--------	--------	--------

Table 6. Sufficient conditions for the full SHA-1

By a Gaussian elimination, we reduce all conditions on the m_i. The result of Gaussian elimination is listed below. There are 167 conditions on $m_{i,j}$:

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

$\cdots$,

$m_{5,0} + m_{1,30} + m_{1,0} = 0$, $m_{4,31} = 0$, $m_{4,30} = 0$, $m_{4,29} = 0$, $m_{4,6} = 0$, $m_{4,1} = 0$, $m_{4,0} = 1$, $m_{3,31} = 1$, $m_{3,30} = 0$, $m_{3,29} = 1$, $m_{3,5} = 1$, $m_{3,3} = 1$, $m_{3,1} = 1$, $m_{3,0} + m_{1,0} = 0$, $m_{2,31} = 1$, $m_{2,30} = 0$, $m_{2,29} = 0$, $m_{2,0} + m_{1,30} = 1$, $m_{1,31} + m_{1,30} = 1$, $m_{1,29} = 0$, $m_{1,5} = 0$, $m_{1,4} = 1$, $m_{0,31} = 1$, $m_{0,30} = 0$, $m_{0,29} = 1$, $m_{0,1} = 1$, $m_{0,0} = 0$.

A.3 Control bits and Controlled Relations

The control bits and controlled relations are presented in Table 7.

ctrl. seq.	control bits	controlled relation
s_{168}	$a_{15,8}$	$a_{30,2} + a_{29,2} = 1$
s_{167}	$a_{16,6}$	$a_{26,2} + a_{25,2} = 1$
s_{166}	$a_{15,7}$	$a_{25,3} + a_{24,3} = 0$
s_{165}	$a_{13,7}$	$a_{24,3} + a_{23,3} = 0$
s_{164}	$a_{13,9}$	$a_{23,0} = 0$
s_{163}	$a_{16,10}$	$a_{22,3} + a_{21,3} = 0$
s_{162}	$a_{16,11}$	$a_{21,29} + a_{20,31} = 0$
s_{161}	$a_{16,8}$	$a_{21,1} = 0$
s_{160}	$a_{16,9}$	$a_{20,29} = 0$
s_{159}	$a_{15,10}$	$a_{20,3} + a_{19,3} = 0$
s_{158}	$a_{15,11}$	$a_{19,31} = 0$
s_{157}	$a_{15,9}$	$a_{19,29} + a_{18,31} = 0$
s_{156}	$a_{14,8}$	$a_{19,1} = 0$
s_{155}	$a_{14,11}$	$a_{18,31} = 1$
s_{154}	$a_{15,14}$	$a_{18,29} = 1$
s_{153}	$a_{13,8}$	$a_{18,1} = 0$
s_{152}	$a_{13,11}$	$a_{17,31} = 0$
s_{151}	$a_{13,10}$	$a_{17,30} = 0$
s_{150}	$a_{13,13}$	$a_{17,1} = 0$
s_{149}	$a_{16,31}$	$m_{15,31} = 0$
s_{148}	$a_{16,29}$	$m_{15,29} = 1$
s_{147}	$a_{16,28}$	$m_{15,28} + m_{10,28} + m_{4,28} + m_{2,28} = 0$
s_{146}	$a_{16,27}$	$m_{15,27} + m_{10,27} + m_{8,28} + m_{4,27} + m_{2,28} + m_{2,27} + m_{0,28} = 1$
s_{145}	$a_{16,26}$	$m_{15,26} + m_{10,28} + m_{10,26} + m_{8,28} + m_{8,27} + m_{7,27} + m_{5,27} + m_{4,26} + m_{2,27} + m_{2,26} + m_{0,27} = 0$
s_{144}	$a_{16,25}$	$m_{15,25} + m_{11,28} + m_{10,27} + m_{10,25} + m_{9,28} + m_{8,27} + m_{8,26} + m_{7,26} + m_{5,26} + m_{4,25} + m_{3,28} + m_{2,28} + m_{2,26} + m_{2,25} + m_{1,28} + m_{0,28} + m_{0,26} = 0$
s_{143}	$a_{16,24}$	$m_{15,24} + m_{12,28} + m_{11,27} + m_{10,26} + m_{10,24} + m_{9,28} + m_{9,27} + m_{8,26} + m_{8,25} + m_{7,25} + m_{6,27} + m_{5,25} + m_{4,28} + m_{4,24} + m_{3,28} + m_{3,27} + m_{2,27} + m_{2,25} + m_{2,24} + m_{1,28} + m_{1,27} + m_{0,27} + m_{0,25} = 1$
s_{142}	$a_{16,23}$	$m_{15,23} + m_{12,28} + m_{12,27} + m_{11,26} + m_{10,25} + m_{10,23} + m_{9,27} + m_{9,26} + m_{8,28} + m_{8,25} + m_{8,24} + m_{7,24} + m_{7,0} + m_{6,27} + m_{6,26} + m_{5,24} + m_{4,27} + m_{4,23} + m_{3,27} + m_{3,26} + m_{2,26} + m_{2,24} + m_{2,23} + m_{1,30} + m_{1,27} + m_{1,26} + m_{1,0} + m_{0,26} + m_{0,24} = 0$
$\vdots$	$\vdots$	$\vdots$
s_3	$a_{2,5}$	$m_{1,5} = 0$
s_2	$a_{1,26}$	$m_{1,31} + m_{1,30} = 1$
s_1	$a_{1,23}$	$m_{1,29} = 0$

Table 7. Control bits and controlled relations for the full SHA-1

Now we give the semi-neutral bits and adjuster in Table 8. In this table :

– 'a', 'A', 'b', 'B', 'c', 'C': as in Section A.2.
– 'L' means the leading term of controlled relation of Table 7.
– 'w', 'W': adjust $a_{i,j}$ so that $m_{i+1,j} = 0, 1$, respectively.
– 'v', 'V': adjust $a_{i,j}$ so that $m_{i,(j+27 \mod 32)} = 0, 1$, respectively.
– 'h': adjust $a_{i,j}$ so that corresponding controlled relation including $m_{i+1,j}$ as leading term holds.
– '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.
– 'F' : etc.

For the full SHA-1, the CMM algorithm will be used to find a collision on 26-round SHA-1. After the conventional message modification, it will remain the 9 following conditions; which are only related to rounds 17-26:
$a_{17,3} = 0$, $a_{18,3}+a_{17,3} = 0$, $a_{22,2}+a_{21,2} = 0$, $a_{23,1} = 0$, $a_{24,30}+a_{22,0} = 0$, $a_{24,3}+a_{23,3} = 0$, $a_{25,30} + a_{24,0} = 1$, $a_{25,1} = 0$, $a_{26,1} = 1$.

Uncontrolled Relations. There is 64 uncontrolled relations on the $a_{i,j}$:
$a_{27,0} = 0$, $a_{28,30} + a_{26,0} = 1$, $a_{28,3} + a_{27,3} = 0$, $a_{29,30} + a_{28,0} = 0$, $a_{29,3} + a_{28,3} = 0$, $a_{29,1} = 1$, $a_{30,1} = 0$, $a_{31,0} = 1$, $a_{32,30} + a_{30,0} = 0$, $a_{33,30} + a_{32,0} = 0$, $a_{33,3} + a_{32,3} = 0$, $a_{34,3} + a_{33,3} = 1$, $a_{34,2} + a_{33,2} = 1$, $a_{34,1} = 0$, $a_{35,1} = 0$, $a_{35,0} = 0$, $a_{36,30} + a_{34,0} = 0$, $a_{36,3}+a_{35,3} = 1$, $a_{37,30}+a_{36,0} = 1$, $a_{37,3}+a_{36,3} = 1$, $a_{37,1} = 1$, $a_{38,1} = 0$, $a_{40,31}+a_{39,1} = 1$, $a_{40,3} + a_{39,3} = 1$, $a_{41,1} = 1$, $a_{42,31} + a_{40,1} = 1$, $a_{43,31} + a_{42,1} = 1$, $a_{44,3} + a_{43,3} = 1$, $a_{45,1} = 0$, $a_{46,31} + a_{44,1} = 1$, $a_{46,3} + a_{45,3} = 1$, $a_{47,31} + a_{46,1} = 1$, $a_{47,1} = 0$, $a_{48,31} + a_{46,1} = 1$, $a_{48,3} + a_{47,3} = 1$, $a_{49,31} + a_{48,1} = 1$, $a_{49,1} = 0$, $a_{50,31} + a_{48,1} = 1$, $a_{50,3} + a_{49,3} = 1$, $a_{51,31} + a_{50,1} = 1$, $a_{51,1} = 0$, $a_{52,31} + a_{50,1} = 1$, $a_{53,31} + a_{52,1} = 1$, $a_{66,4} + a_{65,4} = 1$, $a_{67,2} = 0$, $a_{68,0} + a_{66,2} = 1$, $a_{69,5} + a_{68,5} = 1$, $a_{69,0} + a_{68,2} = 1$, $a_{70,3} = 0$, $a_{71,1}+a_{69,3} = 1$, $a_{72,6}+a_{71,6} = 1$, $a_{72,1}+a_{71,3} = 1$, $a_{73,4} = 0$, $a_{74,5}+a_{73,5} = 1$, $a_{74,2}+a_{72,4} = 1$, $a_{75,7}+a_{74,7} = 1$, $a_{75,3} = 0$, $a_{75,2}+a_{74,4} = 1$, $a_{76,5} = 1$, $a_{76,1}+a_{74,3} = 1$, $a_{77,3} + a_{75,5} = 1$, $a_{77,1} + a_{76,3} = 1$, $a_{78,3} + a_{77,5} = 0$, $a_{79,6} = 1$.

To adjust these 64 conditions, we have tried to use semi-neutral bits and adjusters as explained in the IMM algorithm. We use 10 semi-neutral bits (corresponding to 'N' in Table 8) and 8 *adjuster*s which are the bits 1-bit-left to '`N`'. We present an example of message $m = (m_0, m_1, \ldots, m_{15})$ obtained by Algorithm 2.

$$\begin{aligned} m = {} & b8550bb2\ 5b2e1a15\ 88a0e568\ b0d7cbaf\ 0b430105\ 1e7f1b5e\ 0637da31\ 0dc9d562 \\ & 7d857448\ defac00e\ 9d06ba9e\ 2dd8235a\ 324e9acb\ f7c56578\ c69dfd0e\ 71bf1d08 \end{aligned}$$

The above m satisfies all message conditions of 0-80 rounds and all chaining variable conditions of 0-28 rounds.

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	1-1-----	--------	--------	------10
m_1	L-0-----	--------	--------	--01----
m_2	L00-----	--------	--------	-------L
m_3	101-----	--------	--------	--1-1-1L
m_4	LL0-----	--------	--------	-0----01
m_5	0L01----	--------	--------	-1-----L
m_6	00L0----	--------	--------	-0-1--01
m_7	00-0----	--------	--------	--1L--1-
m_8	L-1-----	--------	--------	----L--L
m_9	L10-----	--------	--------	--00-L1L
m_{10}	L00-----	--------	--------	-0LLLL10
m_{11}	LL1-----	--------	--------	-1LLLLLL
m_{12}	001-----	--------	--------	-1LLL-1L
m_{13}	L11LLLLL	LLLLLLLL	L-L-----	--LLLL0L
m_{14}	1LLLLLLL	LLLLLLLL	L-LL----	--LLLLL0
m_{15}	LLLLLLLL	LLLLLLLL	LL-L----	L-0LLLLL
m_{16}	--------	--------	--------	------01
m_{17}	-1------	--------	--------	-1-1--0-
m_{18}	-1------	--------	--------	-1------
m_{19}	111-----	--------	--------	-1-1--0-
m_{20}	1-1-----	--------	--------	--------
m_{21}	0-------	--------	--------	-1------
m_{22}	--1-----	--------	--------	-------0
m_{23}	--1-----	--------	--------	-11-----
m_{24}	1-------	--------	--------	-------1
m_{25}	-1------	--------	--------	-1----1-
m_{26}	10------	--------	--------	-0----10
m_{27}	-1------	--------	--------	--1---0-
m_{28}	--------	--------	--------	------10
m_{29}	-0------	--------	--------	-0----0-
m_{30}	11------	--------	--------	-1----01
m_{31}	11------	--------	--------	--0---1-
m_{32}	--------	--------	--------	-------1
m_{33}	-0------	--------	--------	------0-
m_{34}	00------	--------	--------	-1----00
m_{35}	-0------	--------	--------	-11---1-
m_{36}	1-------	--------	--------	-------0
m_{37}	-1------	--------	--------	-0----0-
m_{38}	-0------	--------	--------	-1----1-
m_{39}	-1------	--------	--------	------0-
m_{40}	--------	--------	--------	------1-
m_{41}	--------	--------	--------	-0------
m_{42}	0-------	--------	--------	------0-
m_{43}	1-------	--------	--------	--------
m_{44}	0-------	--------	--------	------0-
m_{45}	0-------	--------	--------	-1------
m_{46}	--------	--------	--------	--------
m_{47}	0-------	--------	--------	-1------
m_{48}	0-------	--------	--------	--------
m_{49}	--------	--------	--------	-1------
m_{50}	0-------	--------	--------	--------
m_{51}	--------	--------	--------	-1------
m_{52}	1-------	--------	--------	------1-
m_{53}	--------	--------	--------	--------
m_{54}	1-------	--------	--------	--------
m_{55}	0-------	--------	--------	--------
m_{56}	--------	--------	--------	--------
m_{57}	--------	--------	--------	--------
m_{58}	--------	--------	--------	--------
m_{59}	--------	--------	--------	--------
m_{60}	--------	--------	--------	--------
m_{61}	--------	--------	--------	--------
m_{62}	--------	--------	--------	--------
m_{63}	--------	--------	--------	--------
m_{64}	--------	--------	--------	--------
m_{65}	--------	--------	--------	--------
m_{66}	--------	--------	--------	-----0--
m_{67}	--------	--------	--------	1-------
m_{68}	--------	--------	--------	-----0--
m_{69}	--------	--------	--------	----0--0
m_{70}	--------	--------	-------1	-------0
m_{71}	--------	--------	--------	----0--1
m_{72}	--------	--------	--------	----0-0-
m_{73}	--------	--------	------1-	------0-
m_{74}	--------	--------	--------	---00-1-
m_{75}	--------	--------	-------1	--1--0--
m_{76}	--------	--------	-----0--	----00--
m_{77}	--------	--------	--------	--1--10-
m_{78}	--------	--------	--------	-1--1-0-
m_{79}	--------	--------	----0---	----0-1-
m_{80}	--------	--------	--------	--------

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	010-FrF0	y0-01-0-	10-0-10-	F-Fa0101
a_2	F100-Vv1	0aa10a1a	01a1a011	1-wa11a1
a_3	01011VFV	-1000000	00000000	01FFa0a1
a_4	0w101v-a	y--10000	00101000	010XWF10
a_5	0w0101y1	V1-11110	00111-00	10010100
a_6	1w0a1a0a	a0a1aaa-	--10010F	-W01F0Fh
a_7	ww0w0111	11111111	111-010F	0w0W0110
a_8	w10wvv01	11110000	010-111F	1-Wh000F
a_9	00WV--11	11111111	111----0	---F1F01
a_{10}	W11x-Vvv	--------	-----a--	-1ww1h0w
a_{11}	100V----	--------	-------1	-1hh0hWw
a_{12}	wwWF-v--	--------	--------	-1hhhh0h
a_{13}	0wW--V--	-F-F-F--	FNqNqqqq	q1hhh0WW
a_{14}	1WWhhhhh	hhhhhhhh	hNhNqNNq	NNhhh1wh
a_{15}	WWwhhhhh	hhhhhhhh	hqhhqqqq	qNwh0hh0
a_{16}	w1Whhhhh	hhhhhhhh	hhNhqqqq	hqwh1hAh
a_{17}	00------	--------	--------	----0-0-
a_{18}	1-1-----	--------	--------	----a-0-
a_{19}	0-b-----	--------	--------	------0-
a_{20}	--0-----	--------	--------	----a---
a_{21}	--b-----	--------	--------	------0-
a_{22}	--------	--------	--------	----aa--
a_{23}	--------	--------	--------	------00
a_{24}	-c------	--------	--------	----a---
a_{25}	-B------	--------	--------	----a-0-
a_{26}	--------	--------	--------	-----A1-
a_{27}	--------	--------	--------	-------0
a_{28}	-C------	--------	--------	----a---
a_{29}	-b------	--------	--------	----a-1-
a_{30}	--------	--------	--------	-----A0-
a_{31}	--------	--------	--------	-------1
a_{32}	-c------	--------	--------	--------
a_{33}	-b------	--------	--------	----a---
a_{34}	--------	--------	--------	----AA0-
a_{35}	--------	--------	--------	------00
a_{36}	-c------	--------	--------	----A---
a_{37}	-B------	--------	--------	----A-1-
a_{38}	--------	--------	--------	------0-
a_{39}	--------	--------	--------	--------
a_{40}	B-------	--------	--------	----A---
a_{41}	--------	--------	--------	------1-
a_{42}	C-------	--------	--------	--------
a_{43}	B-------	--------	--------	--------
a_{44}	--------	--------	--------	----A---
a_{45}	--------	--------	--------	------0-
a_{46}	C-------	--------	--------	----A---
a_{47}	B-------	--------	--------	------0-
a_{48}	C-------	--------	--------	----A---
a_{49}	B-------	--------	--------	------0-
a_{50}	C-------	--------	--------	----A---
a_{51}	B-------	--------	--------	------0-
a_{52}	C-------	--------	--------	--------
a_{53}	B-------	--------	--------	--------
a_{54}	--------	--------	--------	--------
a_{55}	--------	--------	--------	--------
a_{56}	--------	--------	--------	--------
a_{57}	--------	--------	--------	--------
a_{58}	--------	--------	--------	--------
a_{59}	--------	--------	--------	--------
a_{60}	--------	--------	--------	--------
a_{61}	--------	--------	--------	--------
a_{62}	--------	--------	--------	--------
a_{63}	--------	--------	--------	--------
a_{64}	--------	--------	--------	--------
a_{65}	--------	--------	--------	--------
a_{66}	--------	--------	--------	---A----
a_{67}	--------	--------	--------	-----0--
a_{68}	--------	--------	--------	-------C
a_{69}	--------	--------	--------	--A----B
a_{70}	--------	--------	--------	----0---
a_{71}	--------	--------	--------	------C-
a_{72}	--------	--------	--------	-A----B-
a_{73}	--------	--------	--------	---0----
a_{74}	--------	--------	--------	--A--C--
a_{75}	--------	--------	--------	A---0B--
a_{76}	--------	--------	--------	--1---C-
a_{77}	--------	--------	--------	----C-B-
a_{78}	--------	--------	--------	----b---
a_{79}	--------	--------	--------	-1------
a_{80}	--------	--------	--------	--------

Table 8. Semi-neutral bits and Adjusters