How To Find Many Collisions of 3-Pass HAVAL

Kazuhiro Suzuki[1], Kaoru Kurosawa[2]

[1] Venture Business Laboratory, Ibaraki University, Hitachi, Ibaraki 316-8511, Japan
[2] Department of Computer and Information Sciences, Ibaraki University, Hitachi, Ibaraki 316-8511, Japan

Abstract. The hash function HAVAL is an Australian extension of well known Merkle-Damgård hash functions such as MD4 and MD5. It has three variants, 3-, 4- and 5-pass HAVAL. On 3-pass HAVAL, the best known attack finds a collision pair with 2^7 computations of the compression function. To find k collision pairs, it requires $2^7 k$ computations. In this paper, we present a better collision attack on 3-pass HAVAL, which can find k collision pairs with only $2k + 33$ computations. Further, our message differential is different from the previous ones. (It is important to find collisions for different message differentials.)

Key words: hash function, HAVAL, collision, differential attack

1 Introduction

The hash function HAVAL was proposed by Zheng, Pieprzyk, and Seberry at Auscrypt '92 [8]. It is an Australian extension of well known Merkle-Damgård hash functions such as MD4 and MD5. [3] HAVAL has three variants, 3-, 4- and 5-pass HAVAL, which means that the compression function has 96, 128, and 160 rounds, respectively. The compression function H of HAVAL takes a 256-bit initial value and a 1024-bit message $M = (m_0, \ldots, m_{31})$ as input, and produces 256-bit hash value as output, where each m_i is a 32-bit word.

On 3-pass HAVAL, Rompay, et al. [2] presented a collision attack that requires 2^{29} computations of the compression function. Their attack can find a one-block (1024-bit) collision pair $M = (m_0, \ldots, m_{31})$ and $M' = (m'_0, \ldots, m'_{31})$ with the differential

$$\Delta m_{28} = 2^0 = 1 \text{ and } \Delta m_i = 0 \text{ for the other } i.$$

X.Y.Wang et al. [4] showed a much better collision attack with 2^7 computations of the compression function. Their attack can find a one-block collision pair $M = (m_0, \ldots, m_{31})$ and $M' = (m'_0, \ldots, m'_{31})$ with the differential

$$\Delta m_0 = 2^{10},\ \Delta m_{11} = 2^{31},\ \Delta m_{18} = 2^3, \text{ and } \Delta m_i = 0 \text{ for the other } i.$$

[3] The newest version is HAVAL 1.1. We can download the program source code at the website [1]. The difference between the first version and HAVAL 1.1 is only the order of initial values and the other constant values.

To find k collision pairs of 3-pass HAVAL, the best known attack [4] requires $2^7 k$ computations.

In this paper, we present a better collision attack on 3-pass HAVAL which can find k collision pairs with only $2k+33$ computations. Further, our message differential is different from the previous ones. [4] (It is important to find collisions for different message differentials.)

The previous attacks [2, 4] are one-block collision attacks (i.e. a collision pair is a pair of 1024-bit message block). On the other hand, our attack is a two-block collision attack which can find a two-block (2048-bit) collision pair $M_0||M_1 = (m_{0,0}, \ldots, m_{0,31}, m_{1,0}, \ldots, m_{1,31})$ and $M'_0||M'_1 = (m'_{0,0}, \ldots, m'_{0,31}, m'_{1,0}, \ldots, m'_{1,31})$ with the differential

$$\Delta m_{j,i} = m'_{j,i} - m_{j,i} \bmod 2^{32} = \begin{cases} 2^{31} & \text{if } i = 5, \\ 0 & \text{otherwise.} \end{cases}$$

In our attack, we first find a near-collision pair (M_0, M'_0) such that $H(M_0)$ and $H(M'_0)$ are almost the same. We then find many full collision pairs $(M_0||M_1, M'_0||M'_1)$ by using the freedom of (M_1, M'_1). Theoretically, our near-collision pair can be found by about 33 computations of the compression function. Once a near collision pair is found, a full collision pair can be found with probability $1/2$. Hence we can find k collision pairs with $2k+33$ computations. (See Table 1.)

	Rompay, et al. [2]	X.Y.Wang et al. [7]	Proposed
Δm_i	$\Delta m_{28} = 2^0 = 1$	$\Delta m_0 = 2^{10}$ $\Delta m_{11} = 2^{31}$ $\Delta m_{18} = 2^3$	$\Delta m_{0,5} = 2^{31}$ $\Delta m_{1,5} = 2^{31}$
complexity for first collision	2^{29}	2^7	$2+33$
complexity for k collision pairs	$2^{29}k$	$2^7 k$	$2k+33$
message length	1024 bits	1024 bits	2048 bits

Table 1. Collision attacks on 3-pass HAVAL

In our personal computer simulation:

1. We found 15147 near-collision pairs by 500000 trials, which agrees with our theoretical estimate because $500000/15147 = 33.0098\cdots$.
2. From a single near-collision pair, we found 249630 full collision pairs by 500000 trials, which also agrees with our theoretical complexity because $500000/249630 \approx 2$.

[4] Our differential is used in an attack on 4-pass HAVAL by H.Yu et al. [7], but it is new for 3-pass HAVAL.

It took about one minute for the first 500000 trials. It also took about one minute for the next 500000 trials.

(Related wroks:)

- Modular differential attack was presented in 1997 by X.Y.Wang [3] and formalized in Eurocrypt '05 [5, 6]. They showed that it is very powerful to break MD4, MD5, SHA-0, SHA-1 and HAVAL. Our attack is also based on the *modular differential* approach.
- On 4-pass HAVAL, H.Yu et al. [7] showed two two-block collision attacks that require 2^{43} and 2^{36} computations of 4-pass HAVAL, respectively. On 5-pass HAVAL, the H.Yu et al. [7] showed a one-block collision attack with 2^{123} computations of the compression function.

This paper is organized as follows. In Section 2, we provide a simple description of 3-pass HAVAL. In Section 3, we give an outline of our attack. In Section 4, we present the algorithm of our attack, and calculate the complexity. In Section 5, we report on our computational experiment and a collision example. In Section 6, we conclude this paper. In this paper, almost Tables and Figures are in the Appendix.

2 3-Pass HAVAL

HAVAL consists of three phases: (1) message padding phase, (2) main hashing phase and (3) optional compression phase.

2.1 Message Padding Phase

HAVAL pads an input message by appending some bit string so that its bit-length becomes a multiple of 1024.

2.2 Main Hashing Phase

HAVAL is a Merkle-Damgård hash function based on a compression function H as follows. Let $M_0||M_1||\cdots||M_t$ be the padded message, where $|M_i| = 1024$. Then for $i = 0, \cdots, t$, compute

$$IV_{i+1} = H(IV_i, M_i)$$

where $|IV_i| = 256$ and $IV_0 = (a, b, c, d, e, f, g, h)$ is the initial value such that

$$a = 0x243f6a88,\ b = 0x85a308d3,\ \ c = 0x13198a2e,\ \ d = 0x03707344,$$
$$e = 0xa4093822,\ \ f = 0x299f31d0,\ g = 0x082efa98,\ h = 0xec4e6c89.$$

The hashed value is given by IV_{t+1}.

H is described as follows. First define three functions as follows.

$$
\begin{aligned}
F_1(x_0,x_1,x_2,x_3,x_4,x_5,x_6) &= (x_2 \bullet x_3) \oplus (x_2 \bullet x_4) \oplus x_4 \oplus (x_0 \bullet x_6) \oplus (x_1 \bullet x_5),\\
F_2(x_0,x_1,x_2,x_3,x_4,x_5,x_6) &= (x_0 \bullet x_2) \oplus (x_1 \bullet x_2) \oplus (x_1 \bullet x_3) \oplus (x_0 \bullet x_3 \bullet x_5)\\
&\quad \oplus (x_1 \bullet x_2 \bullet x_5) \oplus (x_3 \bullet x_5) \oplus (x_4 \bullet x_5)\\
&\quad \oplus x_6 \oplus (x_5 \bullet x_6),\\
F_3(x_0,x_1,x_2,x_3,x_4,x_5,x_6) &= x_0 \quad \oplus (x_0 \bullet x_3) \oplus (x_1 \bullet x_4) \oplus (x_2 \bullet x_5)\\
&\quad \oplus (x_3 \bullet x_4 \bullet x_5) \oplus (x_3 \bullet x_6),
\end{aligned}
$$

where x_i is a 32-bit word, $x_i \bullet x_j$ is the bit-wise multiplication of x_i and x_j, and $x_i \oplus x_j$ is the bit-wise modulo 2 addition.

H next runs the following algorithm $\tilde{H}$ on input

$$
\begin{aligned}
IV &= (a_0, b_0, c_0, d_0, e_0, f_0, g_0, h_0),\\
M &= (m_0, m_1, \ldots, m_{31}),
\end{aligned}
$$

where each of $a_0, \ldots, h_0$ and m_i is a 32-bit word.

For($i = 0$ to 95){

$$
\begin{aligned}
j &: = \lfloor i/32 \rfloor + 1;\\
p_i &: = F_j(a_i, b_i, c_i, d_i, e_i, f_i, g_i);\\
a_{i+1} &: = (p_i \gg 7) + (h_i \gg 11) + m_{ord(i)} + k_i \bmod 2^{32}; \qquad (1)\\
b_{i+1} &: = a_i,\ c_{i+1} := b_i,\ d_{i+1} := c_i,\ e_{i+1} := d_i,\\
f_{i+1} &: = e_i,\ g_{i+1} := f_i,\ h_{i+1} := g_i;
\end{aligned}
$$

}

where $x \gg s$ denotes the s-bit right rotation of x, $+$ denotes the modulo 2^{32} addition, and the word processing orders $ord(i)$ and the constant values k_i are given in Table 3. Note that $\tilde{H}$ consists of 96 rounds, 0-round through 95-round.

Finally, H outputs the following 256-bit value

$$
\begin{aligned}
&IV + \tilde{H}(IV, M)\\
&= (a_0 + a_{96}, b_0 + b_{96}, c_0 + c_{96}, d_0 + d_{96}, e_0 + e_{96}, f_0 + f_{96}, g_0 + g_{96}, h_0 + h_{96}).
\end{aligned}
$$

Figure 1 is an outline sketch of H. In the Appendix, we provide more detailed sketches for each round. (See Figure 2, 3, and 4.)

2.3 Optional Compression Phase

HAVAL supports hash-sizes of 128, 160, 192, 224 and 256 bits. The main algorithm computes 256-bit hash-values, and the other sizes are obtained by post-processing the 256-bit hash-value.

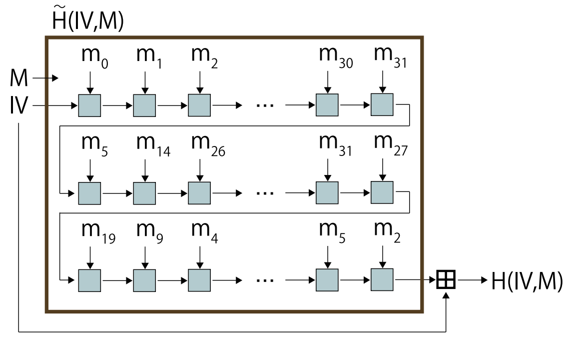

Fig. 1. The compression function H of 3-pass HAVAL.

3 Outline of our Attack

3.1 Notation

For $H(IV, M)$, $a_i, b_i, \cdots$ denote the local values which appear in i-round. Similarly, for $H(IV, M')$, $a'_i, b'_i, \cdots$ denote the local values which appear in i-round. We denote by $aa_0, bb_0, cc_0, dd_0, ee_0, ff_0, gg_0, hh_0$ the 8 words of IV_1. Define $\Delta a_i = a'_i - a_i \bmod 2^{32}$, and so on.

We denote by $x_{i,j}$ the j-th bit of 32-bit word x_i.

- $x'_i = x_i[j]$ means that x'_i is obtained by changing the jth bit of x_i from 0 to 1. That is, $x'_i = x_i$ except for that $x_{i,j} = 0$ and $x'_{i,j} = 1$.
- $x'_i = x_i[-j]$ means that x'_i is obtained by changing the jth bit of x_i from 1 to 0. That is, $x'_i = x_i$ except for that $x_{i,j} = 1$ and $x'_{i,j} = 0$.
- $x'_i = x_i[\pm j]$ means that $x_{i,j} \neq x'_{i,j}$.
- For example, $a_{14}[-21, 22]$ is the value obtained by modifying the 21-th and 22-th bit of a_{14} from 1 to 0 and 0 to 1, respectively.

3.2 Attack

We show an efficient method to find a two-block (2048-bit) collision pair

$$M_0||M_1 = (m_{0,0}, \ldots, m_{0,31}, m_{1,0}, \ldots, m_{1,31})$$
$$M'_0||M'_1 = (m'_{0,0}, \ldots, m'_{0,31}, m'_{1,0}, \ldots, m'_{1,31}).$$

The proposed method first finds a near collision pair (M_0, M'_0) such that

$$\Delta IV_1 = IV'_1 - IV_1 = (0, 2^{31}, 0, 0, 0, 0, 0, 0)$$

We next find a pair (M_1, M'_1) such that

$$\begin{aligned} \Delta A &= \tilde{H}(IV'_1, M'_1) - \tilde{H}(IV_1, M_1) \\ &= (0, 2^{31}, 0, 0, 0, 0, 0, 0) \text{ or } (0, -2^{31}, 0, 0, 0, 0, 0, 0) \end{aligned} \tag{2}$$

Then it holds that

$$H(IV_1', M_1') = IV_1' + \tilde{H}(IV_1', M_1') = IV_1 + \tilde{H}(IV_1, M_1) = H(IV_1, M_1) \qquad (3)$$

That is,

$$\Delta IV_0 = 0 \overset{(M_0, M_0')}{\longrightarrow} \Delta IV_1 \overset{(M_1, M_1')}{\longrightarrow} \Delta A \rightarrow \Delta H = 0$$

Therefore, $(M_0||M_1, M_0'||M_1')$ is a collision pair. [5]

We use a message differential such that

$$\Delta m_{j,i} = m_{j,i}' - m_{j,i} \bmod 2^{32} = \begin{cases} 2^{31} & \text{if } i = 5, \\ 0 & \text{otherwise} \end{cases}$$

for $j = 0$ and 1. [6] That is,

$$\Delta M_0 = (0, 0, 0, 0, 0, 2^{31}, 0, \cdots, 0)$$

$$\Delta M_1 = (0, 0, 0, 0, 0, 2^{31}, 0, \cdots, 0)$$

Note that m_5 is the input to the 5-, 32-, and 94-round in each block because

$$5 = ord(5) = ord(32) = ord(94)$$

from Table 3. Now we will find the first block pair (M_0, M_0') that causes a local collision at the 32-round. Then (M_0, M_0') is automatically a near collision pair just after 94-round with difference $(0, \pm 2^{31}, 0, 0, 0, 0, 0, 0)$. This can be seen from the following table.

round	0	$\cdots$	5	$\cdots$	32	$\cdots$	94	95
Δm_i	$\Delta m_0 = 0$	$\cdots$	$\Delta m_5 = 2^{31}$	$\cdots$	$\Delta m_5 = 2^{31}$	$\cdots$	$\Delta m_5 = 2^{31}$	$\cdots$
$\tilde{H}(IV, M)$					collision		near collision	near collision

Similarly we will find the second block pair (M_1, M_1') which satisfy eq.(2), where $IV_1 = H(IV_0, M_0)$ and $IV_1' = H(IV_0, M_0')$. Then $M_0||M_0'$ and $M_1||M_1'$ are a full collision pair from eq.(3).

We present the (so called) *differential path* in Table 4 and Table 5. [7] In these Table, for example,

round i	m_i'	Δa_{i+1}	Outputs $a_i', b_i', c_i', d_i', e_i', f_i', g_i', h_i'$
6	m_6	0	$a_7, a_6[32], a_5, a_4, a_3, a_2, a_1, a_0$

means that we want the outputs $(a_7', b_7', c_7', d_7', e_7', f_7', g_7', h_7')$ in the 6-round of $H(IV', M')$ to be $(a_7, b_7[32], c_7, d_7, e_7, f_7, g_7, h_7)$. Note that $(a_7, b_7, c_7, d_7, e_7, f_7, g_7, h_7) = (a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$. We can find a full collision if all the conditions of these tables are satisfied.

[5] The operation + is the word-wise modular 2^{32} addition.
[6] This differential was used for 4-pass HAVAL [7]. Its complexity is 2^{43}.
[7] Table 4 was given in [7]. We constructed Table 5.

3.3 Sufficient Conditions

In Table 6, 7, and 8, we present sufficient conditions for the differential path to hold. If Table 6, 7, and 8 are satisfied, then Table 4 and 5 are satisfied. As an example, we prove that the conditions for 5- and 6-round given in Table 6 guarantee that the conditions for 0- to 6-round shown in Table 4. The other conditions are derived similarly.

Since $\Delta m_0 = \cdots = \Delta m_4 = 0$, the differential path of 0- to 4-round hold. In 5-round of $H(IV_0', M_0')$, if the sufficient condition of 5-round in Table 6, that is, $a_{6,32} = 0$, then $a'_{6,32} = 1$ and $a'_{6,i} = a_{6,i}$ for $i \neq 32$ (i.e. $a'_6 = a_6[32]$), because

$$\begin{aligned} a_6 &= (p_5 \gg 7) + (h_5 \gg 11) + m_{ord(5)} + k_5 \\ a'_6 &= (p'_5 \gg 7) + (h'_5 \gg 11) + m'_{ord(5)} + k_5 \\ &= (p_5 \gg 7) + (h_5 \gg 11) + (m_{ord(5)} + 2^{31}) + k_5 \\ &= (p_5 \gg 7) + (h_5 \gg 11) + m_{ord(5)} + k_5 + 2^{31} \\ &= a_6 + 2^{31}. \end{aligned}$$

In 6-round of $H(IV_0', M_0')$, if the sufficient condition of 5-round in Table 6, that is, $a_{0,32} = 0$ then, since

$$\begin{aligned} p_6 &= F_1(a_6, b_6, c_6, d_6, e_6, f_6, g_6) \\ &= F_1(a_6, a_5, a_4, a_3, a_2, a_1, a_0) \\ &= (a_4 \bullet a_3) \oplus (a_4 \bullet a_2) \oplus a_2 \oplus (a_6 \bullet a_0) \oplus (a_5 \bullet a_1), \end{aligned}$$

we have

$$\begin{aligned} p_{6,32} &= (a_{4,32} \bullet a_{3,32}) \oplus (a_{4,32} \bullet a_{2,32}) \oplus a_{2,32} \oplus (a_{6,32} \bullet a_{0,32}) \oplus (a_{5,32} \bullet a_{1,32}), \\ &= (a_{4,32} \bullet a_{3,32}) \oplus (a_{4,32} \bullet a_{2,32}) \oplus a_{2,32} \oplus (0 \bullet 0) \oplus (a_{5,32} \bullet a_{1,32}), \\ &= (a_{4,32} \bullet a_{3,32}) \oplus (a_{4,32} \bullet a_{2,32}) \oplus a_{2,32} \oplus 0 \oplus (a_{5,32} \bullet a_{1,32}), \\ p'_{6,32} &= (a'_{4,32} \bullet a'_{3,32}) \oplus (a'_{4,32} \bullet a'_{2,32}) \oplus a'_{2,32} \oplus (a'_{6,32} \bullet a'_{0,32}) \oplus (a'_{5,32} \bullet a'_{1,32}), \\ &= (a'_{4,32} \bullet a'_{3,32}) \oplus (a'_{4,32} \bullet a'_{2,32}) \oplus a'_{2,32} \oplus (1 \bullet 0) \oplus (a'_{5,32} \bullet a'_{1,32}), \\ &= (a'_{4,32} \bullet a'_{3,32}) \oplus (a'_{4,32} \bullet a'_{2,32}) \oplus a'_{2,32} \oplus 0 \oplus (a'_{5,32} \bullet a'_{1,32}), \end{aligned}$$

Hence, $p'_6 = p_6$. Therefore $a'_7 = a_7$. The above equations for $p_{6,32}$ and $p'_{6,32}$ are clear by Figure 2.

4 Details

In this section, we present the details of our algorithm, and calculate the complexity to find a collision pair.

4.1 Our Algorithm

We observe that from $a_1, \cdots, a_{32}$, $M_0 = (m_0, \ldots, m_{31})$ is uniquley determined from eq.(1), and $a_{33}, \cdots, a_{96}$ are also uniquely determined. Now in Table 6, all the rows except the last three rows specify the conditions on $a_1, \cdots, a_{32}$. Hence:

1. We choose $a_1, \cdots, a_{32}$ which satisfy these conditions randomly.
2. We compute $M_0 = (m_0, \ldots, m_{31})$ from eq.(1). [8]
3. If the last three rows are also satisfied, then we have done.

Next for given M_0, we apply the same strategy to find M_1.

1. We choose $a_1, \cdots, a_{32}$ which satisfy the conditions of Table 7 and Table 8 randomly.
2. We compute M_1 from eq.(1).
3. If the last row of Table 8 is also satisfied, then we have done.

Finding M_0:
1. Randomly select $a_1, \ldots, a_{32}$
 that satisfy the sufficient conditions for 0-31 rounds.
2. For $i = 0$ to 31,
 $b_{i+1} := a_i,\ c_{i+1} := b_i,\ d_{i+1} := c_i,\ e_{i+1} := d_i,$
 $f_{i+1} := e_i,\ g_{i+1} := f_i,\ h_{i+1} := g_i.$
3. Calculate $p_0, \ldots, p_{31}$ of the algorithm of H in Section 2
 and $m_0, \ldots, m_{31}$ as follows,
 $p_i := F_1(a_i, b_i, c_i, d_i, e_i, f_i, g_i),$
 $m_i := a_{i+1} - (p_i \gg 7) - (h_i \gg 11) - k_i \bmod 2^{32}.$
4. Execute 32- to 95-round of the compression function.
5. If $a_{95,32} = 1$, $a_{92,32} = 1$, $bb_{0,32} = 0$, $ff_{0,32} = 0$,
 $cc_{0,32} = dd_{0,32}$, and $aa_{0,32} = 0$,
 then fix $M_0 = (m_0, \ldots, m_{31})$.

Finding M_1:
6. Randomly select $a_1, \ldots, a_{32}$
 that satisfy the sufficient conditions for 0-31 rounds.
7. For $i = 0$ to 31,
 $b_{i+1} := a_i,\ c_{i+1} := b_i,\ d_{i+1} := c_i,\ e_{i+1} := d_i,$
 $f_{i+1} := e_i,\ g_{i+1} := f_i,\ h_{i+1} := g_i.$
8. Calculate $p_0, \ldots, p_{31}$ of the algorithm of H in Section 2
 and $m_0, \ldots, m_{31}$ as follows,
 $p_i := F_1(a_i, b_i, c_i, d_i, e_i, f_i, g_i),$
 $m_i := a_{i+1} - (p_i \gg 7) - (h_i \gg 11) - k_i \bmod 2^{32}.$
9. Execute 32- to 95-round of the compression function.
10. If $a_{92,32} = 1$, then output $(M_0||M_1)$ and $(M_0'||M_1')$ as a collision pair.

4.2 Success Probability

Assume that

$$Pr[x_{i,j} = 0] = Pr[x_{i,j} = 1] = 1/2 \tag{4}$$

[8] Note that our algorithm doesn't requre message modification.

for any word x_i. We can find M_0 if the last three condition are satisfied in Table 6. Thereore, the success probability P of finding M_0 is given by

$$P = Pr[a_{95,32} = 1, a_{92,32} = 1, aa_{0,32} = 0, bb_{0,32} = 0, cc_{0,32} = dd_{0,32}, ff_{0,32} = 0,$$
$$= 1/2^5 \times Pr[bb_{0,32} = 0]$$

For bb_0, note that

$$bb_0 = b_0 + b_{96} = b + a_{95} = (10000101 \cdots) + (1??????? \cdots).$$

Thus, if $a_{95,31}$, $a_{95,30}$, $a_{95,29}$, or $a_{95,28}$ is 0, or $a_{95,31} = a_{95,30} = a_{95,29} = a_{95,28} = 1$ and $a_{95,27} = a_{95,26} = 0$, then $bb_{0,32} = 0$. Hence

$$\begin{aligned}
&Pr[bb_{0,32} = 0] \\
&\geq Pr[a_{95,31} = 0, a_{95,30} = 0, a_{95,29} = 0, \text{ or } a_{95,28} = 0] \\
&\quad + Pr[a_{95,31} = a_{95,30} = a_{95,29} = a_{95,28} = 1 \text{ and } a_{95,27} = a_{95,26} = 0] \\
&= (1 - Pr[a_{95,31} = 1, a_{95,30} = 1, a_{95,29} = 1, \text{ and } a_{95,28} = 1]) + 1/2^6 \\
&= (1 - 1/2^4) + 1/64 = 15/16 + 1/64 = 61/64.
\end{aligned}$$

Therefore

$$P \geq 1/2^5 \times 61/64 = 61/2^{11} \approx 1/33.$$

Next suppose that the above M_0 is given. Then we can find M_1 if the last row of Table 8 is satisfied. Thereore, the success probability of finding M_1 is given by

$$Pr[a_{92,32} = 1] = 1/2.$$

4.3 How to Find Many Collisions

We can find many collision pairs from fixed (M_0, M_0') by running the algorithm "Finding M_1" many times. In this method, the complexity of finding k collision pairs is $2k + 33$.

5 Computational experiment

We implemented our attack by a personal computer. First we found 15,147 desired M_0s by running the algorithm "Finding M_0" $500,000$ times. In this experiment, the success probability $15,147/500,000 \approx 1/33$. It coincides with our theoretical probability shown in Section 3.

Next for fixed M_0, we found $249,630$ desired M_1s by running the algorithm "Finding M_1" $500,000$ times. [9] In this experiment, the success probability $249,630/500,000 \approx 1/2$. It coincides with our theoretical probability shown in Section 3.

In total, we found $249,630$ full collision pairs by running the algorithms "Finding M_0" 39 times and "Finding M_1" $500,000$ times.

Consequently, our experiment supports our claim that we can find k collision pairs with $2k + 33$ computations of the compression functions. We illustrate one of the 249630 collision pairs in Table 2.

[9] It takes about one minute on our computer.

M_0	c7f10962	08cf4e0c	ddfd60a8	597cbd0d	b050440c	205560d0	84569b2f	43b834dc
	1270d097	2b027ff7	32247646	8056892d	906feca6	a0a6b4ec	fbc11aca	d12586db
	f7e7bae1	ca89b85f	2d5a3e0f	8b4557da	8596d1bb	2bf5e1fd	b5b7f669	9445ea09
	343860ec	5c746759	bbce300c	d0985871	5229b382	8dab9e3e	f89f39d6	9179329b
M_1	cc4e7f72	c195d858	e5e2baf1	af7db590	84ddea8e	5990fd91	f6865ea5	9db928ce
	d3555dbd	6bf9b53a	694e5fff	e96766dc	2d541b98	d394d721	6a84b2c2	0d2bd1a1
	3afdac64	f0f67f58	60dd3e5d	aec84176	575012f1	24878a2f	304720ed	25eed9ae
	447f0e6e	b03eaa86	9fa12c2a	e98b9370	2e5cb01c	a2e23d56	cdaf12f2	2efb842d
M'_0	c7f10962	08cf4e0c	ddfd60a8	597cbd0d	b050440c	a05560d0	84569b2f	43b834dc
	1270d097	2b027ff7	32247646	8056892d	906feca6	a0a6b4ec	fbc11aca	d12586db
	f7e7bae1	ca89b85f	2d5a3e0f	8b4557da	8596d1bb	2bf5e1fd	b5b7f669	9445ea09
	343860ec	5c746759	bbce300c	d0985871	5229b382	8dab9e3e	f89f39d6	9179329b
M'_1	cc4e7f72	c195d858	e5e2baf1	af7db590	84ddea8e	d990fd91	f6865ea5	9db928ce
	d3555dbd	6bf9b53a	694e5fff	e96766dc	2d541b98	d394d721	6a84b2c2	0d2bd1a1
	3afdac64	f0f67f58	60dd3e5d	aec84176	575012f1	24878a2f	304720ed	25eed9ae
	447f0e6e	b03eaa86	9fa12c2a	e98b9370	2e5cb01c	a2e23d56	cdaf12f2	2efb842d
H	c9f26b47	513d34a2	0ad20a17	3d207470	04848b80	fc90cc0a	ef1cf172	d48c0d25

Table 2. Collision example.

6 Conclusion

On 3-pass HAVAL, the best known attack finds a collision pair with 2^7 computations of the compression function. To find k collision pairs, it requires 2^7k computations.

In this paper, we presented a better collision attack on 3-pass HAVAL using modular differential method. It can find k collision pairs with only $2k + 33$ computations. Further, our message differential is different from the previous ones. (It is important to find collision pairs for different message differentials.)

References

1. Calyptix Security Corporation http://labs.calyptix.com/haval.php.: HAVAL Version 1.1, 2003.
2. Rompay, Bart Van; Biryukov, Alex; Preneel, Bart; Vandewalle, Joos: Cryptanalysis of 3-Pass HAVAL, Asiacrypt 2003, LNCS 2894, pp.228-245, 2003.
3. Wang, Xiaoyun: The Collision attack on SHA-0, in Chinese, to appear on www.infosec.sdu.edu.cn, 1997.
4. Wang, Xiaoyun; Feng, Dengguo; Yu, Xiuyuan: An attack on hash function HAVAL-128, Science in China Ser.F Information Sciences 2005, Vol.48, No.5, pp.545-556.
5. Wang, Xiaoyun; Lai, Xuejia; Feng, Dengguo; Chen, Hui; Yu, Xiuyuan: Cryptanalysis of the Hash Functions MD4 and RIPEMD, Eurocrypt '05, LNCS 3494, pp.1-18, 2005.
6. Wang, Xiaoyun; Yu, Hongbo: How to Break MD5 and Other Hash Functions, Eurocrypt '05, LNCS 3494, pp.19-35, 2005.

7. Yu, Hongbo; Wang, Xiaoyun; Yun, Aaram; Park, Sangwoo: Cryptanalysis of the Full HAVAL with 4 and 5 Passes, FSE 2006, LNCS 4047, pp.89-110, 2006.
8. Zheng, Yuliang; Pieprzyk, Josef; Seberry, Jennifer: HAVAL – A One-Way Hashing Algorithm with Variable Length of Output (Extended Abstract), Auscrypt '92, LNCS 718, pp.83-104, 1993.

Appendix : Tables and Figures

i	$ord(i)$
0 to 31	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
32 to 63	5 14 26 18 11 28 7 16 0 23 20 22 1 10 4 8 30 3 21 9 17 24 29 6 19 12 15 13 2 25 31 27
64 to 95	19 9 4 20 28 17 8 22 29 14 25 12 24 30 16 26 31 15 7 3 1 0 18 27 13 6 21 10 23 11 5 2

i	k_i
0 to 31	00000000
32 to 63	452821e6 38d01377 be5466cf 34e90c6c c0ac29b7 c97c50dd 3f84d5b5 b5470917 9216d5d9 8979fb1b d1310ba6 98dfb5ac 2ffd72db d01adfb7 b8e1afed 6a267e96 ba7c9045 f12c7f99 24a19947 b3916cf7 0801f2e2 858efc16 636920d8 71574e69 a458fea3 f4933d7e 0d95748f 728eb658 718bcd58 82154aee 7b54a41d c25a59b5
64 to 95	9c30d539 2af26013 c5d1b023 286085f0 ca417918 b8db38ef 8e79dcb0 603a180e 6c9e0e8b b01e8a3e d71577c1 bd314b27 78af2fda 55605c60 e65525f3 aa55ab94 57489862 63e81440 55ca396a 2aab10b6 b4cc5c34 1141e8ce a15486af 7c72e993 b3ee1411 636fbc2a 2ba9c55d 741831f6 ce5c3e16 9b87931e afd6ba33 6c24cf5c

Table 3. Word processing orders $ord(i)$ and constant values k_i (hexadecimal numbers).

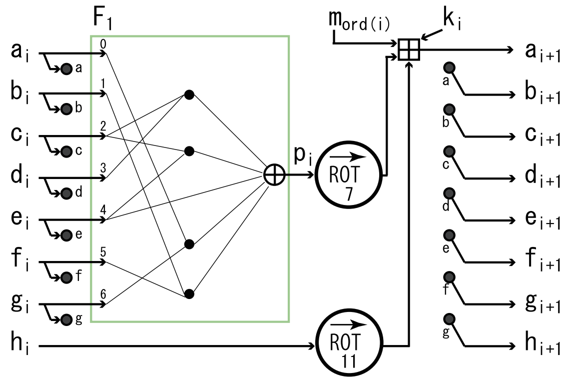

Fig. 2. i-round $(0 \leq i \leq 31)$ in the compression function.

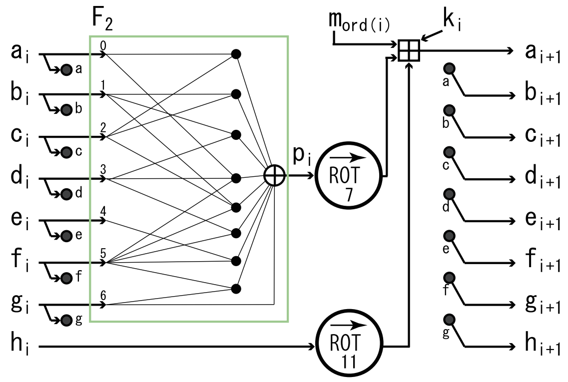

Fig. 3. i-round $(32 \leq i \leq 63)$ in the compression function.

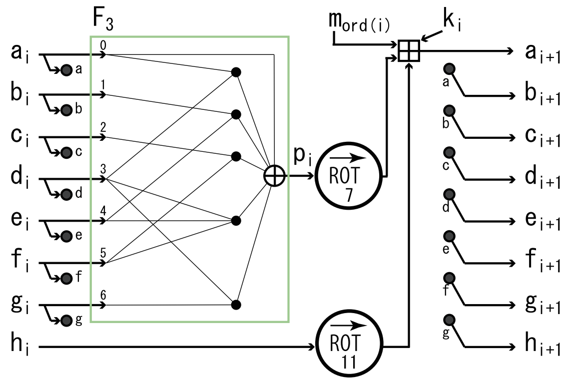

Fig. 4. i-round $(64 \leq i \leq 95)$ in the compression function.

round i	m'_i	Δa_{i+1}	Outputs $a'_i, b'_i, c'_i, d'_i, e'_i, f'_i, g'_i, h'_i$
IV'_0			$a_0, b_0, c_0, d_0, e_0, f_0, g_0, h_0$
0	m_0	0	$a_1, a_0, b_0, c_0, d_0, e_0, f_0, g_0$
1	m_1	0	$a_2, a_1, a_0, b_0, c_0, d_0, e_0, f_0$
2	m_2	0	$a_3, a_2, a_1, a_0, b_0, c_0, d_0, e_0$
3	m_3	0	$a_4, a_3, a_2, a_1, a_0, b_0, c_0, d_0$
4	m_4	0	$a_5, a_4, a_3, a_2, a_1, a_0, b_0, c_0$
5	m'_5	2^{31}	$a_6[32], a_5, a_4, a_3, a_2, a_1, a_0, b_0$
6	m_6	0	$a_7, a_6[32], a_5, a_4, a_3, a_2, a_1, a_0$
7	m_7	0	$a_8, a_7, a_6[32], a_5, a_4, a_3, a_2, a_1$
8	m_8	0	$a_9, a_8, a_7, a_6[32], a_5, a_4, a_3, a_2$
9	m_9	0	$a_{10}, a_9, a_8, a_7, a_6[32], a_5, a_4, a_3$
10	m_{10}	0	$a_{11}, a_{10}, a_9, a_8, a_7, a_6[32], a_5, a_4$
11	m_{11}	0	$a_{12}, a_{11}, a_{10}, a_9, a_8, a_7, a_6[32], a_5$
12	m_{12}	0	$a_{13}, a_{12}, a_{11}, a_{10}, a_9, a_8, a_7, a_6[32]$
13	m_{13}	2^{20}	$a_{14}[-21, 22], a_{13}, a_{12}, a_{11}, a_{10}, a_9, a_8, a_7$
14	m_{14}	0	$a_{15}, a_{14}[-21, 22], a_{13}, a_{12}, a_{11}, a_{10}, a_9, a_8$
15	m_{15}	0	$a_{16}, a_{15}, a_{14}[-21, 22], a_{13}, a_{12}, a_{11}, a_{10}, a_9$
16	m_{16}	0	$a_{17}, a_{16}, a_{15}, a_{14}[-21, 22], a_{13}, a_{12}, a_{11}, a_{10}$
17	m_{17}	-2^{14}	$a_{18}[15, 16, 17, -18], a_{17}, a_{16}, a_{15}, a_{14}[-21, 22], a_{13}, a_{12}, a_{11}$
18	m_{18}	0	$a_{19}, a_{18}[15, 16, 17, -18], a_{17}, a_{16}, a_{15}, a_{14}[-21, 22], a_{13}, a_{12}$
19	m_{19}	0	$a_{20}, a_{19}, a_{18}[15, 16, 17, -18], a_{17}, a_{16}, a_{15}, a_{14}[-21, 22], a_{13}$
20	m_{20}	0	$a_{21}, a_{20}, a_{19}, a_{18}[15, 16, 17, -18], a_{17}, a_{16}, a_{15}, a_{14}[-21, 22]$
21	m_{21}	0	$a_{22}, a_{21}, a_{20}, a_{19}, a_{18}[15, 16, 17, -18], a_{17}, a_{16}, a_{15}$
22	m_{22}	0	$a_{23}, a_{22}, a_{21}, a_{20}, a_{19}, a_{18}[15, 16, 17, -18], a_{17}, a_{16}$
23	m_{23}	0	$a_{24}, a_{23}, a_{22}, a_{21}, a_{20}, a_{19}, a_{18}[15, 16, 17, -18], a_{17}$
24	m_{24}	2^{10}	$a_{25}[11], a_{24}, a_{23}, a_{22}, a_{21}, a_{20}, a_{19}, a_{18}[15, 16, 17, -18]$
25	m_{25}	0	$a_{26}, a_{25}[11], a_{24}, a_{23}, a_{22}, a_{21}, a_{20}, a_{19}$
26	m_{26}	0	$a_{27}, a_{26}, a_{25}[11], a_{24}, a_{23}, a_{22}, a_{21}, a_{20}$
27	m_{27}	0	$a_{28}, a_{27}, a_{26}, a_{25}[11], a_{24}, a_{23}, a_{22}, a_{21}$
28	m_{28}	0	$a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[11], a_{24}, a_{23}, a_{22}$
29	m_{29}	0	$a_{30}, a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[11], a_{24}, a_{23}$
30	m_{30}	0	$a_{31}, a_{30}, a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[11], a_{24}$
31	m_{31}	0	$a_{32}, a_{31}, a_{30}, a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[11]$
32	m'_5	0	$a_{33}, a_{32}, a_{31}, a_{30}, a_{29}, a_{28}, a_{27}, a_{26}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
94	m'_5	-2^{31}	$a_{95}[-32], a_{94}, a_{93}, a_{92}, a_{91}, a_{90}, a_{89}, a_{88}$
95	m_2	0	$a_{96}, a_{95}[-32], a_{94}, a_{93}, a_{92}, a_{91}, a_{90}, a_{89}$
IV'_1			$aa_0, bb_0[32], cc_0, dd_0, ee_0, ff_0, gg_0, hh_0$

Table 4. A differential path for $H(IV_0, M_0)$ and $H(IV_0, M'_0)$.

round i	m'_i	Δa_{i+1}	Outputs $a'_i, b'_i, c'_i, d'_i, e'_i, f'_i, g'_i, h'_i$
IV'_1			$a_0, b_0[32], c_0, d_0, e_0, f_0, g_0, h_0$
0	m_0	0	$a_1, a_0, b_0[32], c_0, d_0, e_0, f_0, g_0$
1	m_1	0	$a_2, a_1, a_0, b_0[32], c_0, d_0, e_0, f_0$
2	m_2	0	$a_3, a_2, a_1, a_0, b_0[32], c_0, d_0, e_0$
3	m_3	0	$a_4, a_3, a_2, a_1, a_0, b_0[32], c_0, d_0$
4	m_4	0	$a_5, a_4, a_3, a_2, a_1, a_0, b_0[32], c_0$
5	m'_5	2^{31}	$a_6[32], a_5, a_4, a_3, a_2, a_1, a_0, b_0[32]$
6	m_6	2^{20}	$a_7[21], a_6[32], a_5, a_4, a_3, a_2, a_1, a_0$
7	m_7	0	$a_8, a_7[21], a_6[32], a_5, a_4, a_3, a_2, a_1$
8	m_8	-2^{24}	$a_9[25, 26, 27, -28], a_8, a_7[21], a_6[32], a_5, a_4, a_3, a_2$
9	m_9	0	$a_{10}, a_9[25, 26, 27, -28], a_8, a_7[21], a_6[32], a_5, a_4, a_3$
10	m_{10}	-2^{18}	$a_{11}[19, 20, -21], a_{10}, a_9[25, 26, 27, -28], a_8, a_7[21], a_6[32], a_5, a_4$
11	m_{11}	0	$a_{12}, a_{11}[19, 20, -21], a_{10}, a_9[25, 26, 27, -28], a_8, a_7[21], a_6[32], a_5$
12	m_{12}	-2^{11}	$a_{13}[12, 13, 14, 15, 16, -17], a_{12}, a_{11}[19, 20, -21], a_{10}, a_9[25, 26, 27, -28], a_8, a_7[21], a_6[32]$
13	m_{13}	-2^7	$a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17], a_{12}, a_{11}[19, 20, -21], a_{10}, a_9[25, 26, 27, -28], a_8, a_7[21]$
14	m_{14}	0	$a_{15}, a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17], a_{12}, a_{11}[19, 20, -21], a_{10}, a_9[25, 26, 27, -28], a_8$
15	m_{15}	0	$a_{16}, a_{15}, a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17], a_{12}, a_{11}[19, 20, -21], a_{10}, a_9[25, 26, 27, -28]$
16	m_{16}	0	$a_{17}, a_{16}, a_{15}, a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17], a_{12}, a_{11}[19, 20, -21], a_{10}$
17	m_{17}	0	$a_{18}, a_{17}, a_{16}, a_{15}, a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17], a_{12}, a_{11}[19, 20, -21]$
18	m_{18}	0	$a_{19}, a_{18}, a_{17}, a_{16}, a_{15}, a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17], a_{12}$
19	m_{19}	0	$a_{20}, a_{19}, a_{18}, a_{17}, a_{16}, a_{15}, a_{14}[-8], a_{13}[12, 13, 14, 15, 16, -17]$
20	m_{20}	0	$a_{21}, a_{20}, a_{19}, a_{18}, a_{17}, a_{16}, a_{15}, a_{14}[-8]$
21	m_{21}	-2^{28}	$a_{22}[-29], a_{21}, a_{20}, a_{19}, a_{18}, a_{17}, a_{16}, a_{15}$
22	m_{22}	-2^{21}	$a_{23}[22, 23, 24, -25], a_{22}[-29], a_{21}, a_{20}, a_{19}, a_{18}, a_{17}, a_{16}$
23	m_{23}	-2^{14}	$a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25], a_{22}[-29], a_{21}, a_{20}, a_{19}, a_{18}, a_{17}$
24	m_{24}	-2^{10}	$a_{25}[-11], a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25], a_{22}[-29], a_{21}, a_{20}, a_{19}, a_{18}$
25	m_{25}	0	$a_{26}, a_{25}[-11], a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25], a_{22}[-29], a_{21}, a_{20}, a_{19}$
26	m_{26}	0	$a_{27}, a_{26}, a_{25}[-11], a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25], a_{22}[-29], a_{21}, a_{20}$
27	m_{27}	0	$a_{28}, a_{27}, a_{26}, a_{25}[-11], a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25], a_{22}[-29], a_{21}$
28	m_{28}	0	$a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[-11], a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25], a_{22}[-29]$
29	m_{29}	0	$a_{30}, a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[-11], a_{24}[15, 16, 17, -18], a_{23}[22, 23, 24, -25]$
30	m_{30}	0	$a_{31}, a_{30}, a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[-11], a_{24}[15, 16, 17, -18]$
31	m_{31}	0	$a_{32}, a_{31}, a_{30}, a_{29}, a_{28}, a_{27}, a_{26}, a_{25}[-11]$
32	m'_5	0	$a_{33}, a_{32}, a_{31}, a_{30}, a_{29}, a_{28}, a_{27}, a_{26}$
...	...	...	...
94	m'_5	$\pm 2^{31}$	$a_{95}[\pm 32], a_{94}, a_{93}, a_{92}, a_{91}, a_{90}, a_{89}, a_{88}$
95	m_2	0	$a_{96}, a_{95}[\pm 32], a_{94}, a_{93}, a_{92}, a_{91}, a_{90}, a_{89}$
IV'_2			$a_0 + a_{96}, b_0[32] + a_{95}[\pm 32], c_0 + a_{94}, d_0 + a_{93}, e_0 + a_{92}, f_0 + a_{91}, g_0 + a_{90}, h_0 + a_{89}$ Full Collision !!

Table 5. A differential path for $H(IV_1, M_1)$ and $H(IV'_1, M'_1)$.

round i	Sufficient conditions for each round
5	$a_{6,32} = 0$
6	$a_{0,32} = 0$
7	$a_{2,32} = 0$
8	$a_{5,32} = a_{4,32}$
9	$a_{7,32} = 0$
10	$a_{8,32} = 1$
11	$a_{10,32} = 0$
12	$a_{12,32} = 0$
13	$a_{14,21} = 1, a_{14,22} = 0$
14	$a_{8,21} = 0, a_{8,22} = 0$
15	$a_{10,21} = 0, a_{10,22} = 0$
16	$a_{13,21} = a_{12,21}, a_{13,22} = a_{12,22}$
17	$a_{18,15} = 0$, $a_{18,16} = 0$, $a_{18,17} = 0$, $a_{18,18} = 1$, $a_{15,21} = 0$, $a_{15,22} = 1$, $a_{11,22} = 0, a_{12,22} = 1, a_{13,22} = 1, a_{16,22} = 1$
18	$a_{12,15} = 0, a_{12,16} = 0, a_{12,17} = 0, a_{12,18} = 0, a_{16,21} = 1, a_{16,22} = 1$
19	$a_{14,15} = 0, a_{14,16} = 0, a_{14,17} = 0, a_{14,18} = 0, a_{18,21} = 0, a_{18,22} = 0$
20	$a_{16,15} = a_{17,15}, a_{16,16} = a_{17,16}$, $a_{16,17} = a_{17,17}$, $a_{16,18} = a_{17,18}$, $a_{20,21} = 0, a_{20,22} = 0$
21	$a_{19,15} = 0, a_{19,16} = 0, a_{19,17} = 1, a_{19,18} = 0, a_{15,17} = 1, a_{16,17} = 0, a_{17,17} = 0, a_{21,17} = 1$
22	$a_{20,15} = 1, a_{20,16} = 1, a_{20,17} = 1, a_{20,18} = 1$
23	$a_{22,15} = 0, a_{22,16} = 0, a_{22,17} = 0, a_{22,18} = 0$
24	$a_{24,15} = 0, a_{24,16} = 0, a_{24,17} = 0, a_{24,18} = 1, a_{25,11} = 0$
25	$a_{19,11} = 1, a_{20,11} = 0, a_{22,11} = 0, a_{21,11} = 0$
26	$a_{21,11} = 0$
27	$a_{24,11} = a_{23,11}$
28	$a_{26,11} = 0$
29	$a_{27,11} = 1$
30	$a_{29,11} = 0$
31	$a_{31,11} = 0$
94	$a_{95,32} = 1$
95	$a_{92,32} = 1$
IV_1	$aa_{0,32} = 0, bb_{0,32} = 0, cc_{0,32} = dd_{0,32}, ff_{0,32} = 0$

Table 6. Sufficient conditions on a_i for the differential path in Table 4.

round i	Sufficient conditions for each round
0	$f_{0,32} = 0$
1	$c_{0,32} = d_{0,32}$
2	$a_{0,32} = 0$
3	$a_{1,32} = 1$
4	$a_{3,32} = 0$
5	$a_{5,32} = 0, a_{6,32} = 0$
6	$a_{7,21} = 0, a_{0,32} = 0$
7	$a_{1,21} = 0, a_{2,32} = 0$
8	$a_{9,25} = 0, a_{9,26} = 0, a_{9,27} = 0, a_{9,28} = 1, a_{3,21} = 0, a_{2,32} = 0, a_{3,32} = 0, a_{4,32} = 1, a_{5,32} = 0$
9	$a_{7,32} = 0, a_{5,21} = a_{6,21}, a_{3,25} = 0, a_{3,26} = 0, a_{3,27} = 0, a_{3,28} = 0$
10	$a_{11,19} = 0, a_{11,20} = 0, a_{11,21} = 1, a_{8,32} = 1, a_{8,21} = 0, a_{5,25} = 0, a_{5,26} = 1, a_{5,27} = 0, a_{5,28} = 0, a_{8,26} = 0, a_{6,26} = 1, a_{4,26} = 0$
11	$a_{5,19} = 0, a_{5,20} = 0, a_{9,21} = a_{5,21} + 1, a_{7,25} = a_{8,25}, a_{7,26} = a_{8,26}, a_{8,27} = a_{7,27}, a_{7,28} = a_{8,28}, a_{10,32} = 0$
12	$a_{13,12} = 0, a_{13,13} = 0, a_{13,14} = 0, a_{13,15} = 0, a_{13,16} = 0, a_{13,17} = 1, a_{12,32} = 0, a_{7,19} = 1, a_{7,20} = 0, a_{10,25} = 0, a_{10,26} = 0, a_{10,27} = 0, a_{10,28} = 0, a_{8,19} = 1, a_{10,19} = 0, a_{6,19} = 0$
13	$a_{14,8} = 1, a_{7,12} = 0, a_{7,13} = 0, a_{7,14} = 0, a_{7,15} = 1, a_{7,16} = 0, a_{7,17} = 0, a_{10,19} = a_{9,19}, a_{10,20} = a_{9,20}, a_{10,21} = a_{9,21} + 1, a_{11,25} = 1, a_{11,26} = 1, a_{11,27} = 1, a_{11,28} = 0, a_{7,28} = 0, a_{8,28} = 0, a_{8,15} = 0, a_{9,15} = 0, a_{10,15} = 1, a_{11,15} = 1, a_{13,21} = 1$
14	$a_{8,8} = 0, a_{9,12} = 0, a_{9,13} = 0, a_{9,14} = 0, a_{9,15} = 0, a_{9,16} = 0, a_{9,17} = 1, a_{12,19} = 0, a_{12,20} = 0, a_{12,21} = 0, a_{13,25} = 0, a_{13,26} = 0, a_{13,27} = 0, a_{13,28} = 0, a_{8,17} = 0, a_{10,17} = 0, a_{11,17} = 0$
15	$a_{10,8} = 0, a_{12,12} = a_{11,12}, a_{12,13} = a_{11,13}, a_{12,14} = a_{11,14}, a_{12,15} = a_{11,15}, a_{12,16} = a_{11,16}, a_{12,17} = a_{11,17}, a_{13,19} = 1, a_{13,20} = 1, a_{13,21} = 1, a_{15,25} = 0, a_{15,26} = 0, a_{15,27} = 0, a_{15,28} = 0$
16	$a_{13,8} = a_{12,8}, a_{14,12} = 0, a_{14,13} = 0, a_{14,14} = 0, a_{14,15} = 0, a_{14,16} = 0, a_{14,17} = 0, a_{15,19} = 0, a_{15,20} = 0, a_{15,21} = 1, a_{10,21} = 0, a_{12,21} = 0, a_{13,21} = 1, a_{14,21} = 1$
17	$a_{15,8} = 0, a_{15,12} = 1, a_{15,13} = 1, a_{15,14} = 1, a_{15,15} = 1, a_{15,16} = 1, a_{15,17} = 1, a_{17,19} = 0, a_{17,20} = 0, a_{17,21} = 0$
18	$a_{16,8} = 1, a_{17,12} = 0, a_{17,13} = 0, a_{17,14} = 0, a_{17,15} = 1, a_{17,16} = 0, a_{17,17} = 0, a_{14,15} = 0, a_{16,15} = 0, a_{18,15} = 0$
19	$a_{18,8} = 0, a_{19,12} = 0, a_{19,13} = 0, a_{19,14} = 0, a_{19,15} = 0, a_{19,16} = 0, a_{19,17} = 0$
20	$a_{15,8} = 0, a_{16,8} = 1, a_{18,8} = 0, a_{20,8} = 1$

Table 7. Sufficient conditions on a_i for 0-20 rounds of the differential path in Table 5.

round i	Sufficient conditions for each round
21	$a_{22,29}=1$
22	$a_{23,22}=0,\ a_{23,23}=0,\ a_{23,24}=0,\ a_{23,25}=1,\ a_{16,19}=1,\ a_{17,29}=0,$ $a_{18,29}=0,\ a_{19,29}=0$
23	$a_{24,15}=0, a_{24,16}=0, a_{24,17}=0, a_{24,18}=1, a_{18,29}=0, a_{17,22}=1, a_{17,23}=0, a_{17,24}=0, a_{17,25}=0, a_{18,22}=0, a_{19,22}=0, a_{20,22}=1, a_{21,22}=1$
24	$a_{25,11}=1, a_{21,29}=a_{20,29}, a_{19,22}=0, a_{19,23}=0, a_{19,24}=0, a_{19,25}=0, a_{18,15}=0, a_{18,16}=0, a_{18,17}=0, a_{18,18}=1, a_{19,18}=0, a_{20,18}=0, a_{22,18}=0$
25	$a_{19,11}=0, a_{20,15}=0, a_{20,16}=0, a_{20,17}=0, a_{20,18}=0, a_{22,22}=a_{21,22}, a_{22,23}=a_{21,23}, a_{22,24}=a_{21,24}, a_{22,25}=a_{21,25}, a_{23,29}=0$
26	$a_{21,11}=0,\ a_{23,15}=a_{22,15},\ a_{23,16}=a_{22,16}, a_{23,17}=a_{22,17}, a_{23,18}=a_{22,18}, a_{24,22}=0, a_{24,23}=0, a_{24,24}=0, a_{24,25}=0, a_{24,29}=1,$
27	$a_{24,11}=a_{23,11}, a_{25,15}=0, a_{25,16}=0, a_{25,17}=0, a_{25,18}=0, a_{25,22}=1, a_{25,23}=1, a_{25,24}=1, a_{25,25}=1, a_{26,29}=0$
28	$a_{26,11}=0, a_{26,15}=1, a_{26,16}=1, a_{26,17}=1, a_{26,18}=1, a_{27,22}=0, a_{27,23}=0, a_{27,24}=0, a_{27,25}=0, a_{28,29}=0$
29	$a_{27,11}=1, a_{28,15}=0, a_{28,16}=0, a_{28,17}=0, a_{28,18}=0, a_{29,22}=0, a_{29,23}=0, a_{29,24}=0, a_{24,25}=0, a_{25,25}=1, a_{27,25}=0, a_{29,25}=1$
30	$a_{29,11}=0, a_{30,15}=0, a_{30,16}=0, a_{30,17}=0, a_{30,18}=1, a_{25,18}=0, a_{26,18}=1, a_{28,18}=0$
31	$a_{31,11}=1, a_{26,11}=0, a_{27,11}=1, a_{29,11}=0$
95	$a_{92,32}=1$

Table 8. Sufficient conditions on a_i for 21-95 rounds of the differential path in Table 5.