# MultiCollision Attack on the Compression Functions of MD4 and 3-Pass HAVAL

Hongbo Yu[1] and Xiaoyun Wang[2⋆]

[1] School of Mathematics and System Sciences,
Shandong University, Jinan 250100, China
yhb@mail.sdu.edu.cn

[2] Tsinghua University and Shandong University, China
xiaoyunwang@tsinghua.edu.cn, xywang@sdu.edu.cn

**Abstract.** In this paper, we present a new type of MultiCollision attack on the compression functions both of MD4 and 3-Pass HAVAL. For MD4, we utilize two feasible different collision differential paths to find a 4-collision with $2^{19}$ MD4 computations. For 3-Pass HAVAL, we present three near-collision differential paths to find a 8-NearCollision with $2^9$ HAVAL computations.

**Keywords:** Hash function, MultiCollision, NearCollision, differential path, sufficient condition.

## 1 Introduction

Recently, the cryptanalysis on hash functions has become a hot topic within the cryptographic community. Most existing hash functions have succumbed to the modular differential attack announced two years ago [9, 11, 12, 10, 13, 14].

For an ideal secure hash function with $n$-bit output, the complexity to find a pairwise collision is about $O(2^{n/2})$ computations, and to find a $k$(*multi*)-collision needs about $O(2^{n(k-1)/k})$ computations. Here a $k$-collision consists of $k$ different messages which are compressed to the same hash value. At Crypto'04[4], using the flaw of the iterated structure of the hash functions, Joux proposed a method to construct $2^t$-collisions based on the pairwise collisions. Joux showed that for any iterated hash function it is relatively easy to find a $2^t$-collision and it only costs $t$ times as much as that of finding an ordinary pairwise collision. Based on the result, Joux proved that the concatenation of several hash functions does not increase their security. In 2005[3], Nandi and Stinson extended Joux's technique to handle iterated hash functions in which each message block is used at most twice. In FSE 2006[5], Hoch and Shamir considered the general case and proved

⋆ Supported by the National Natural Science Foundation of China(NSFC Grant No.90604036) and 973 Project(No.2007CB807900)

that even if allowing each iterated hash function to scan the input multiple times in an arbitrary expanded order, their concatenation is not stronger than a single function.

Motivated by Shamir's talk [5] and Joux's method, we provide a new attack to build the MultiCollisions and Multi-NearCollisions towards hash functions directly instead of the combined MultiCollisions from the pairwise collisions. The Multi-NearCollisions is a generalization for the pairwise near collisions presented by Biham and Chen[1]. The difference between two types of MultiCollisions are shown in Fig.1 and Fig.2 respectively.

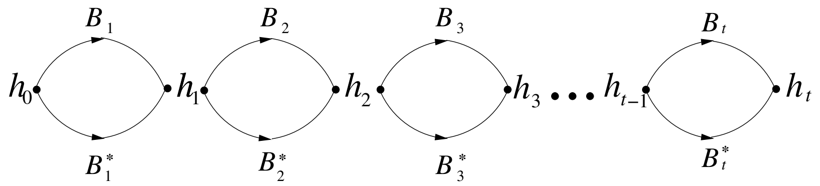


**Fig. 1.** Joux's $2^t$-Collisions Construction. The $2^t$ messages are of the form $(b_1, b_2, ..., b_t)$ where $b_i$ is one of the two blocks $B_i$ and $B_i^*$.

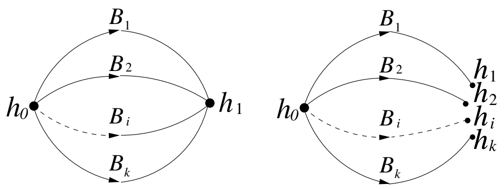


**Fig. 2.** Our $k$-Collisions Construction. The left denotes the MultiCollision, and the right is the Multi NearCollision. Each of $B_i$ is a one-block message.

The paper is organized as follows. In Section 2, we give a brief description of MD4 and 3-Pass HAVAL compression functions. In section 3, we recall the modular differential attack on hash functions which is used as a fundamental tool to find MultiCollisons. In section 4, we propose our new MultiCollision and Multi-NearCollision attack. The details for finding 4-collsions on MD4 and 8-NearCollisons on 3-Pass HAVAL are introduced in section 5 and 6 respectively. Finally we conclude the paper in section 7.

## 2 Description of MD4 and 3-Pass HAVAL

In this paper, we study the MultiCollisions for MD4 compression function and the Multi-NearCollisions for 3-Pass HAVAL compression function, so we only give a brief description for their compression functions.

### 2.1 MD4 Compression Function

The MD4 compression function takes a 128-bit chaining value and a 512-bit message block as the input value, process 48 step operations and outputs a 128 -bit chaining value as hash value. For one 512-bit block $M = (m_0, m_1, ..., m_{15})$, the compressing process is as follows:

1. Let $(aa, bb, cc, dd)$ be the input 128-bit chaining variable.

$$a \longleftarrow aa,\ b \longleftarrow bb,\ c \longleftarrow cc,\ d \longleftarrow dd$$

2. Perform the following 48 steps (three rounds):
For $i$=0, 1, 2
  For $j$=0, 1, 2, 3

$$\begin{aligned} a &:= (a + \phi_i(b, c, d) + w_{i,4j} + k_i) \lll s_{i,4j} \\ d &:= (d + \phi_i(a, b, c) + w_{i,4j+1} + k_i) \lll s_{i,4j+1} \\ c &:= (c + \phi_i(d, a, b) + w_{i,4j+2} + k_i) \lll s_{i,4j+2} \\ b &:= (b + \phi_i(c, d, a) + w_{i,4j+3} + k_i) \lll s_{i,4j+3} \end{aligned}$$

$s_{i,4j+t}$ $(t = 0,\ 1,\ 2,\ 3)$ are step-dependent constants. $w_{i,4j+t}$ is a message word and $k_i$ is a fixed constant for every round. Symbol $\lll s$ represents the circular shift $s$ bit positions to the left. And symbol $+$ denotes addition modulo $2^{32}$. The details of the message order and shift positions can be referred to paper[6]. The round functions $\phi_0$, $\phi_1$ and $\phi_2$ are defined as:

$$\begin{aligned} &\phi_0(x, y, z) = (x \wedge y) \vee (\neg x \wedge z) \\ &\phi_1(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \\ &\phi_2(x, y, z) = x \oplus y \oplus z \end{aligned}$$

3. Add $a$, $b$, $c$ and $d$ respectively to the input chaining value.

$$\begin{aligned} aa &:= a + aa \\ bb &:= b + bb \\ cc &:= c + cc \\ dd &:= d + dd \end{aligned}$$

4. $H(M) := aa \| bb \| cc \| dd$. Here, $\|$ denotes the bit concatenation.

Because searching the differential paths in our paper is greatly based on the properties of the round functions $\phi_1$ and $\phi_2$ which have been summarized in [11, 16, 8]. For convenience, we list them in the Appendix (See Table 3) as reference.

### 2.2 3-Pass HAVAL Compression Function

The 3-Pass HAVAL compression function takes a 256-bit chaining value and a 1024-bit message block $M = (m_0, m_1, ..., m_{31})$ as input value and outputs a 256-bit chaining value. The compressing process is described as follows:

1. Let $(aa, bb, cc, dd, ee, ff, gg, hh)$ be the 256-bit input chaining value. Initialize chaining variables $(a,\ b,\ c,\ d,\ e,\ f,\ g,\ h)$ as $(aa, bb, cc, dd, ee, ff, gg, hh)$.
2. Perform the following 96 steps:
   For $i$=0,1,2
       For $j = 0$ to 31
           $p := f_{i+1}(g, f, e, d, c, b, a)$
           $r := (p \gg 7) + (h \gg 11) + m_{ord(i,j)} + k_{i,j}$
           $h := g$
           $g := f$
           $f := e$
           $e := d$
           $d := c$
           $c := b$
           $b := a$
           $a := r$
   The operation in each step employs a constant $k_{j,i}$(See ref.[15]). Symbol $\gg s$ represents the circular shift $s$ bit positions to the right. The orders of message words in each Pass can be referred to [15]. The round functions $f_1$, $f_2$ and $f_3$ are defined as follows:

$$f_1(g, f, e, d, c, b, a) = cd \oplus ag \oplus bf \oplus ce \oplus e$$
$$f_2(g, f, e, d, c, b, a) = adf \oplus bcf \oplus ef \oplus ef \oplus ac \oplus df \oplus bd \oplus bc \oplus fg \oplus g$$
$$f_3(g, f, e, d, c, b, a) = def \oplus cf \oplus be \oplus dg \oplus ad \oplus a$$

3. Add $a$, $b$, $c$, $d$, $e$, $f$, $g$, $h$ respectively to the input value.
   $aa := a + aa$, $bb := b + bb$, ... ..., $hh := h + hh$
4. $H(M) := hh\|gg\|ff\|ee\|dd\|cc\|bb\|aa$.

Some main properties of the round function $f_1$ which are used to find differential paths are listed in the Appendix (See Table 4).

## 3 Modular Differential Attack on Hash Functions

In this section, we take the MD4 as an example to outline the modular differential attack developed by Wang et al.

### 3.1 Selecting a Message Difference

The first step of the modular differential attack is to select an appropriate message difference which determines the success probability of the attack. The choice of the message differences is based on the attack purpose. For example, if we want to find a differential path for collisions, we can select the message differences as in paper [11]. These message differences not only result in a 5-step local collision in the third round, but also can decide a potential internal collision located at the first round and the previous steps of the second round. Thus, selecting the message differences is a key step to produce a possible collision path with high probability. If we want to apply the second-preimage attack or recover the keys of the MACs based on MD4, we can select the message differences which lead to a differential path [16] with minimal sufficient conditions in total.

### 3.2 Searching the Differential Path

The search of the differential path for some hash functions is really a hard work. We can utilize the properties of the round functions to produce some wanted bit differences and cancel the unwanted non-zero bit differences or message bit differences. Another important technique is to introduce the bit carries which can produce the above wanted bit differences. For the weak hash function MD4, it's easy to find a differential path by hand or by the computer searching [8]. Because of the computation limitation, the number of bit carries must be fixed. So there does not appear computer searching method to find a differential path for those stronger hash functions such as MD5 and SHA-1 etc.

### 3.3 Determining the Chaining Variable Conditions

In the process of the differential path searching, the chaining variable conditions can be determined. A feasible differential path implies that all the chaining variable conditions deduced from the path don't contradict each other. It means that if a message $M$ satisfies all the chaining variable conditions, $M$ and $M+\Delta M$ ($\Delta M$ is a fixed message difference) must collide. So these conditions are called the sufficient conditions.

### 3.4 Message Modification

Once the collision differential path and the corresponding sufficient conditions are determined, the remaining is how to find a message $M$ so that $M$ satisfies all the chaining variable conditions. Usually for a random message $M$, $M$ and $M+\Delta M$ cannot compose collisions because of the large amount of sufficient conditions. If a condition is inconsistent with that of the sufficient conditions, we call it a wrong condition. According to the chaining variable conditions distribution, we can adopt different message modification techniques to force the modified message $M$ to satisfy more sufficient conditions. For the conditions in the first round, we can implement the basic message modification technique to correct the wrong conditions. And for the conditions in the second round, the advanced message modification can be applied to correct part of wrong conditions. Usually, more conditions in the second round can be corrected by employing more fine and complex advanced message modifications. The detail of the techniques can be seen in [11, 12, 16].

## 4 New MultiCollision and Multi NearCollision Attack

In this section, we describe our MultiCollision and Multi-NearCollision attack on hash functions. Given two different collision differential paths for a hash function, if two sets of their sufficient conditions are not contrary each other, we will show how to utilize two collision paths to produce 4-collisions.

Provided that the first collision differential path $P_1$ corresponds to the message difference $\Delta M_1$, $(M, M+\Delta M_1)$ is a collision under the sufficient conditions $C_1$. The second collision differential path $P_2$ corresponding to the message difference $\Delta M_2$, $(M, M+\Delta M_2)$ is a collision under the sufficient conditions $C_2$.

If there are no contradictory conditions between $C_1$ and $C_2$, we set up $C$=$C_1\cup C_2$. It is clear that, if $M$ satisfies all conditions in $C$, $(M, M+\Delta M_1)$ is a collision that obeys the differential path $P_1$, $(M, M+\Delta M_2)$ is also a collision simultaneously which obeys the second collision path $P_2$. So, $(M, M+\Delta M_1, M+\Delta M_2)$ composes a 3-collision.

Let's look at another message $M+\Delta M_1+\Delta M_2$. If the message $M+\Delta M_1$ satisfies all the conditions $C_2$, $(M+\Delta M_1, M+\Delta M_1+\Delta M_2)$ is a collision corresponding to the path $P_2$. On the other hand, when the message $M+\Delta M_2$ satisfies all the conditions $C_1$, $(M+\Delta M_2, M+\Delta M_2+\Delta M_1)$ is a collision corresponding to the path $P_1$. This is an interesting phenomenon. For each case, $(M, M+\Delta M_1, M+\Delta M_2, M+\Delta M_1+\Delta M_2)$ consists of a 4-collision. For any two collision differential paths, even if the two sets of sufficient conditions have no contrary conditions, the message $M+\Delta M_1+\Delta M_2$ may not collide with other three messages because of a few subtle conditions on intersect bits between the two paths.

We generalize the above MultiCollisions to $k$-collisions generated by $t$ multi collision differential paths where $(k \leq 2^t)$. Firstly, we select $t$ message differences $\Delta M_i, i=0,1,..t-1$. For each $\Delta M_i$, search a feasible collision differential path and deduce its corresponding condition set $C_i$. If there are some contradictory conditions among $C_i, i=0,1,..t-1$, adjust some differential paths so that no contrary condition occurs. Finally find a message $M$ which satisfies all conditions $C=\cup C_i$. Then the messages $(M, M+\Delta M_0, M+\Delta M_1, ...M+\Delta M_{t-1}))$ is a $t$-collision. Perfectly, if the $t$ differential paths are independent completely (i.e. there is no intersect bit among $C_i, i=0,1,..t-1$), all the messages in $\{M+\Delta M \mid \Delta M$ is a linear expression of $\Delta M_i, i=0,1,..t-1\}$ collide with $M$, and produce $2^t$-collision.

Similarly, we can find the Multi-NearCollisions by the above method.

## 5 Finding 4-Collisions on MD4

In this section, we construct a 4-collisions for the MD4 compression function which is illustrated in Fig.3.

We select the first message difference

$$\Delta M_1 = (0,0,0,0,2^{25},0,0,0,0,0,0,0,0,0,0,0),$$

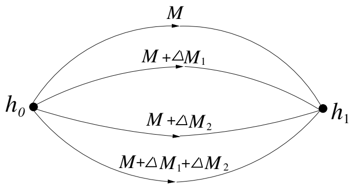


**Fig. 3.** The 4-Collisions construction on MD4 compression function. Each of $M_i$ is a one-block message.

and decide a collision differential path and its sufficient conditions (See Table 5). In fact, this collision differential path is a special instance of the 64 differential paths in [16]. The path has the least conditions among all the known MD4 collision paths.

The collision differential path in paper [11] is selected as the second differential path which is most efficient to find the pairwise collisions so far. The message difference is taken as

$$\Delta M_2 = (0, 2^{31}, -2^{28} + 2^{31}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2^{16}, 0, 0, 0).$$

We make a little simplification for the original differential path in [11] and get a new one which is showed in Table 6. Combining the two sets of conditions in the column 3 of Table 5 and Table 6, we get the final conditions (See Table 7).

The rest work is to find a one-block message $M$ which satisfies all the conditions in Table 7 so that ($M$, $M + \Delta M_1$, $M + \Delta M_2$ ,$M + \Delta M_1 + \Delta M_2$) composes a 4-collision.

It is easy to correct all the conditions in the first 16 steps by the basic message modifications. There are 63 conditions in step 17-48, we fulfill the advanced message modification to force all the 44 conditions in step 17-23 hold. All the precise details for advanced message modifications are omitted, and the main techniques are shown in [11, 12, 16].

After the advanced message modifications, there are 19 conditions left. So, the probability that the modified message $M$ satisfies all the conditions in Table 7 is improved to $2^{-19}$. By computer searching, it is very easy to find 4-collisions, and an example of 4-collisions is given in Table 1.

For any initial value, we can build $4^t$-collisions for MD4 as Joux $2^t$-collisions construction. Let $h$ be a hash function, and $H$ is its compression function. $A$ denotes the attack algorithm for MD4 4-collisions introduced above. The $4^t$-collisions construction is as follows (See Fig.4):

- Let $h_0$ be the initial value of h.

- For $i$ from 0 to $t-1$ do:

  - Call $A$ and find four different 512-bit messages $B_{i,0}$, $B_{i,1}$, $B_{i,2}$ and $B_{i,3}$ such that $H(h_{i-1}, B_{i,0})$=$H(h_{i-1}, B_{i,1})$=$H(h_{i-1}, B_{i,2})$=$H(h_{i-1}, B_{i,3})$.

**Table 1.** A 4-collision for MD4 compression function. The $IV$ and messages are hexadecimal format.

| IV | 67452301 efcdab89 98badcfe 10325476 |
|---|---|
| $M$ | 74c5f8d6 33fc9eaa 0fd0a9e2 e83340d1 246c716a c0d1931b ef06af4c e7a20583<br>898483db 0d9a1026 fd62bb0f bb29de31 886af4fa 5c772a7d 6ce0f4fb 6a9c8ce8 |
| $M+\Delta M_1$ | 74c5f8d6 33fc9eaa 0fd0a9e2 e83340d1 266c716a c0d1931b ef06af4c e7a20583<br>898483db 0d9a1026 fd62bb0f bb29de31 886af4fa 5c772a7d 6ce0f4fb 6a9c8ce8 |
| $M+\Delta M_2$ | 74c5f8d6 b3fc9eaa 7fd0a9e2 e83340d1 246c716a c0d1931b ef06af4c e7a20583<br>898483db 0d9a1026 fd62bb0f bb29de31 8869f4fa 5c772a7d 6ce0f4fb 6a9c8ce8 |
| $M+\Delta M_1$<br>$+\Delta M_2$ | 74c5f8d6 b3fc9eaa 7fd0a9e2 e83340d1 266c716a c0d1931b ef06af4c e7a20583<br>898483db 0d9a1026 fd62bb0f bb29de31 8869f4fa 5c772a7d 6ce0f4fb 6a9c8ce8 |
| Common output value | cbe16ea1 2d600674 3b42a32d a1458b54 |

- Let $h_i = H(h_{i-1}, B_i)$.

– Output the $4^t$ messages of the form $(b_0, b_1, ..., b_{t-1})$ where $b_i$ is one of the four messages $B_{i,j} (j = 0, 1, 2, 3)$, i=0,1,..t-1.

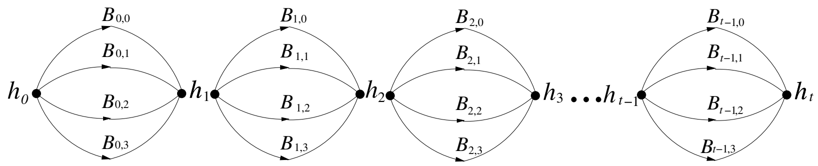


**Fig. 4.** Our $4^t$-Collisions construction. Each of $B_{i,j}$ is a one-block message.

## 6 Finding 8-NearCollisions on 3-Pass HAVAL

In order to construct 8-NearCollisions for 3-Pass HAVAL compression function, we need to find three feasible near-collision differential paths. One of them is shown in Table 8 where the message difference is selected as

$$\Delta M_1 = (\Delta m_i)_{0 \le i \le 31},\ \Delta m_i = 2^{11} \text{ when } i = 5, \text{ else } \Delta m_i = 0.$$

If a message $M$ satisfies all 73 conditions in column 4 of Table 8, the value of $H(M)$ is almost the same as $H(M + \Delta M)$ except one bit.

In fact, a similar differential path can be constructed for any one of the 48 message differences $\Delta m_5 = \pm 2^j (0 \leq j \leq 31,\ j \neq 3, 4, 5, 10, 14, 15, 16, 31)$. We select two other differential paths corresponding to $\Delta m_5$=$2^{19}$ and $2^{27}$ respectively, and denote their message differences as $\Delta M_2$ and $\Delta M_3$. So we get three near-collision differential paths.

In order to implement the message modification, we sum up three sets of sufficient conditions which are shown in Table 9. There are $73 \times 3$ conditions totally in which 210 conditions focus on the first round(steps 1-32). So only using the basic message modification, the probability to find a message which satisfies all the 219 conditions is high to $2^{-9}$.

It is remarked that there are 6 conditions on $IV(a_0)$ which is contrary to the original initial value $IV_0$ of HAVAL. In order to avoid of this situation, an extra message block $M_0$ is needed so that the hash value $H(M_0)$ satisfies these 6 conditions. Under the new initial value $IV_1(H(M_0))$, we find 8 different messages $M$, $M + \Delta M_0$, $M + \Delta M_1$, $M + \Delta M_2$, $M + \Delta M_0 + \Delta M_1$, $M + \Delta M_0 + \Delta M_2$, $M + \Delta M_1 + \Delta M_2$ $M + \Delta M_0 + \Delta M_1 + \Delta M_2$. Their hash values are almost the same besides 3 bits. Our results are shown in Table 2. The hash values are expressed in binary format(the most left bit is MSB) to reveal three different bits positions. The 8-NearCollision exactly traverses all the 8 cases corresponding to these three positions.

## 7 Conclusion

We have presented a dedicated MultiCollision attack on the compression functions of MD4, and Multi-NearCollision attack on 3-Pass HAVAL. In fact, the attack is available to find real MultiCollsions for 3-Pass HAVAL. The further work needs to find two or more new collision differential paths carefully. We believe that the MultiCollisions bring more dangers for practical application of these hash functions.

## 8 Acknowledgments

We wish to thank Adi Shamir for his report presented in Tsinghua University which motivated us to start this research.

**Table 2.** The 8(Multi) Near-collisions attack example

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $IV_0$ | 243f6a88 | 85a308d3 | 13198a2e | 03707344 | a4093822 | 299f31d0 | 082efa98 | ec4e6c89 |
| $M_0$ | e6c99bc9 | c99bc914 | 9bc914d8 | c914d80b | 14d80bf6 | d80bf605 | 0bf605ef | f605ef83 |
| | 05ef831b | ef831b24 | 831b2486 | 1b24868f | 24868fd3 | 868fd399 | 8fd39945 | d399459d |
| | 99459d9c | 459d9cb7 | 9d9cb7ba | 9cb7ba3f | b7ba3f31 | ba3f31dd | 3f31dd06 | 31dd067f |
| | dd067f7e | 067f7ebb | 7f7ebb63 | 7ebb63e8 | bb63e8b9 | 63e8b986 | e8b986dc | b986dc14 |
| $IV_1$ | 0444e787 | 978e0d0a | c408c64f | 74a629f6 | ee1eb57d | fdb20640 | 6126dd36 | 4563c119 |
| $M$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 14490ac3 | 30523a85 | 12cbd516 |
| | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_1$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 144912c3 | 30523a85 | 12cbd516 |
| | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_2$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 14510ac3 | 30523a85 | 12cbd516 |
| | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_3$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 1c490ac3 | 30523a85 | 12cbd516 |
| | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_1$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 145112c3 | 30523a85 | 12cbd516 |
| $+\Delta M_2$ | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_1$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 1c4912c3 | 30523a85 | 12cbd516 |
| $+\Delta M_3$ | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_2$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 1c510ac3 | 30523a85 | 12cbd516 |
| $+\Delta M_3$ | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| $M+\Delta M_1$ | f09e3e1e | 862e1e8a | 3e1e8a1c | 0e8a1c49 | 88ca9872 | 1c5112c3 | 30523a85 | 12cbd516 |
| $+\Delta M_2$ | c3e51637 | cd35b784 | 163503b0 | a78401f9 | e5539c0d | b37acef6 | 7a0e9574 | cdf616d7 |
| $+\Delta M_3$ | f5d83836 | 16d7f41a | 99f81c43 | f61c4105 | 1c0105da | 4105da86 | 05d88403 | 9a860533 |
| | c6453355 | c53355d0 | 3355d113 | 55d0d398 | 90d397dd | d3981d16 | 97dd1617 | 1d16175b |
| Near-collision value | **a:** 10110000011011011010110110001011 | | | | **e:** 11110011110100101000011010101101 | | | |
| | **b**: 1110?1111000?1011010?01000101001 | | | | **f:** 10010000111000001010000101101010 | | | |
| | **c**: 00010011011011000100100011101101 | | | | **g:** 01100111101001000110110111110111 | | | |
| | **d**: 11001101110000111101101000001111 | | | | **h:** 00001011010101101100111010011010 | | | |

## References


1. E.Biham and R.Chen, Near-Collisions of SHA-0, Crypto 2004, LNCS 3152, pp.290-305.
2. B. Rompay, A. Biryukov, B. Preneel, and J.Vandewalle, Cryptanalysis of 3-Pass HAVAL, ASIACRYPT 2003, LNCS 2894, pp. 228-245.
3. M. Nandi and D. R. Stinson, Multicollision Attacks on a Class of Hash Functions, IACR preprint archive, 2005.
4. A. Joux, Multicollisions in Iterated Hash Functions, Crypto 2004, LNCS 3152, pp. 306-316.
5. Jonathan J. Hoch and A. Shamir, Breaking the ICE - Finding Multicollisions in Iterated Concatenated and Expanded (ICE) Hash Functions, FSE 2006, LNCS 4047, pp. 179-194.
6. R.L.Rivest, The MD4 Message Digest Algorithm, Advances in Cryptology, Crypto90, Springer-Verlag, 1991, 303-311.
7. R.L.Rivest, The MD5 message-digest algorithm, Request for Comments(RFC 1320), Internet Activities Board, Internet Privacy Task Force, 1992.
8. M.Schlaffer, E.Oswald, Searching for Differential Paths in MD4, FSE 2006, LNCS 4047, pp. 242-261.
9. X.Y.Wang, F.D.Feng, X.J.Lai and H.B.Yu, Collisions for Some Hash Functions MD4, MD5, HAVAL-128, RIPEMD. http:eprint.iacr.org/2004/264/199.pdf.
10. X.Y.Wang, F.D.Feng, X.Yu, An attack on HAVAL function HAVAL-128. Science in China Ser. F Information Sciences 2005 Vol.48, No.5,1-12.
11. X.Y.Wang, X.J.Lai etc, Cryptanalysis for Hash Functions MD4 and RIPEMD, Eurocrypt'05, LNCS 3494, pp.1-18.
12. X.Y.Wang, H.B.Yu, How to Break MD5 and Other Hash Functions, Eurocrypt'05, LNCS 3494, pp.19-35.
13. X.Y.Wang, H.B.Yu, Y.Lisa, Efficient Collision Search Attacks on SHA-0, Crypto'05, LNCS 3621, pp.1-16, 2005.
14. X.Y.Wang, Y.lisa, H.B.Yu, Finding collisions on the Full SHA-1, Crypto'05, LNCS 3621, pp.17-36.
15. Y.Zheng, J.Pieprzyk and J.Seberry, HAVAL–A One-way Hashing Algorithm with Variable Length of Output, Advances in Cryptology, Auscrypto'92 Proceedings, pp. 83-104.
16. H.B.Yu, G.L.Wang, G.Y.Zhang and X.Y.Wang, The Second-Preimage Attack on MD4, CANS 2005, LNCS 3810, pp.1-12.

## Appendix

**Table 3.** The property for the round function $\phi_1$ and $\phi_2$ of MD4. $\Delta x = 1$ denotes $x$ changes from 0 to 1, $\Delta x = -1$ denotes $x$ changes from 1 to 0.

| $\Delta x$ | $\Delta y$ | $\Delta z$ | $\Delta\phi_1 = 0$ | $\Delta\phi_1 = 1$ | $\Delta\phi_1 = -1$ | $\Delta\phi_2 = 0$ | $\Delta\phi_2 = 1$ | $\Delta\phi_2 = -1$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | – | – | 1 | – | – |
| 0 | 0 | 1 | $x = 1$ | $x = 0$ | – | $x = y$ | $x \neq y$ | – |
| 0 | 0 | -1 | $x = 1$ | – | $x = 0$ | $x = y$ | – | $x \neq y$ |
| 0 | 1 | 0 | $x = 0$ | $x = 1$ | – | $x = z$ | $x \neq z$ | – |
| 0 | 1 | 1 | – | 1 | – | – | 1 | – |
| 0 | 1 | -1 | – | $x = 1$ | $x = 0$ | 1 | – | – |
| 0 | -1 | 0 | $x = 0$ | – | $x = 1$ | $x = z$ | – | $x \neq z$ |
| 0 | -1 | 1 | – | $x = 0$ | $x = 1$ | 1 | – | – |
| 0 | -1 | -1 | – | – | 1 | – | – | 1 |
| 1 | 0 | 0 | $y = z$ | $y = 1,\ z = 0$ | $y = 0,\ z = 1$ | $y = z$ | $y \neq z$ | – |
| 1 | 0 | 1 | $y = 0$ | $y = 1$ | – | – | 1 | – |
| 1 | 0 | -1 | $y = 1$ | – | $y = 0$ | 1 | – | – |
| 1 | 1 | 0 | $z = 1$ | $z = 0$ | – | – | 1 | – |
| 1 | 1 | 1 | – | 1 | – | – | 1 | – |
| 1 | 1 | -1 | 1 | – | – | – | 1 | – |
| 1 | -1 | 0 | $z = 0$ | – | $z = 1$ | 1 | – | – |
| 1 | -1 | 1 | 1 | – | – | – | 1 | – |
| 1 | -1 | -1 | – | – | 1 | – | – | 1 |
| -1 | 0 | 0 | $y = z$ | $y = 0,\ z = 1$ | $y = 1,\ z = 0$ | $y = z$ | – | $y \neq z$ |
| -1 | 0 | 1 | $y = 1$ | $y = 0$ | – | 1 | – | – |
| -1 | 0 | -1 | $y = 0$ | – | $y = 1$ | – | – | 1 |
| -1 | 1 | 0 | $z = 0$ | $z = 1$ | – | 1 | – | – |
| -1 | 1 | 1 | – | 1 | – | – | 1 | – |
| -1 | 1 | -1 | 1 | – | – | – | – | 1 |
| -1 | -1 | 0 | $z = 1$ | – | $z = 0$ | – | – | 1 |
| -1 | -1 | 1 | 1 | – | – | – | – | 1 |
| -1 | -1 | -1 | – | – | 1 | – | – | 1 |

**Table 4.** Some properties for the first round function $f_1$ of the 3-Pass Haval

| $\Delta a$ | $\Delta b$ | $\Delta c$ | $\Delta d$ | $\Delta e$ | $\Delta f$ | $\Delta g$ | $\Delta f_1 = 0$ | $\Delta f_1 = 1$ | $\Delta f_1 = -1$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | – | – |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | $g = 0$ | $g = 1,\ cd + bf + ce + e = 0$ | $g = 1,\ cd + bf + ce + e = 1$ |
| -1 | 0 | 0 | 0 | 0 | 0 | 0 | $g = 0$ | $g = 1,\ cd + bf + ce + e = 1$ | $g = 1,\ cd + bf + ce + e = 0$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | $f = 0$ | $f = 1,\ cd + ag + ce + e = 0$ | $f = 1,\ cd + ag + ce + e = 1$ |
| 0 | -1 | 0 | 0 | 0 | 0 | 0 | $f = 0$ | $f = 1, cd + ag + ce + e = 1$ | $f = 1,\ cd + ag + ce + e = 0$ |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | $e = d$ | $e \neq d,\ ag + bf + e = 0$ | $e \neq d,\ ag + bf + e = 1$ |
| 0 | 0 | -1 | 0 | 0 | 0 | 0 | $e = d$ | $e \neq d,\ ag + bf + e = 1$ | $e \neq d,\ ag + bf + e = 0$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | $c = 0$ | $c = 1,\ ag + bf + ce + e = 0$ | $c = 1,\ ag + bf + ce + e = 1$ |
| 0 | 0 | 0 | -1 | 0 | 0 | 0 | $c = 0$ | $c = 1,\ ag + bf + ce + e = 1$ | $c = 1,\ ag + bf + ce + e = 0$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | $c = 1$ | $c = 0,\ cd + ag + bf = 0$ | $c = 0,\ cd + ag + bf = 1$ |
| 0 | 0 | 0 | 0 | -1 | 0 | 0 | $c = 1$ | $c = 0,\ cd + ag + bf = 1$ | $c = 0,\ cd + ag + bf = 0$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | $b = 0$ | $b = 1,\ cd + ag + ce + e = 0$ | $b = 1,\ cd + ag + ce + e = 1$ |
| 0 | 0 | 0 | 0 | 0 | -1 | 0 | $b = 0$ | $b = 1,\ cd + ag + ce + e = 1$ | $b = 1,\ cd + ag + ce + e = 0$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | $a = 0$ | $a = 1,\ cd + bf + ce + e = 0$ | $a = 1,\ cd + bf + ce + e = 1$ |
| 0 | 0 | 0 | 0 | 0 | 0 | -1 | $a = 0$ | $a = 1,\ cd + bf + ce + e = 1$ | $a = 1,\ cd + bf + ce + e = 0$ |

**Table 5.** The collision differential path 1 of MD4.

| Step | Output for $M$ | $m_i$ | $s_i$ | $\Delta m_i$ | Output for $M'$ | Sufficient conditions |
|---|---|---|---|---|---|---|
| 1 | $a_1$ | $m_0$ | 3 | | | |
| 2 | $d_1$ | $m_1$ | 7 | | | |
| 3 | $c_1$ | $m_2$ | 11 | | | |
| 4 | $b_1$ | $m_3$ | 19 | | | |
| 5 | $a_2$ | $m_4$ | 3 | $2^{25}$ | $a_2[29]$ | $a_{2,29} = 0$ |
| 6 | $d_2$ | $m_5$ | 7 | | $d_2$ | $b_{1,29} = c_{1,29}$ |
| 7 | $c_2$ | $m_6$ | 11 | | $c_2$ | $d_{2,29} = 0$ |
| 8 | $b_2$ | $m_7$ | 19 | | $b_2$ | $c_{2,29} = 1$ |
| 9 | $a_3$ | $m_8$ | 3 | | $a_3[32]$ | $a_{3,32} = 0$ |
| 10 | $d_3$ | $m_9$ | 7 | | $d_3$ | $b_{2,32} = c_{2,32}$ |
| 11 | $c_3$ | $m_{10}$ | 11 | | $c_3[-11]$ | $c_{3,11} = 1$, $d_{3,32} = 1$ |
| 12 | $b_3$ | $m_{11}$ | 19 | | $b_3$ | $c_{3,32} = 1$, $d_{3,11} = a_{3,11}$ |
| 13 | $a_4$ | $m_{12}$ | 3 | | $a_4[3]$ | $a_{4,3} = 0$, $b_{3,11} = 0$ |
| 14 | $d_4$ | $m_{13}$ | 7 | | $d_4$ | $b_{3,3} = c_{3,3}$, $a_{4,11} = 1$ |
| 15 | $c_4$ | $m_{14}$ | 11 | | $c_4[-22]$ | $c_{4,22} = 1$, $d_{4,3} = 0$ |
| 16 | $b_4$ | $m_{15}$ | 19 | | $b_4$ | $d_{4,22} = a_{4,22}$, $c_{4,3} = 1$ |
| 17 | $a_5$ | $m_0$ | 3 | | $a_5[6]$ | $a_{5,6} = 0$, $b_{4,22} = d_{4,22}$ |
| 18 | $d_5$ | $m_4$ | 5 | $2^{25}$ | $d_5[11, 31]$ | $d_{5,11} = 0$, $d_{5,31} = 0$, $a_{5,22} = b_{4,22}$, $b_{4,6} = c_{4,6} + 1$ |
| 19 | $c_5$ | $m_8$ | 9 | | $c_5[-31]$ | $c_{5,31} = 1$, $a_{5,11} = b_{4,11}$, $d_{5,6} = b_{4,6}$, $a_{5,31} = b_{4,31}$ |
| 20 | $b_5$ | $m_{12}$ | 13 | | $b_5$ | $c_{5,6} = d_{5,6}$, $c_{5,11} = a_{5,11}$ |
| 21 | $a_6$ | $m_1$ | 3 | | $a_6[9]$ | $a_{6,9} = 0$, $b_{5,11} = c_{5,11}$ |
| 22 | $d_6$ | $m_5$ | 5 | | $d_6[16]$ | $d_{6,16} = 0$, $b_{5,9} = c_{5,9}$, $a_{6,31} = b_{5,31} + 1$ |
| 23 | $c_6$ | $m_9$ | 9 | | $c_6[8, -9]$ | $c_{6,8} = 0$, $c_{6,9} = 1$, $d_{6,9} = b_{5,9}$, $a_{6,16} = b_{5,16}$ |
| 24 | $b_6$ | $m_{13}$ | 13 | | $b_6$ | $d_{6,8} = a_{6,8}$, $c_{6,16} = a_{6,16}$ |
| 25 | $a_7$ | $m_2$ | 3 | | $a_7$ | $b_{6,9} = d_{6,9} + 1$, $b_{6,8} = d_{6,8}$, $b_{6,16} = c_{6,16}$ |
| 26 | $d_7$ | $m_6$ | 5 | | $d_7[21]$ | $a_{7,8} = b_{6,8}$, $a_{7,9} = b_{6,9}$, $d_{7,21} = 0$ |
| 27 | $c_7$ | $m_{10}$ | 9 | | $c_7[-17]$ | $c_{7,17} = 1$, $a_{7,21} = b_{6,21}$ |
| 28 | $b_7$ | $m_{14}$ | 13 | | $b_7$ | $d_{7,17} = a_{7,17}$, $c_{7,21} = a_{7,21}$ |
| 29 | $a_8$ | $m_3$ | 3 | | $a_8$ | $b_{7,17} = d_{7,17}$, $b_{7,21} = c_{7,21}$ |
| 30 | $d_8$ | $m_7$ | 5 | | $d_8[26]$ | $d_{8,26} = 0$, $a_{8,17} = b_{7,17}$ |
| 31 | $c_8$ | $m_{11}$ | 9 | | $c_8[-26]$ | $c_{8,26} = 1$, $a_{8,26} = b_{7,26}$ |
| 32 | $b_8$ | $m_{15}$ | 13 | | $b_8$ | |
| 33 | $a_9$ | $m_0$ | 3 | | $a_9$ | |
| 34 | $d_9$ | $m_8$ | 9 | | $d_9$ | $a_{9,26} = b_{8,26}$ |
| 35 | $c_9$ | $m_4$ | 11 | $2^{25}$ | $c_9$ | |
| 36 | $b_9$ | $m_{12}$ | 15 | | $b_9$ | |

**Table 6.** The collision differential path 2 of MD4 .

| Step | Output for $M$ | $m_i$ | $s_i$ | $\Delta m_i$ | Output for $M'$ | Sufficient conditions |
|---|---|---|---|---|---|---|
| 1 | $a_1$ | $m_0$ | 3 | | | |
| 2 | $d_1$ | $m_1$ | 7 | $2^{31}$ | $d_1[7]$ | $d_{1,7}=0$ |
| 3 | $c_1$ | $m_2$ | 11 | $-2^{28}+2^{31}$ | $c_1[-8,11]$ | $c_{1,8}=1$, $c_{1,11}=0$, $a_{1,7}=b_{0,7}$ |
| 4 | $b_1$ | $m_3$ | 19 | | $b_1[26]$ | $b_{1,26}=0$, $c_{1,7}=1$, $d_{1,8}=a_{1,8}$, $d_{1,11}=a_{1,11}$ |
| 5 | $a_2$ | $m_4$ | 3 | | $a_2$ | $c_{1,26}=d_{1,26}$, $b_{1,8}=0$, $b_{1,11}=0$, $b_{1,7}=1$ |
| 6 | $d_2$ | $m_5$ | 7 | | $d_2[14]$ | $d_{2,14}=0$, $a_{2,26}=0$, $a_{2,8}=1$, $a_{2,11}=1$ |
| 7 | $c_2$ | $m_6$ | 11 | | $c_2[-19,22]$ | $c_{2,19}=1$, $c_{2,22}=0$, $a_{2,14}=b_{1,14}$, $d_{2,26}=1$ |
| 8 | $b_2$ | $m_7$ | 19 | | $b_2[13]$ | $b_{2,13}=0$, $c_{2,14}=0$, $d_{2,19}=a_{2,19}$, $d_{2,22}=a_{2,22}$ |
| 9 | $a_3$ | $m_8$ | 3 | | $a_3[17]$ | $a_{3,17}=0$, $c_{2,13}=d_{2,13}$, $b_{2,14}=0$, $b_{2,19}=0$, $b_{2,22}=0$ |
| 10 | $d_3$ | $m_9$ | 7 | | $d_3[20,-21,-22,23]$ | $d_{3,20}=0$, $d_{3,21}=1$, $d_{3,22}=1$, $d_{3,23}=0$, $b_{2,17}=c_{2,17}$, $a_{3,13}=1$, $a_{3,19}=1$, $a_{3,22}=1$ |
| 11 | $c_3$ | $m_{10}$ | 11 | | $c_3[-30]$ | $c_{3,30}=1$, $a_{3,20}=b_{2,20}$, $a_{3,21}=b_{2,21}$, $a_{3,23}=b_{2,23}$, $d_{3,17}=0$, $d_{3,13}=1$ |
| 12 | $b_3$ | $m_{11}$ | 19 | | $b_3[32]$ | $b_{3,32}=0$, $d_{3,30}=a_{3,30}$, $c_{3,20}=0$, $c_{3,21}=0$, $c_{3,22}=0$, $c_{3,23}=0$, $c_{3,17}=1$ |
| 13 | $a_4$ | $m_{12}$ | 3 | $-2^{16}$ | $a_4[23,26]$ | $a_{4,23}=0$, $a_{4,26}=0$, $b_{3,20}=0$, $b_{3,21}=1$, $b_{3,22}=1$, $b_{3,23}=0$, $c_{3,32}=d_{3,32}$, $b_{3,30}=0$ |
| 14 | $d_4$ | $m_{13}$ | 7 | | $d_4[-27,-29,30]$ | $d_{4,27}=1$, $d_{4,29}=1$, $d_{4,30}=0$, $b_{3,26}=c_{3,26}$, $a_{4,32}=0$, $a_{4,30}=1$ |
| 15 | $c_4$ | $m_{14}$ | 11 | | | $a_{4,27}=b_{3,27}$, $a_{4,29}=b_{3,29}$, $d_{4,23}=0$, $d_{4,26}=0$, $d_{4,32}=1$ |
| 16 | $b_4$ | $m_{15}$ | 19 | | $b_4[17,19]$ | $b_{4,17}=0$, $b_{4,19}=0$, $c_{4,27}=0$, $c_{4,29}=0$, $c_{4,30}=1$, $c_{4,23}=1$, $c_{4,26}=1$ |
| 17 | $a_5$ | $m_0$ | 3 | | $a_5[-26,27,-29,-32]$ | $a_{5,26}=1$, $a_{5,27}=0$, $a_{5,29}=1$, $a_{5,32}=1$, $b_{4,27}=1$, $b_{4,29}=1$, $b_{4,30}=1$, $c_{4,19}=d_{4,19}$, $c_{4,17}=d_{4,17}$ |
| 18 | $d_5$ | $m_4$ | 5 | | $d_5$ | $b_{4,26}=c_{4,26}$, $b_{4,32}=c_{4,32}$, $a_{5,19}=c_{4,19}$, $a_{5,17}=c_{4,17}$ |
| 19 | $c_5$ | $m_8$ | 9 | | $c_5$ | $d_{5,19}=a_{5,19}$, $d_{5,26}=b_{4,26}$, $d_{5,27}=b_{4,27}$, $d_{5,29}=b_{4,29}$, $d_{5,32}=b_{4,32}$, $d_{5,17}=a_{5,17}$ |
| 20 | $b_5$ | $m_{12}$ | 13 | $-2^{16}$ | $b_5[32]$ | $b_{5,32}=0$, $c_{5,26}=d_{5,26}$, $c_{5,27}=d_{5,27}$, $c_{5,29}=d_{5,29}$, $c_{5,32}=d_{5,32}$ |
| 21 | $a_6$ | $m_1$ | 3 | | $a_6[29,-32]$ | $a_{6,29}=0$, $a_{6,32}=1$ |
| 22 | $d_6$ | $m_5$ | 5 | | | $b_{5,29}=c_{5,29}$ |
| 23 | $c_6$ | $m_9$ | 9 | | | $d_{6,29}=b_{5,29}$ |
| 24 | $b_6$ | $m_{13}$ | 13 | | $b_6$ | $c_{6,29}=d_{6,29}$, $c_{6,32}=d_{6,32}+1$ |
| 25 | $a_7$ | $m_2$ | 3 | $-2^{28}+2^{31}$ | $a_7$ | |
| 36 | $b_9$ | $m_{12}$ | 15 | $-2^{16}$ | $b_9[-32]$ | $b_{9,32}=1$ |
| 37 | $a_{10}$ | $m_2$ | 3 | $-2^{28}+2^{31}$ | $a_{10}[-32]$ | $a_{10,32}=1$ |
| 38 | $d_{10}$ | $m_{10}$ | 9 | | | |
| 39 | $c_{10}$ | $m_6$ | 11 | | | |
| 40 | $b_{10}$ | $m_{14}$ | 15 | | | |
| 41 | $a_{11}$ | $m_1$ | 3 | $2^{31}$ | | |

**Table 7.** A set of sufficient conditions for the 4-multicollision of MD4.

| Step | Output variable | Variable conditions |
|---|---|---|
| 1 | $a_1$ | $a_{1,7} = b_{0,7}$ |
| 2 | $d_1$ | $d_{1,7} = 0,\ d_{1,8} = a_{1,8},\ d_{1,11} = a_{1,11}$ |
| 3 | $c_1$ | $c_{1,7} = 1,\ c_{1,8} = 1,\ c_{1,11} = 0,\ c_{1,26} = d_{1,26}$ |
| 4 | $b_1$ | $b_{1,7} = 1,\ b_{1,8} = 0,\ b_{1,11} = 0,\ b_{1,26} = 0,\ b_{1,29} = c_{1,29}$ |
| 5 | $a_2$ | $a_{2,8} = 1,\ a_{2,11} = 1,\ a_{2,14} = b_{1,14},\ a_{2,26} = 0,\ a_{2,29} = 0$ |
| 6 | $d_2$ | $d_{2,14} = 0,\ d_{2,19} = a_{2,19},\ d_{2,22} = a_{2,22}d_{2,26} = 1,\ d_{2,29} = 0$ |
| 7 | $c_2$ | $c_{2,13} = d_{2,13},\ c_{2,14} = 0,\ c_{2,19} = 1,\ c_{2,22} = 0,\ c_{2,29} = 1,\ c_{2,32} = 0$ |
| 8 | $b_2$ | $b_{2,13} = 0,\ b_{2,14} = 0,\ b_{2,17} = c_{2,17},\ b_{2,19} = 0,\ b_{2,22} = 0,\ b_{2,32} = 0$ |
| 9 | $a_3$ | $a_{3,13} = 1,\ a_{3,17} = 0,\ a_{3,19} = 1,\ a_{3,20} = b_{2,20},\ a_{3,21} = b_{2,21},a_{3,22} = 1,$ $a_{3,23} = b_{2,23},\ a_{3,32} = 0$ |
| 10 | $d_3$ | $d_{3,11} = a_{3,11},\ d_{3,13} = 1,\ d_{3,17} = 0,\ d_{3,20} = 0,\ d_{3,21} = 1,\ d_{3,22} = 1,$ $d_{3,23} = 0,\ d_{3,30} = a_{3,30},\ d_{3,32} = 1$ |
| 11 | $c_3$ | $c_{3,11} = 1,\ c_{3,17} = 1,\ c_{3,20} = 0,\ c_{3,21} = 0,\ c_{3,22} = 0,c_{3,23} = 0,$ $c_{3,30} = 1,\ c_{3,32} = 1$ |
| 12 | $b_3$ | $b_{3,3} = c_{3,3},\ b_{3,11} = 0,\ b_{3,20} = 0,\ b_{3,21} = 1,\ b_{3,22} = 1,\ b_{3,23} = 0,$ $b_{3,26} = c_{3,26},\ b_{3,30} = 0,\ b_{3,32} = 0$ |
| 13 | $a_4$ | $a_{4,3} = 0,\ a_{4,11} = 1,\ a_{4,23} = 0,\ a_{4,26} = 0,\ a_{4,27} = b_{3,27},\ a_{4,29} = b_{3,29},$ $a_{4,30} = 1,\ a_{4,32} = 0$ |
| 14 | $d_4$ | $d_{4,3} = 0,\ d_{4,22} = a_{4,22},\ d_{4,23} = 0,\ d_{4,26} = 0,\ d_{4,27} = 1,\ d_{4,29} = 1,$ $d_{4,30} = 0,\ d_{4,32} = 1$ |
| 15 | $c_4$ | $c_{4,3} = 1,\ c_{4,17} = d_{4,17},\ c_{4,19} = d_{4,19},\ c_{4,22} = 1,\ c_{4,23} = 1,\ c_{4,26} = 1,$ $c_{4,27} = 0,\ c_{4,29} = 0,\ c_{4,30} = 1$ |
| 16 | $b_4$ | $b_{4,6} = c_{4,6} + 1,\ b_{4,17} = 0,\ b_{4,19} = 0,\ b_{4,22} = d_{4,22},\ b_{4,26} = 1,\ b_{4,27} = 1,$ $b_{4,29} = 1,\ b_{4,30} = 1,\ b_{4,32} = c_{4,32}$ |
| 17 | $a_5$ | $a_{5,6} = 0,\ a_{5,11} = b_{4,11},\ a_{5,17} = c_{4,17},\ a_{5,19} = c_{4,19}\ ,\ a_{5,22} = b_{4,22},$ $a_{5,26} = 1,\ a_{5,27} = 0,a_{5,29} = 1,\ a_{5,31} = b_{4,31},\ a_{5,32} = 1$ |
| 18 | $d_5$ | $d_{5,6} = b_{4,6},\ d_{5,11} = 0,\ d_{5,17} = a_{5,17},\ d_{5,19} = a_{5,19},\ d_{5,26} = b_{4,26},$ $d_{5,27} = b_{4,27},\ d_{5,29} = b_{4,29},\ d_{5,31} = 0,\ d_{5,32} = b_{4,32}$ |
| 19 | $c_5$ | $c_{5,6} = d_{5,6},\ c_{5,11} = a_{5,11},\ c_{5,26} = d_{5,26},\ c_{5,27} = d_{5,27},\ c_{5,29} = d_{5,29},$ $c_{5,31} = 1,\ c_{5,32} = d_{5,32}$ |
| 20 | $b_5$ | $b_{5,9} = c_{5,9},\ b_{5,11} = c_{5,11},\ b_{5,29} = c_{5,29},\ b_{5,32} = 0$ |
| 21 | $a_6$ | $a_{6,9} = 0,\ a_{6,16} = b_{5,16},\ a_{6,29} = 0,\ a_{6,31} = b_{5,31} + 1,\ a_{6,32} = 1$ |
| 22 | $d_6$ | $d_{6,8} = a_{6,8},\ d_{6,9} = b_{5,9},\ d_{6,16} = 0,\ d_{6,29} = b_{5,29}$ |
| 23 | $c_6$ | $c_{6,8} = 0,\ c_{6,9} = 1,\ c_{6,16} = a_{6,16},\ c_{6,29} = d_{6,29},\ c_{6,32} = d_{6,32} + 1$ |
| 24 | $b_6$ | $b_{6,8} = d_{6,8},\ b_{6,9} = d_{6,9} + 1,\ b_{6,16} = c_{6,16}$ |
| 25 | $a_7$ | $a_{7,8} = b_{6,8},\ a_{7,9} = b_{6,9},\ a_{7,21} = b_{6,21}$ |
| 26 | $d_7$ | $d_{7,17} = a_{7,17},\ d_{7,21} = 0$ |
| 27 | $c_7$ | $c_{7,17} = 1,\ c_{7,21} = a_{7,21}$ |
| 28 | $b_7$ | $b_{7,17} = d_{7,17},\ b_{7,21} = c_{7,21}$ |
| 29 | $a_8$ | $a_{8,17} = b_{7,17},\ a_{8,26} = b_{7,26}$ |
| 30 | $d_8$ | $d_{8,26} = 0$ |
| 31 | $c_8$ | $c_{8,26} = 1$ |
| 32 | $b_8$ | |
| 33 | $a_9$ | $a_{9,26} = b_{8,26}$ |
| 34 | $d_9$ | |
| 35 | $c_9$ | |
| 36 | $b_9$ | $b_{9,32} = 1$ |
| 37 | $a_{10}$ | $a_{10,32} = 1$ |

**Table 8.** The near-collision differential path for the 3-Pass HAVAL.

| Step | $m'_{i-1}$ | Outputs for $M'_0$ | Sufficient conditions |
|---|---|---|---|
| 6 | $m'_5$ | $a_6[-12,13]$, $a_5$, $a_4$, $a_3$, $a_2$, $a_1$, $a_0$, $b_0$ | $a_{6,12}=1$, $a_{6,13}=0$ |
| 7 | $m_6$ | $a_7$, $a_6[-12,13]$, $a_5$, $a_4$, $a_3$, $a_2$, $a_1$, $a_0$ | $a_{0,12}=0$, $a_{0,13}=0$ |
| 8 | $m_7$ | $a_8$, $a_7$, $a_6[-12,13]$, $a_5$, $a_4$, $a_3$, $a_2$, $a_1$ | $a_{2,12}=0$, $a_{2,13}=0$ |
| 9 | $m_8$ | $a_9$, $a_8$, $a_7$, $a_6[-12,13]$, $a_5$, $a_4$, $a_3$, $a_2$ | $a_{5,12}=a_{4,12}$, $a_{5,13}=a_{4,13}$ |
| 10 | $m_9$ | $a_{10}$, $a_9$, $a_8$, $a_7$, $a_6[-12,13]$, $a_5$, $a_4$, $a_3$ | $a_{7,12}=0$, $a_{7,13}=0$ |
| 11 | $m_{10}$ | $a_{11}[-6,-7,-8,9]$, $a_{10}$, $a_9$, $a_8$, $a_7$, $a_6[-12,13]$, $a_5$, $a_4$ | $a_{8,12}=1$, $a_{8,13}=0$, $a_{5,13}=0$, $a_{11,6}=1$, $a_{11,7}=1$, $a_{11,8}=1$, $a_{11,9}=0$ |
| 12 | $m_{11}$ | $a_{12}$, $a_{11}[-6,-7,-8,9]$, $a_{10}$, $a_9$, $a_8$, $a_7$, $a_6[-12,13]$, $a_5$ | $a_{10,12}=0$, $a_{10,13}=0$, $a_{6,6}=0$, $a_{6,7}=0$, $a_{6,8}=0$, $a_{6,9}=0$ |
| 13 | $m_{12}$ | $a_{13}$, $a_{12}$, $a_{11}[-6,-7,-8,9]$, $a_{10}$, $a_9$, $a_8$, $a_7$, $a_6[-12,13]$ | $a_{12,12}=0$, $a_{12,13}=0$, $a_{7,6}=0$, $a_{7,7}=0$, $a_{7,8}=0$, $a_{7,9}=0$ |
| 14 | $m_{13}$ | $a_{14}$, $a_{13}$, $a_{12}$, $a_{11}[-6,-7,-8,9]$, $a_{10}$, $a_9$, $a_8$, $a_7$ | $a_{10,6}=a_{9,6}$, $a_{10,7}=a_{9,7}$, $a_{10,8}=a_{9,8}+1$, $a_{10,9}=a_{9,9}$, $a_{9,8}=0$ |
| 15 | $m_{14}$ | $a_{15}$, $a_{14}$, $a_{13}$, $a_{12}$, $a_{11}[-6,-7,-8,9]$, $a_{10}$, $a_9$, $a_8$ | $a_{12,6}=0$, $a_{12,7}=0$,$a_{12,8}=0$, $a_{12,9}=0$ |
| 16 | $m_{15}$ | $a_{16}$, $a_{15}$, $a_{14}$, $a_{13}$, $a_{12}$, $a_{11}[-6,-7,-8,9]$, $a_{10}$, $a_9$ | $a_{13,6}=1$, $a_{13,7}=1$,$a_{13,8}=1$, $a_{13,9}=1$ |
| 17 | $m_{16}$ | $a_{17}[-2]$, $a_{16}$, $a_{15}$, $a_{14}$, $a_{13}$, $a_{12}$, $a_{11}[-6,-7,-8,9]$, $a_{10}$ | $a_{15,6}=0$, $a_{15,7}=0$, $a_{15,8}=0$, $a_{15,9}=1$, $a_{10,9}=0$, $a_{14,9}=1$, $a_{17,2}=1$ |
| 18 | $m_{17}$ | $a_{18}$, $a_{17}[-2]$, $a_{16}$, $a_{15}$, $a_{14}$, $a_{13}$, $a_{12}$, $a_{11}[-6,-7,-8,9]$ | $a_{17,6}=0$, $a_{17,7}=0$, $a_{17,8}=0$, $a_{17,9}=0$, $a_{11,2}=0$ |
| 19 | $m_{18}$ | $a_{19}$, $a_{18}$, $a_{17}[-2]$, $a_{16}$, $a_{15}$, $a_{14}$, $a_{13}$, $a_{12}$ | $a_{13,2}=1$, $a_{14,2}=0$, $a_{15,2}=0$ |
| 20 | $m_{19}$ | $a_{20}$, $a_{19}$, $a_{18}$, $a_{17}[-2]$, $a_{16}$, $a_{15}$, $a_{14}$, $a_{13}$ | $a_{16,2}=a_{15,2}$ |
| 21 | $m_{20}$ | $a_{21}$, $a_{20}$, $a_{19}$, $a_{18}$, $a_{17}[-2]$, $a_{16}$, $a_{15}$, $a_{14}$ | $a_{18,2}=0$ |
| 22 | $m_{21}$ | $a_{22}$, $a_{21}$, $a_{20}$, $a_{19}$, $a_{18}$, $a_{17}[-2]$, $a_{16}[$, $a_{15}$ | $a_{19,2}=1$ |
| 23 | $m_{22}$ | $a_{23}$, $a_{22}$, $a_{21}$, $a_{20}$, $a_{19}$, $a_{18}$, $a_{17}[-2]$, $a_{16}$ | $a_{21,2}=0$ |
| 24 | $m_{23}$ | $a_{24}$, $a_{23}$, $a_{22}$, $a_{21}$, $a_{20}$, $a_{19}$, $a_{18}$, $a_{17}[-2]$ | $a_{23,2}=0$ |
| 25 | $m_{24}$ | $a_{25}[-23]$, $a_{24}$, $a_{23}$, $a_{22}$, $a_{21}$, $a_{20}$, $a_{19}$, $a_{18}$ | $a_{25,23}=1$ |
| 26 | $m_{25}$ | $a_{26}$, $a_{25}[-23]$, $a_{24}$, $a_{23}$, $a_{22}$, $a_{21}$, $a_{20}$, $a_{19}$ | $a_{19,23}=0$ |
| 27 | $m_{26}$ | $a_{27}$, $a_{26}$, $a_{25}[-23]$, $a_{24}$, $a_{23}$, $a_{22}$, $a_{21}$, $a_{20}$ | $a_{21,23}=0$ |
| 28 | $m_{27}$ | $a_{28}$, $a_{27}$, $a_{26}$, $a_{25}[-23]$, $a_{24}$, $a_{23}$, $a_{22}$, $a_{21}$ | $a_{24,23}=a_{23,23}$ |
| 29 | $m_{28}$ | $a_{29}$, $a_{28}$, $a_{27}$, $a_{26}$, $a_{25}[-23]$, $a_{24}$, $a_{23}$, $a_{22}$ | $a_{26,23}=0$ |
| 30 | $m_{29}$ | $a_{30}$, $a_{29}$, $a_{28}$, $a_{27}$, $a_{26}$, $a_{25}[-23]$, $a_{24}$, $a_{23}$ | $a_{27,23}=1$ |
| 31 | $m_{30}$ | $a_{31}$, $a_{30}$, $a_{29}$, $a_{28}$, $a_{27}$, $a_{26}$, $a_{25}[-23]$, $a_{24}$ | $a_{29,23}=0$ |
| 32 | $m_{31}$ | $a_{32}$, $a_{31}$, $a_{30}$, $a_{29}$, $a_{28}$, $a_{27}$, $a_{26}$, $a_{25}[-23]$ | $a_{31,23}=0$ |
| 33 | $m'_5$ | $a_{33}$, $a_{32}$, $a_{31}$, $a_{30}$, $a_{29}$, $a_{28}$, $a_{27}$, $a_{26}$ | |
| ... | ... | ... | ... |
| 95 | $m'_5$ | $a_{95}[12]$, $a_{94}$, $a_{93}$, $a_{92}$, $a_{91}$, $a_{90}$, $a_{89}$, $a_{88}$ | $a_{95,12}=0$ |
| 96 | $m_2$ | $a_{96}$, $a_{95}[12]$, $a_{94}$, $a_{93}$, $a_{92}$, $a_{91}$, $a_{90}$, $a_{89}$ | $a_{92,12}=1$ |

**Table 9.** A set of sufficient conditions for the 8 near-collision of 3-Pass HAVAL.

| Step | Output variable | Varialbe conditions |
|---|---|---|
| 0 | $IV$ | $a_{0,12}=0,\ a_{0,13}=0,\ a_{0,20}=0,\ a_{0,21}=0,\ a_{0,28}=0,\ a_{0,29}=0$ |
| 1 | $a_1$ | |
| 2 | $a_2$ | $a_{2,12}=0,\ a_{2,13}=0,\ a_{2,20}=0,\ a_{2,21}=0,\ a_{2,28}=0,\ a_{2,29}=0$ |
| 3 | $a_3$ | |
| 4 | $a_4$ | $a_{4,13}=0,\ a_{4,21}=0,\ a_{4,29}=0$ |
| 5 | $a_5$ | $a_{5,6}=0,\ a_{5,7}=0,\ a_{5,8}=0,\ a_{5,9}=0,\ a_{5,12}=a_{4,12},\ a_{5,13}=a_{4,13},$ $a_{5,14}=0,\ a_{5,15}=0,\ a_{5,16}=0,\ a_{5,17}=0,\ a_{5,20}=a_{4,20},\ a_{5,21}=a_{4,21},$ $a_{5,22}=0,\ a_{5,23}=0,\ a_{5,24}=0,\ a_{5,25}=0,\ a_{5,28}=a_{4,28},\ a_{5,29}=a_{4,29}$ |
| 6 | $a_6$ | $a_{6,12}=1,\ a_{6,13}=0,\ a_{6,20}=1,\ a_{6,21}=0,\ a_{6,28}=1,\ a_{6,29}=0$ |
| 7 | $a_7$ | $a_{7,6}=0,\ a_{7,7}=0,\ a_{7,8}=0,\ a_{7,9}=0,\ a_{7,12}=0,\ a_{7,13}=0,\ a_{7,14}=0,$ $a_{7,15}=0,\ a_{7,16}=0,\ a_{7,17}=0,\ a_{7,20}=0,\ a_{7,21}=0,\ a_{7,22}=0,\ a_{7,23}=0,$ $a_{7,24}=0,\ a_{7,25}=0,\ a_{7,28}=0,\ a_{7,29}=0$ |
| 8 | $a_8$ | $a_{8,12}=1,\ a_{8,13}=0,\ a_{8,20}=1,\ a_{8,21}=0,\ a_{8,28}=1,\ a_{8,29}=0$ |
| 9 | $a_9$ | $a_{9,8}=0,\ a_{9,9}=0,\ a_{9,16}=0,\ a_{9,17}=0,\ a_{9,24}=0,\ a_{9,25}=0$ |
| 10 | $a_{10}$ | $a_{10,6}=a_{9,6},\ a_{10,7}=a_{9,7},\ a_{10,8}=a_{9,8}+1,\ a_{10,9}=a_{9,9},\ a_{10,12}=0,$ $a_{10,13}=0,\ a_{10,14}=a_{9,14},\ a_{10,15}=a_{9,15},\ a_{10,16}=a_{9,16}+1,\ a_{10,17}=a_{9,17},$ $a_{10,20}=0,\ a_{10,21}=0,\ a_{10,22}=a_{9,22},\ a_{10,23}=a_{9,23},\ a_{10,24}=a_{9,24}+1,$ $a_{10,25}=a_{9,25},\ a_{10,28}=0,\ a_{10,29}=0$ |
| 11 | $a_{11}$ | $a_{11,2}=0,\ a_{11,6}=1,\ a_{11,7}=1,\ a_{11,8}=1,\ a_{11,9}=0,\ a_{11,10}=0,\ a_{11,14}=1,$ $a_{11,15}=1,\ a_{11,16}=1,\ a_{11,17}=0,\ a_{11,18}=0,\ a_{11,22}=1,\ a_{11,23}=1,$ $a_{11,24}=1,\ a_{11,25}=0$ |
| 12 | $a_{12}$ | $a_{12,6}=0,\ a_{12,7}=0,\ a_{12,8}=0,\ a_{12,9}=0,\ a_{12,12}=0,\ a_{12,13}=0,\ a_{12,14}=0,$ $a_{12,15}=0,\ a_{12,16}=0,\ a_{12,17}=0,\ a_{12,20}=0,\ a_{12,21}=0,\ a_{12,22}=0,\ a_{12,23}=0,$ $a_{12,24}=0,\ a_{12,25}=0,\ a_{12,28}=0,\ a_{12,29}=0$ |
| 13 | $a_{13}$ | $a_{13,2}=1,\ a_{13,6}=1,\ a_{13,7}=1,\ a_{13,8}=1,\ a_{13,9}=1,\ a_{13,10}=1,\ a_{13,14}=1,$ $a_{13,15}=1,\ a_{13,16}=1,\ a_{13,17}=1,\ a_{13,18}=1,\ a_{13,22}=1,\ a_{13,23}=1,\ a_{13,24}=1,$ $a_{13,25}=1$ |
| 14 | $a_{14}$ | $a_{14,2}=0,\ a_{14,9}=1,\ a_{14,10}=0,\ a_{14,17}=1,\ a_{14,18}=0,\ a_{14,25}=1$ |
| 15 | $a_{15}$ | $a_{15,2}=0,\ a_{15,6}=0,\ a_{15,7}=0,\ a_{15,8}=0,\ a_{15,9}=1,\ a_{15,10}=0,\ a_{15,14}=0,$ $a_{15,15}=0,\ a_{15,16}=0,\ a_{15,17}=1,\ a_{15,18}=0,\ a_{15,22}=0,\ a_{15,23}=0,\ a_{15,24}=0,$ $a_{15,25}=1$ |
| 16 | $a_{16}$ | $a_{16,2}=a_{15,2},\ a_{16,10}=a_{15,10},\ a_{16,18}=a_{15,18}$ |
| 17 | $a_{17}$ | $a_{17,2}=1,\ a_{17,6}=0,\ a_{17,7}=0,\ a_{17,8}=0,\ a_{17,9}=0,\ a_{17,10}=1,\ a_{17,14}=0,$ $a_{17,15}=0,\ a_{17,16}=0,\ a_{17,17}=0,\ a_{17,18}=1,\ a_{17,22}=0,\ a_{17,23}=0,\ a_{17,24}=0,$ $a_{17,25}=0$ |
| 18 | $a_{18}$ | $a_{18,2}=0,\ a_{18,10}=0,\ a_{18,18}=0$ |
| 19 | $a_{19}$ | $a_{19,2}=1,\ a_{19,7}=0,\ a_{19,10}=1,\ a_{19,18}=1,\ a_{19,23}=0,\ a_{19,31}=0$ |
| 20 | $a_{20}$ | |
| 21 | $a_{21}$ | $a_{21,2}=0,\ a_{21,7}=0,\ a_{21,10}=0,\ a_{21,18}=0,\ a_{21,23}=0,\ a_{21,31}=0$ |
| 22 | $a_{22}$ | |
| 23 | $a_{23}$ | $a_{23,2}=0,\ a_{23,10}=0,\ a_{23,18}=0$ |
| 24 | $a_{24}$ | $a_{24,7}=a_{23,7},\ a_{24,23}=a_{23,23},\ a_{24,31}=a_{23,31}$ |
| 25 | $a_{25}$ | $a_{25,7}=1,\ a_{25,23}=1,\ a_{25,31}=1$ |
| 26 | $a_{26}$ | $a_{26,7}=0,\ a_{26,23}=0,\ a_{26,31}=0$ |
| 27 | $a_{27}$ | $a_{27,7}=1,\ a_{27,23}=1,\ a_{27,31}=1$ |
| 28 | $a_{28}$ | |
| 29 | $a_{29}$ | $a_{29,7}=0,\ a_{29,23}=0,\ a_{29,31}=0$ |
| 30 | $a_{30}$ | |
| 31 | $a_{31}$ | $a_{31,7}=0,\ a_{31,23}=0,\ a_{31,31}=0$ |
| 92 | $a_{92}$ | $a_{92,12}=1,\ a_{92,20}=1,\ a_{92,28}=1$ |
| 95 | $a_{95}$ | $a_{95,12}=0,\ a_{95,20}=0,\ a_{95,28}=0$ |
| | | $(b_0+a_{95})_{12}=0,\ (b_0+a_{95})_{20}=0,\ (b_0+a_{95})_{28}=0$ |