社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2006－CSEC－34（19）
2006／7／20

MD5の衝突条件の検証

仲野　有登†　桑門　秀典††　森井　昌克††

† 神戸大学大学院自然科学研究科　〒657-8501 兵庫県神戸市灘区六甲台町 1-1
†† 神戸大学工学部電気電子工学科　〒657-8501 兵庫県神戸市灘区六甲台町 1-1
E-mail: †065t236n@stu.kobe-u.ac.jp, ††{kuwakado,mmorii}@kobe-u.ac.jp

あらまし　本論文ではMD5の衝突探索で用いられる十分条件を実験的に検証した．具体的には，衝突メッセージの作成を行い，得られたメッセージが十分条件を満たすかどうかを検証した．その結果，従来示されている十分条件のなかに不要な条件が七つあることを発見し，これらの条件が不要であることを理論的に証明した．そのうちの三つ条件については満たさない割合を理論的に導出する．

キーワード　ハッシュ関数，MD5，十分条件，衝突探索

Inspection of sufficient conditions of MD5

Yuto NAKANO†, Hidenori KUWAKADO††, and Masakatu MORII††

† Graduate School of Science and Technology, Kobe University
Rokkodai-cho 1-1, Nada-ku, Kobe-shi, 657-8501 Japan
†† Fuctory of Engneering, Kobe University
Rokkodai-cho 1-1, Nada-ku, Kobe-shi, 657-8501 Japan
E-mail: †065t236n@stu.kobe-u.ac.jp, ††{kuwakado,mmorii}@kobe-u.ac.jp

Abstract　Sufficient conditions for finding the collision pairs of MD5 are about 600 conditions on internal variables, but the necessity of the conditions has not been studied. We investigate their necessity by a computer simulation, that is, check 1724 pairs of collision messages generated with the collision finding algorithm. As a result, we found that seven conditions are unnecessary. We also show the reason that three conditions of them are unnecessary. The theoretical analysis on the remaining four conditions is a future work.

Key words　hash function, MD5, sufficient condition, collision search

1. まえがき

電子メールの送受信やオンライン取り引きにおいて，メッセージが正当な送信者から発信され，途中で改ざんされていないことを示すためにディジタル署名が用いられている．ハッシュ関数はディジタル署名の偽造を防止するための中核をなしている．また，ネットワークを通じて受信したデータが通信途中で改ざんされていないかを検査する改ざん検知にも用いられる．

一般にハッシュ関数には次に示すような四つの性質が求められる．第一に，現実的な計算時間でハッシュ値を求めることができることが必要である．ディジタル署名にハッシュ関数を用いる場合，署名を行う度にハッシュ値を求める必要があるので計算が高速にできるという性質が求められる．第二に，任意の長さの入力に対して固定長の疑似乱数を出力する性質が必要である．第三に，同じハッシュ値をもつ異なる入力メッセージを求めることは困難であることが挙げられる．ディジタル署名では文章のハッシュ値に対して署名処理を行うため，衝突を起こす二つのメッセージが容易に求まればディジタル署名の偽造が可能となる．また，この性質はデータの改ざん検知にハッシュ関数を用いる際に重要となる．第四に，ハッシュ値から入力を求めるのは困難であることが求められる．

本論文で扱うMD5は1992年にRivestによって提案されたハッシュ関数である[3]．MD5では128ビットの初期値と512ビットの入力から128ビットのハッシュ値を出力する．MD5に対する攻撃としてWangとYu[4]により提案された差分攻撃がある．この攻撃では2^{32}を法とした算術差分を用いて，二つの入力メッセージが衝突をおこすように各ステップの差分値を決める．圧縮関数内の変数がこの差分値になる十分条件をWangとYuは示した．

しかし，矢嶋と下山[5]はWangとYuの十分条件には誤りと不足があることを示し，追加すべき四つの条件と修正すべき一つの条件を示した．本論文では十分条件についてさらに検証

を行う．具体的にはまず，矢嶋と下山の十分条件を用いて衝突メッセージを作成する．そして，得られたメッセージが十分条件を満たすかどうかを一つ一つ検証し，条件を満たす割合を計算した．その結果，すべての衝突メッセージが満たす条件と不要な条件があることがわかり，不要な六つの条件と修正すべき一つの条件を明らかにした．

最後に本論文の構成を示す．2章でMD5のアルゴリズムについて説明し，3章でWangとYuによるMD5の攻撃アルゴリズムと矢嶋と下山によって再構築された攻撃アルゴリズムを示し，修正された十分条件を示す．4章で十分条件についての検証を行い，5章で4章にて得られた条件を用いる各ステップについて計算することで十分条件の考察を行い，6章に結論を示す．

2. MD5のアルゴリズム

圧縮関数は128ビットの初期値と512ビットのメッセージから128ビットの出力を生成する．圧縮関数の出力は次の圧縮関数の初期値として用いられる．最初に使用する初期値は次のように与えられる(0xは16進数であることを表す)．

A = 0x67452301 B = 0xefcdab89

C = 0x98badcfe D = 0x10325476

MD5の圧縮関数は四つのラウンドからなる．それぞれのラウンドで行われる演算は16のステップから構成される．内部変数としてa,b,c,dを定義する．まず次のように各内部変数に初期値を代入する．

$$a = A,\ b = B,\ c = C,\ d = D$$

各ステップでa,b,c,dのうちの一つが更新され，その結果を次のステップに用いてさらに計算を進める．これにより雪崩効果がより速くなる．内部変数a,b,c,dのそれぞれについての演算は次のように定義されている．この演算で等式の左辺が更新された内部変数の値であり，右辺は更新される前の値である．図1にこの様子を示す．

$$a = b + ((a + f(b,c,d) + m + const) \lll s)$$
$$d = a + ((d + f(a,b,c) + m + const) \lll s)$$
$$c = d + ((c + f(d,a,b) + m + const) \lll s)$$
$$b = c + ((b + f(c,d,a) + m + const) \lll s)$$

ここで，$+$は2^{32}を法とした算術加算，mは入力メッセージを32ビットごとにわけたサブブロックである．以後，特に断らない限り2^{32}を法とした算術加算，減算を単に加算，減算という．すべてのメッセージのサブブロックが各ラウンドで1度ずつ使われ，使用される順序はラウンドによって異なる．$const, s$は予め与えられたステップによって異なる定数で，$\lll s$はsビットの左回転シフトを表す．またfはラウンドによって異なる非線形関数であり，次のように定義される．

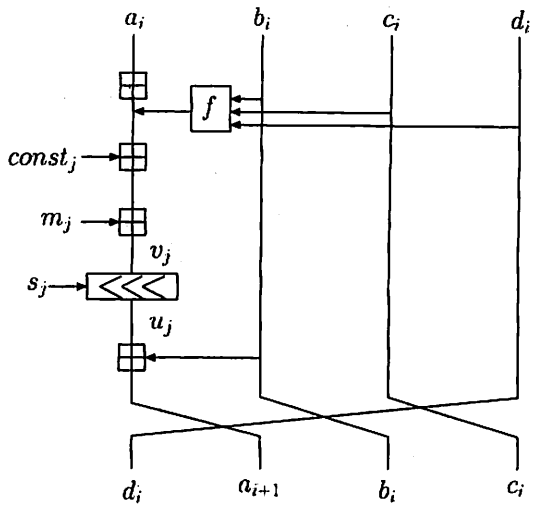

図1 MD5の1ステップの演算

$$\text{round 1}: f = F(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$
$$\text{round 2}: f = G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$$
$$\text{round 3}: f = H(X,Y,Z) = X \oplus Y \oplus Z$$
$$\text{round 4}: f = I(X,Y,Z) = Y \oplus (X \vee (\neg Z))$$

ただし$\oplus$はXOR，$\wedge$はAND，$\vee$はOR，$\neg$はNOTを表す．

本論文で用いる記号を次のように定義する．

a_i, b_i, c_i, d_i: それぞれi番目のa,b,c,dの値 ($1 \leqq i \leqq 16$)

$a_{i,j}$, $b_{i,j}$, $c_{i,j}$, $d_{i,j}$: a_i, b_i, c_i, d_i の右からjビット目の値 ($1 \leqq j \leqq 32$)

aa_0, bb_0, cc_0, dd_0: ブロック1の圧縮関数の出力

aa_1, bb_1, cc_1, dd_1: ブロック2の圧縮関数の出力

$aa_{i,j}$, $bb_{i,j}$, $cc_{i,j}$, $dd_{i,j}$: 圧縮関数の出力のjビット目 ($i = 0,1$, $1 \leqq j \leqq 32$)

ϕ_k: ステップkの関数fの出力 ($0 \leqq k \leqq 63$)

$\phi_{k,j}$: ϕ_kの右からj番目の値 ($0 \leqq k \leqq 63$, $1 \leqq j \leqq 32$)

v_k: 圧縮関数の1ステップの処理における左回転シフトを行う前の値 ($0 \leqq k \leqq 63$)

u_k: 圧縮関数の1ステップの処理における左回転シフトを行った後の値 ($0 \leqq k \leqq 63$)

m_ℓ: 入力メッセージを32ビットごとに区切ったサブブロックのℓ番目 ($0 \leqq \ell \leqq 15$)

$x_i[j]$: xの右からj番目のビットが0から1に変化 (xはa,b,c,d,ϕのいずれか)

$x_i[-j]$: xの右からj番目のビットが1から0に変化 (xはa,b,c,d,ϕのいずれか)

$\lll s$: sビットの左回転シフト

$\ggg s$: sビットの右回転シフト

m^{old}: 変更前のメッセージのサブブロック

m^{new}: 変更後のメッセージのサブブロック

3. MD5の衝突探索

1996年にDobbertin[1]はMD5の初期値を自由に選ぶことで512ビットの衝突メッセージを発見した．2004年にWangとYuによってMD5の完全な衝突メッセージが示された．この衝突メッセージの長さは1024ビットであった．

3.1 WangとYuによる衝突探索

WangとYuの攻撃は算術差分を用いた差分攻撃であり，aとa'の差分は

$$\Delta a = (a' - a) \bmod 2^{32}$$

で定義される．

（1） ブロック1の512ビットのメッセージM_0をランダムに決定する．

（2） (1)で決定したメッセージのラウンド1とラウンド2の初めの2ステップに関してmessage modificationを行い，十分条件を満たすようにする．single-message modificationと十分条件については後述する．

（3） $M_0' = M_0 + \Delta M_0$を生成する．ここで，ΔM_0は次のように定められている．

$\Delta M_0 = (0,0,0,0,2^{31},0,0,0,0,0,0,2^{15},0,0,2^{31},0)$

（4） (3)までで得られた二つのメッセージM_0とM_0'について1ブロック分のMD5の処理を行い，出力H_0とH_0'を生成する．

（5） $\Delta H = H_0 - H_0' = (2^{31}, 2^{31}+2^{25}, 2^{31}+2^{25}, 2^{31}+2^{25})$となっていることを確認する．

（6） (5)で求めたΔHが特定の値となっていなければ(1)からやり直す．特定の値となっていれば次のブロックの探索へ進む．

（7） ブロック2の512ビットのメッセージM_1をランダムに決定する．

（8） (7)で決定したメッセージのラウンド1とラウンド2の初めのステップについてmessage modificationを行い，十分条件を満たすようにメッセージを変更する．

（9） $M_1' = M_1 + \Delta M_1$を生成する．ここで，ΔM_1は次のように定められている．

$\Delta M_1 = (0,0,0,0,2^{31},0,0,0,0,0,-2^{15},0,0,2^{31},0)$

（10） メッセージM_1とM_1'について1ブロック分のMD5の処理をし，H_1とH_1'を得る．

（11） $H_1 = H_1'$となることを確認する．なっていなければ(7)からやり直す．

3.2 十分条件

十分条件とは各ステップにおける非線形関数fの差分値$\Delta\phi$を確率1で望みどおりの値とするために内部変数に与える条件である．この条件を満たすようにメッセージを変更すると高い確率で衝突を起こすことができる．以下にブロック1，ステップ5(d_2を求めるステップ)の十分条件の導出を示す．ステップ5の計算は次式で与えられる．

$$d_2 = a_2 + ((d_1 + F(a_2,b_1,c_1) + m_5 + const_5) \lll 12)$$
$$d_2' = a_2' + ((d_1' + F(a_2',b_1',c_1') + m_5' + const_5) \lll 12)$$

ここで，a_2は関数Fの中と外に表れるのでFの中のa_2をa_2^F，Fの外のa_2をa_2^{NF}と書く．また，ステップ5における非線形関数の出力ϕ_5は

$$\phi_5 = F(a_2,b_1,c_1) = (a_2 \wedge b_1) \vee (\neg a_2 \wedge c_1)$$

となる．文献[4]のTable 3より差分入力は次のように与えられる．

$$d_1' = d_1\ ,\ c_1' = c_1\ ,\ b_1' = b_1$$
$$a_2' = a_2[7,8,...,22,-23]\ ,\ d_2' = d_2[-7,24,32]$$

さらに$\Delta m_5 = 0$であるからΔd_2は$\Delta d_2 = \Delta a_2^{NF} + (\Delta\phi_5 \lll 12)$と計算できる．

- Δd_2のゼロでないビットについて

◇ $b_{1,12} = \overline{c_{1,12}} = 1$かつ$d_{2,24} = 0$であれば，次のようにして$\Delta d_{2,24} = 1$を得る．

Table3から$c_{1,12} = c_{1,12}'$，$b_{1,12} = b_{1,12}'$，$a_{2,12}' = 1$，$a_{2,12} = 0$なので

$$\Delta\phi_{5,12} = \phi_{5,12}' - \phi_{5,12} = 1 - 0 = 1$$

これを12ビット左回転シフトさせると$\Delta\phi_{5,12}$は24ビットに移動する．$\Delta a_{2,24}^{NF} = 0$であるから$\Delta d_{2,24} = 1$となる．

◇ $b_{1,20} = \overline{c_{1,20}} = 1$かつ$a_{2,20} = 0$とすると$\Delta\phi_{5,20} = 1$となる．これを12ビット左回転シフトさせると32ビットに移動し，$\Delta a_{2,32}^{NF} = 0$と$d_{2,32} = 0$から$\Delta d_{2,32} = 1$となる．◇ $a_{2,27} = a_{2,27}'$ $b_{1,27} = b_{1,27}'$ $c_{1,27} = c_{1,27}'$であるから$\Delta\phi_{5,27} = 0$となる．これを12ビット左回転シフトさせると$\Delta\phi_{5,27} = 0$は7ビット目に移動する．よって

$$\begin{aligned}\Delta d_2 &= \Delta a_2^{NF}[7,8,\ldots,22,-23] + (\Delta\phi_5 \lll 12)\\ &= 2^6 + 2^7 + \cdots + 2^{21} - 2^{22} + 0\\ &= 2^6 + 2^7 + \cdots + 2^{20} - 2^{21} = \ldots = -2^{-6}\end{aligned}$$

から$\Delta d_{2,7} = -1$

- Δd_2のゼロであるビットについて

◇ $b_{1,21} = c_{1,21}$，$b_{1,22} = c_{1,22}$，$b_{1,23} = c_{1,23}$とすれば

$$\begin{aligned}\Delta\phi_{5,i} &= (a_{2,i}' \wedge b_{1,i}) \vee (\neg a_{2,i}' \wedge c_{1,i})\\ &\quad - (a_{2,i} \wedge b_{1,i}) \vee (\neg a_{2,i} \wedge c_{1,i}) = 0\ \ (i = 21,22,23)\end{aligned}$$

となる．12ビット左回転シフトさせると$\Delta\phi_{5,i}$はそれぞれ1ビット目，2ビット目，3ビット目に移動する．$\Delta a_{2,j}^{NF} = 0$ $(j = 1,2,3)$であるから$\Delta d_{2,j} = 0$ $(j = 1,2,3)$を得る．◇ 24ビット目から26ビット目までは入力差分が0であるから$\Delta\phi_{5,i} = 0$ $(i = 24,25,26)$となり，$\Delta d_{2,j} = 0$ $(j = 4,5,6)$を得る．

◇ 28ビット目から32ビット目までは入力差分が0であるから$\Delta\phi_{5,i} = 0$ $(i = 28,29,\ldots 32)$となる．また1ビット目から11ビット目は$b_{1,j} = c_{1,j}$ $(j = 7,8,\ldots,11)$とすれば

表 1 Multi-message modification の例

step	m_i	s	m_i の変更	内部変数の変更
1	m_1	12	$m_1^{new} = m_1^{old} + 2^{26}$	d_1^{new}, a_1, b_0, c_0
2	m_2	17	$m_2^{new} = ((c_1 - d_1^{new}) \ggg 17)$ $-c_0 - \phi_2(d_1^{new}, a_1, b_0) - const_2$	c_1, d_1^{new}, a_1, b_0
3	m_3	22	$m_3^{new} = ((b_1 - c_1) \ggg 22)$ $-b_0 - \phi_3(c_1, d_1^{new}, a_1) - const_3$	b_1, c_1, d_1^{new}, a_1
4	m_4	7	$m_4^{new} = ((a_2 - b_1) \ggg 7)$ $a_1 - phi_4(b_1, c_1, d_1^{new}) - const_4$	a_2, b_1, c_1, d_1^{new}
5	m_5	12	$m_5^{new} = ((d_2 - a_2) \ggg 12)$ $-d_1^{new} - \phi_5(a_2, b_1, c_1) - const_5$	d_2, a_2, b_1, c_1

$\Delta\phi_{5,i} = 0$ $(i = 1, 2, \ldots, 11)$ となる．以上から $\Delta d_{2,j} = 0$ $(j = 8, 9, \ldots, 23)$ となる．

◇ $b_{1,j} = c_{1,j}$ $(j = 13, 14, \ldots, 19)$ とすれば $\Delta d_{2,i} = 0$ $(i = 25, 26, \ldots, 31)$ となる．

上記と同様にして約 600 個の十分条件をすべて導出できる．

3.3 Message modification

十分条件を満たすようにメッセージを変更する処理を message modification と呼ぶ．message modification には二種類あり，ラウンド 1 の十分条件を満たすようにメッセージを変更するのが single-message modification であり，ラウンド 2 の十分条件を満たすようにメッセージを変更するのが multi-message modification である．

single-message modification ではステップ 2 において十分条件 $c_{1,7} = 0$, $c_{1,12} = 0$, $c_{1,20} = 0$ を満たすようにメッセージに変更を加えることを考える．

まず，ランダムに選んだ c_1^{old} を次のように変更する．

$$c_1^{new} = c_1^{old} - c_{1,7}^{old} \cdot 2^6 - c_{1,12}^{old} \cdot 2^{11} - c_{1,20}^{old} \cdot 2^{19}$$

これにより c_1 は十分条件をみたすように変更されたのでこれを用いて m_2^{new} を導出する．

$$m_2^{new} = ((c_1^{new} - c_1^{old}) \ggg 17) + m_2^{old}$$

multi-message modification ではラウンド 2 の初めの数ステップ (ブロック 1 では 2 ステップ，ブロック 2 では 1 ステップ) に対して十分条件を満たすように以下のように multi-message modification を行う．

例として $a_{5,32} = 1$ であるメッセージがあったとし，それを $a_{5,32} = 0$ を満たすために m_1 を変更するとする．しかし，m_1 はラウンド 1 でも使用されるためこの変更によりすでに探索を終えているラウンド 1 のメッセージに影響が生じ，ラウンド 1 の十分条件を満たさなくなる恐れがある．そこで変更による影響を吸収するため次の 4 ステップ分についても修正する．この計算の詳細は表 1 に示す．

3.4 矢嶋と下山による衝突探索

矢嶋と下山は Wang と Yu の探索アルゴリズムを次のように再構築した．

(1) ステップ 0 から 15 について (1-i) から (1-iii) を実行する．

(1-i) 探索中のステップの出力が十分条件を満たすように出力を決定する．十分条件が存在しないビットについてはランダムに決定する．

(1-ii) single-message modification を実行する．ただし，ここでは差分値からではなく出力値からメッセージを導出する．例えばステップ 6 では次のように導出する．

$$m_6 = ((c_2 - d_2) \ggg 17) - const_6 - c_1 - \phi(d_2, a_2, b_1)$$

(1-iii) そのステップの出力差分が特定の値となっていることを確認する．なっていなければ (1-i) から再実行する．これを一定回繰り返し，それでも特定の値とならなければ前のステップに戻って探索をやり直す．

(2) ステップ 16 について (2-i) から (2-iii) を実行する．

(2-i) ステップ 16 の出力差分が特定の値となっていることを確認する．なっていなければ m_1 を導出し，再確認する．これを一定回数繰り返しても特定の値とならなければステップ 15 からやり直す．

(2-ii) ステップ 16 の出力を用いてステップ 17, 18(ブロック 2 ではステップ 17 のみ) の出力差分が特定の値となっていることを確認する．なっていなければ (2-i) から再実行する．

(2-iii) (2-i) で導出した m_1 を用いて multi-message modification を行いステップ 1 からステップ 5 の出力差分が特定の値となっていることを確認する．なっていなければ (2-i) からやり直す．

(3) ステップ 19(ブロック 2 ではステップ 18) 以降について MD5 の処理を行い，各ステップの出力差分が特定の値となっていることを確認する．なっていなければ (2) から再実行する．

矢嶋と下山はこのアルゴリズムを用いて追試を行い文献 [4] 十分条件にはブロック 1 の四つの条件 $a_{2,31} = 0$, $a_{2,30} = 0$, $a_{2,29} = 0$, $a_{2,27} = 0$, が不足していることを示した．ブロック 2 については，$a_{1,32} = 1$ を $a_{1,32} \mp bb_{0,32}$ と置き換え，さらに残りのステップについても 32 ビット目の条件を修正し，修正した十分条件を示している．

4. 十分条件の検証

文献 [2] で示されたプログラムをもとにし，ハッシュ値が衝突するメッセージペアを 1724 個作成した．得られたすべてのメッセージについて MD5 の処理を行いすべての内部変数の値を導出し，十分条件を満たす割合を調査した．その結果を表 2, 表 3 に示す．一つの衝突メッセージを作成するのに要した時間は約二時間である．

5. 理論的考察

本章では 4 章で得られた満たす必要のない条件の割合を理論的に考察する．ただし本章で扱う内部変数は 32 ビット目と 26 ビット目以外に差分を持つことがないのでこれら二つのビットに注目して議論する．

5.1 ブロック 1

(1) 条件 $\phi_{34,32} = 0$ について

ステップ 34 では入力差分は $\Delta c_8 = 0, \Delta d_9 = 0, \Delta a_9 =$

表 2　不必要な条件とそれを満たす割合 (ブロック 1)

条件	満たす割合 [%]
$\phi_{34,32}=0$	51.3
$a_{16,27}=0$	73.0
$c_{16,32}=d_{16,32}$	27.0

表 3　不必要な条件とそれを満たす割合 (ブロック 2)

条件	満たす割合 [%]
$\phi_{34,32}=1$	49.5
$d_{16,26}=1$	93.7
$c_{16,26}=1$	72.7
$b_{16,26}=1$	49.1

$0, \Delta b_8 = 0$ であるので $\Delta\phi_{34} = 0$ となる．また $\Delta m_{11} = 2^{15}$ であり，これを 16 ビット左回転シフトさせると 32 ビット目にうつる．したがって $\Delta c_9 = 2^{31}$ となり c_9 の出力差分は望み通りの差分となっている．よって，ステップ 34 では $\phi_{34,32} = 0$ は必要ないことがわかる．$\phi_{34,32} = 0$ は他の十分条件に影響しないので必要ない．出力差分を特定の値とするために $\phi_{34,32} = 0$ は必要ないので $\phi_{34,32} = 0$ を満たさない確率は 1/2 となり，これは表 2 の結果と一致する．

（2）　条件 $a_{16,27} = 0$ について

ステップ 61 の出力差分は $\Delta d_{16} = 2^{31} + 2^{25}$ であるため以下の計算から $\Delta\phi_{61} = 2^{31}$ となればよい．

$$\Delta u_{61} = \Delta d_{16} - \Delta a_{16} = 2^{25}$$
$$\Delta v_{61} = \Delta u_{61} \ggg 10 = 2^{15}$$
$$\Delta\phi_{61} = \Delta v_{61} - \Delta m_{11} - \Delta d_{15} = 2^{15} - 2^{15} - 2^{31} = 2^{31}$$

$\phi_{61} = b_{15} \oplus (a_{16} \vee (\neg c_{15}))$ と $a_{16,32} = c_{15,32}$ から $\Delta\phi_{61} = 2^{31}$ となる．ステップ 62 では以下の計算から $\Delta\phi_{62} = 2^{31}$ となればよいことがわかる．

$$\Delta u_{62} = \Delta c_{16} - \Delta d_{16} = 2^{31} + 2^{25} - 2^{31} - 2^{25} = 0$$
$$\Delta v_{62} = \Delta u_{62} \ggg 15$$
$$\Delta\phi_{62} = \Delta v_{62} - \Delta m_2 - \Delta c_{15} = 2^{31}$$

32 ビット目については $d_{16,32} = b_{15,32}$ とすればステップ 61 と同様にして $\Delta\phi_{62,32} = 1$ を得る．
26 ビット目については $b_{15,26} = 0$, $a_{16,26} = 1$ から

$$\Delta\phi_{62,26} = \phi'_{62,26} - \phi_{62,26}$$
$$= (1 \oplus (1 \vee (\neg 0))) - (1 \oplus (0 \vee (\neg 0))) = 0$$

となり，$\Delta\phi_{62} = 2^{31}$ を得る．ステップ 63 の出力差分は $\Delta d_{16} = 2^{31} + 2^{25}$ であるため以下の計算から $\Delta\phi_{61} = 2^{31}$ となればよい．

$$\Delta u_{63} = \Delta b_{16} - \Delta c_{16} = 2^{31} + 2^{25} - 2^{31} - 2^{25} = 0$$
$$\Delta v_{63} = \Delta u_{63} \ggg 21$$
$$\Delta\phi_{63} = \Delta v_{63} - \Delta m_9 - b_{15} = 2^{31}$$

26 ビット目については $a_{16,26} = 0$ と入力差分より $\Delta\phi_{63,26} = 0$ を得る．

32 ビット目については前の二つのステップと同様に考えて $\Delta\phi_{61,32} = 1$ とするためには $c_{16,32} = a_{16,32}$ が必要となる．ところが文献 [4] には $c_{16,32} = d_{16,32}$ と記載されており，これは $c_{16,32} = a_{16,32}$ と訂正する必要があることが分かる．ステップ 61 からステップ 63 の考察より $a_{16,27} = 0$ については満たす必要がないことがわかる．

また，$a_{16,27} = 0$ を満たすのはステップ 60 の計算で $u_{60,27} = b_{15,27}$ が成り立つ時で，この割合を調べてみると 73.2 %となっており，表 2 の結果と一致する．$u_{60,27} = 1$ となる確率は 1/2 で $b_{15,27} = 1$ についても 1/2 となっている．さらに $u_{60,27} = b_{15,27} = 1$ となる割合は $u_{60,27} = b_{15,27}$ となる割合のおよそ半分で 36.5 %である．$u_{60,27} = b_{15,27} = 0$ となる割合については 35.8 %となっており，二つの場合に偏りはほとんどないことがわかる．値の偏りがほとんどないにもかかわらず，$u_{60,27} = b_{15,27}$ が成り立つ．現在のところ $a_{16,27} = 0$ が成立する割合を理論的に説明できていない．

（3）　条件 $c_{16,32} = d_{16,32}$ について

この条件がステップ 62，63 の計算において特定の差分を得るために必要でないことは (2) のステップ 62 とステップ 63 の考察からわかる．そこで $c_{16,32} = d_{16,32}$ がブロック 1 の圧縮関数の出力が満たすべき十分条件の $cc_{0,32} = dd_{0,32}$ のために必要であると仮定する．
ブロック 1 の圧縮関数の出力は次のように計算される．

$$dd_0 = d_{16} + D,\ cc_0 = c_{16} + C$$

初期値の 32 ビット目をそれぞれ D_{32}，C_{32} と表すとする．$D_{32} = 0$ で $C_{32} = 1$ であるから算術加算において下位ビットからの桁上げがないとすれば $c_{16,32} \neq d_{16,32}$ でなければ $cc_{0,32} = dd_{0,32}$ とはならない．よって $c_{16,32} = d_{16,32}$ は必要ない．ブロック 1 の出力を求める加算において，下位ビットから 32 ビット目に桁上げが起こる確率は次のように求められる．

$$\frac{1}{4}(2^0 + 2^{-1} + 2^{-2} + \cdots + 2^{-30}) = 0.499\ldots \approx \frac{1}{2}$$

どちらか一方にだけ下位ビットからの桁上げが生じる場合，$c_{16,32} = d_{16,32}$ を満たす．よってこの条件を満たす確率は 1/4 となり，これは表 2 の結果と一致する．

5.2　ブロック 2

（1）　条件 $\phi_{34,32} = 0$ について

ステップ 34 では入力差分は $\Delta c_8 = 0, \Delta d_9 = 0, \Delta a_9 = 0, \Delta b_8 = 0$ であり，$\Delta\phi_{34} = 0$ となる．$\Delta m_{11} = -2^{15}$ となっている．これを 16 ビット左回転シフトすると $\Delta m_{11} = -2^{15}$ は 32 ビット目にうつる．したがって $\Delta c_9 = -2^{31}$ となり mod 2^{32} をとると $\Delta c_9 = -2^{31} = 2^{31}$ であるから，$\phi_{34,32} = 0$ を満たさなくともこのステップの出力差分値は望み通りの値となっておりこの条件は必要ないことが分かる．この条件は以降のステップの計算でも使われることはないので満たす必要はない．出力差分値を特定の値とするためにこの条件は必要ないので $\phi_{34,32} = 0$ を満たす確率は 1/2 となり，これは表 3 の結果と一致する．

（2） 条件 $d_{16,26}=1$ について

ステップ 61 では $dd_{0,26}=0$, $\Delta dd_{0,26}=1$ であり，衝突を起こすためには $dd'_{1,26}=dd_{1,26}$ となる必要がある．$dd'_{1,26}$ と $dd_{1,26}$ の計算は式で与えられる．

$$dd_1=dd_0+d_{16}\ ,\ dd'_1=dd'_0+d'_{16}$$

この計算より $dd_{1,26}=dd'_{1,26}$ を満たすには次の二つの場合が考えられる．

- $d_{16,26}=d'_{16,26}$ かつ 25 ビット目からどちらか一方にだけ桁上げがある場合
- $d_{16,26}\neq d'_{16,26}$ である場合

しかし，25 ビット目までは $dd_0=dd'_0$ かつ $d_{16}=d'_{16}$ であるため，桁上げが片方だけに生じることはない．したがって出力差分を $\Delta d_{16}=2^{31}-2^{25}$ とするためには $d_{16,26}\neq d'_{16,26}$ である必要がある．考えられる差分として

$$\begin{aligned}2^{31}-2^{25}&=2^{31}-2^{26}+2^{25}=\cdots\\&=2^{31}-2^{30}+2^{29}+\cdots+2^{25}\end{aligned}$$

の 6 通りがあり，どの差分であっても $d_{16,26}\neq d'_{16,26}$ を満たす．差分が $2^{31}-2^{25}$ である場合は $d_{16,26}=1$ が必要であるが $2^{31}-2^{26}+2^{25}$ などである場合については $d_{16,26}=1$ を満たさなくてもよい．このように複数の差分が可能である理由としては d_{16} を用いてさらに d の値を更新することはないためと考えられる．よって，$\Delta d_{16}=2^{31}-2^{25}$ を得るために $d_{16,26}=1$ は必要ではない．ステップ 62 では出力差分が $\Delta c_{16}=2^{31}-2^{25}$ であるので以下の計算から $\Delta\phi_{62}=2^{31}$ であればよいことがわかる．

$$\begin{aligned}\Delta u_{62}&=\Delta c_{16}-d_{16}=2^{31}-2^{25}-2^{31}+2^{25}=0\\\Delta u_{62}&=\Delta u_{62}\ggg 15=0\\\Delta\phi_{62}&=\Delta v_{62}-\Delta m_2-\Delta c_{15}=-2^{31}=2^{31}\end{aligned}$$

32 ビット目については $d_{16,32}=b_{15,32}$ から $\Delta\phi_{62,32}=1$ を得る．26 ビット目については差分が -2^{25} であるので考えられる差分として

$$-2^{26}+2^{25}=-2^{27}+2^{26}+2^{25}=\cdots=-2^{30}+2^{29}+\cdots+2^{25}$$

がある．差分が $2^{31}-2^{25}$ である場合は $d_{16,26}=1$ が必要であるが差分が $2^{31}-2^{26}+2^{25}$ などである場合については $d_{16,26}=1$ を満たさなくても望み通りの差分が得られる．

ステップ 63 では出力差分が $\Delta b_{16}=2^{31}-2^{25}$ であるので以下の計算から $\Delta\phi_{63}=2^{31}$ であればよいことがわかる．

$$\begin{aligned}\Delta u_{63}&=\Delta b_{16}-c_{16}=2^{31}-2^{25}-2^{31}+2^{25}=0\\\Delta u_{63}&=\Delta u_{63}\ggg 21=0\\\Delta\phi_{63}&=\Delta v_{63}-\Delta m_9-\Delta b_{15}=-2^{31}=2^{31}\end{aligned}$$

32 ビット目については $c_{16,32}=a_{16,32}$ から $\Delta\phi_{63,32}=1$ となる．
26 ビット目についてはステップ 62 と同様に，考えられる差分が複数あるため必ずしも $d_{16,26}=1$ が必要とは言えない．

ステップ 61 の計算において $u_{61,26}=1$ となる確率は約 1/2 で $a_{16,26}=1$ となる確率は 1 であるので，$d_{16,26}=1$ となる確率は 1/2 となるはずであるが，表 3 からわかるようにこの条件を満たす確率は非常に大きくなっている．これはこのステップ以降での差分を特定の値とするために $d_{16,26}=0$ となった場合はさらに d_{16}，もしくは他の内部変数が満たすべき条件があるからだと考えられる．

（3） 条件 $c_{16,26}=1$ について

ステップ 62 では $cc_{0,26}=0$, $\Delta cc_{0,26}=1$ であり，衝突を起こすためには $cc'_{1,26}=cc_{1,26}$ となる必要がある．$cc'_{1,26}$ と $cc_{1,26}$ の計算は式で与えられる．

$$cc_1=cc_0+c_{16}\ ,\ cc'_1=cc'_0+c'_{16}$$

この計算より $cc_{1,26}=cc'_{1,26}$ を満たすには次の二つの場合が考えられる．

- $c_{16,26}=c'_{16,26}$ かつ 25 ビット目からどちらか一方にだけ桁上げがある場合
- $c_{16,26}\neq c'_{16,26}$ である場合

しかし，25 ビット目までは $cc_0=cc'_0$ かつ $c_{16}=c'_{16}$ であるため，桁上げが片方だけに生じることはないので出力差分を $\Delta c_{16}=2^{31}-2^{25}$ とするためには $c_{16,26}\neq c'_{16,26}$ である必要がある．また，考えられる差分として

$$\begin{aligned}2^{31}-2^{25}&=2^{31}-2^{26}+2^{25}=\cdots\\&=2^{31}-2^{30}+2^{29}+\cdots+2^{25}\end{aligned}$$

の 6 通りがあり，どの差分であっても $c_{16,26}\neq c'_{16,26}$ をみたす．差分が $\Delta c_{16}=2^{31}-2^{25}$ であれば $c_{16,26}=1$ が必要であるが，差分が $2^{31}-2^{26}+2^{25}$ などである場合は必要ではない．
よって，$\Delta c_{16}=2^{31}-2^{25}$ を得るために $c_{16,26}=1$ は必要でない．ステップ 63 では $\Delta\phi_{63}=2^{31}$ となればよい．32 ビット目については $c_{16,32}=a_{16,32}$ から $\Delta\phi_{63,32}=1$ となる．26 ビット目については考えられる差分が複数あるため $d_{16,26}=1$ は必ずしも必要ではない．

（4） 条件 $b_{16,26}=1$ について

$bb_{0,26}=0$, $\Delta bb_{0,26}=1$ であり，衝突を起こすためには $bb'_{1,26}=bb_{1,26}$ となる必要がある．$bb'_{1,26}$ と $bb_{1,26}$ の計算は式で与えられる．

$$\begin{aligned}bb_1&=bb_0+b_{16}\\bb'_1&=bb'_0+b'_{16}\end{aligned}$$

この計算より $bb_{1,26}=bb'_{1,26}$ を満たすには次の二つの場合が考えられる．

- $b_{16,26}=b'_{16,26}$ かつ 25 ビット目からどちらか一方にだけ桁上げがある場合
- $b_{16,26}\neq b'_{16,26}$ である場合

しかし，25 ビット目までは $bb_0=bb'_0$ かつ $b_{16}=b'_{16}$ であるため，桁上げが片方だけに生じることはないので出力差分を $\Delta b_{16}=2^{31}-2^{25}$ とするためには $b_{16,26}\neq b'_{16,26}$ である必要がある．

また，考えられる差分として

$$2^{31} - 2^{25} = 2^{31} - 2^{26} + 2^{25} = \cdots$$
$$= 2^{31} - 2^{30} + 2^{29} + \cdots + 2^{25}$$

の6通りがあり，どの差分であっても $b_{16,26} \neq b'_{16,26}$ をみたす．差分が $\Delta b_{16} = 2^{31} - 2^{25}$ であれば $b_{16,26} = 1$ が必要であるが，差分が $2^{31} - 2^{26} + 2^{25}$ などである場合は必要ではない．よって，$\Delta b_{16} = 2^{31} - 2^{25}$ を得るために $b_{16,26} = 1$ は必要でない．

上述のように最後の3ステップにおいては複数の差分が考えられる．実際に衝突を起こしているメッセージを1ビットずつ確認するとブロック2の三つの条件 $d_{16,26} = 1, c_{16,26} = 1, b_{16,26} = 1$ を満たさないものは各ステップの差分が $2^{31} - 2^{26} + 2^{25}$ などとなっている．

以上から文献[5]の Appendix2 に示される First Message Block 3R の $\phi_{34,32} = 0$ と 4R の $a_{16,27} = 0$ は必要なく，$c_{16,32} = d_{16,32}$ については $c_{16,32} = a_{16,32}$ と修正するべきである．また Second Message Block については 3R の $\phi_{34,32} = 0$ と 4R の $d_{16,26} = 1$, $c_{16,26} = 1$, $b_{16,26} = 1$ も必要ないことがわかった．

6. 結　　論

本論文では従来示されている十分条件には不必要なものがあるのではないかと考え，矢嶋と下山の十分条件を用いて実際に衝突メッセージを作成し，作成したメッセージペアが十分条件を満たすかどうか確認した．具体的な確認方法としては衝突メッセージを MD5 の入力として内部変数を全て求め，それぞれについて条件を満たすものとそうでないものを確認した．

その結果，4章で示したように合計七つの不要な条件が得られた．このうちブロック1の $\phi_{34,32} = 0$ とブロック2の $\phi_{34,32} = 1$ については満たす割合が約 50 % なのでランダムに決まっていると考えられる．さらにブロック1の $c_{16,32} = d_{16,32}$ は誤りであり，正しくは $c_{16,32} = a_{16,32}$ であることがわかった．そのほかの四つの条件についても必要でないことが理論的に示した．

文　　献

[1] H.Dobbertin, "Cryptanalysis of MD5 compress," http://www.cs.ru.ac.za/courses/Honours/mmcourse/security/md5/dobbertin.pdf, 1996.

[2] P.Stach and V.Liu, "MD5 collision generation," http://www.stachliu.com/collisions.html, 2005.

[3] R.Rivest, "The MD5 message-digest algorithm," RFC1321, 1992.

[4] X. Wang and H. Yu, "How to break MD5 and other hash functions," http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf, 2004.

[5] 矢嶋純，下山武司，"MD5 のコリジョン探索および sufficient conditions について，" 電子情報通信学会 技術報告書，ISEC2005-78, pp.15–22 2005.