SHA-256/-384/-512 の評価報告

東京理科大学理工学部電気電子情報工学科
金子敏信

2006 年 3 月

概 要

SHA-256/-384/-512 は 2002 年に FIPS 180-2 SECURE HASH STANDARD として SHA-1 と共に、規定されたハッシュ関数群の中の 256 ビット以上のハッシュ値を出力するＭＤ型ハッシュ関数である。これらはハッシュビット長およびその構造から、SHA-1 の上位バージョンと想定され、まとめて SHA-2 とも呼ばれる。2004 年以降の Wang らの攻撃により、SHA-1 の安全性に関し、疑問が呈されて活発な議論が行われている。本報告では、SHA-2 の安全性に関する Gilbert の 2003 年の論文以来、2006 年１月末までに公表された論文で、SHA-2 の安全性係わる記述が見られるものを網羅しまとめた。Wang らの一連のハッシュ関数の衝突探索の基本的な戦略は、

(1.) 差分パスの局部衝突の重ね合わせで大域的な衝突を見つける (disturbance vector 探索)

(2.) XOR 差分と符号付き算術差分を同時に解析し、確率の高い差分パスを探索する

(3.) 差分伝播の十分条件を解析し、確率を高く保つメッセージ語変更法を見つけること

と考えることが出来る。

SHA-2 に関しても、現在までの研究で、同様な戦略がとられている。Gilbert らは、SHA-2 の圧縮関数を線形近似し、XOR 差分について、9 ステップで確率 2^{-66} の局部衝突を導いている。これは最良の局部衝突であるが、大域的衝突には、SHA-256 の場合、それが少なくとも３個以上、SHA-384/-512 においては５個以上必要と見積もられ、SHA-2 は (1.) の戦略のみを持つ Joux 型の攻撃に対し安全であると結論づけている。

Hawkes らは、この局部衝突のみに,(2.),(3.) の戦略を適用している。符号付き算術差分も合わせて考察し、拡張メッセージワード差分変更法を提案している。それにより、確率が 2^{-39} から 2^{-42} に上昇することを示している。

Lee らは、(3.) のメッセージ変更法が存在すると仮定して、SHA-256 の第 24 から 55 ステップを調査し、衝突には、15 個以上の局部衝突の重ね合わせが必要であり、Hawkes らの結果と合わせることにより、衝突発見の確率は、2^{-585} 以下であるとしている。彼らは、この確率は、思いの外大きいとしているが、SHA-256 の安全性を脅かす結果ではない。

Yoshida らは、SHA-2 の全ての算術加算を XOR 加算で置き換えた SHA-2-XOR の差分特性を調査している。確率 2^{-8} の１ステップ繰り返し差分パスを発見しそれを用い、15 ステップ SHA-2-XOR の擬似衝突が 2^{120} の計算量で発見できること、31 ステップ SHA-2-XOR は、ランダムハッシュ関数では無いことを示している。先頭 19 ステップの差分確率を１とするメッセージ変更法を示し、それを用いて、34 ステップ SHA-2-XOR の擬似衝突が、計算量 2^{120} で発見できると推定している。

Pramstaller らは、予備的調査として、SHA-256 のメッセージ拡張関数の効果を調べている。算術加算を XOR で置き換えて簡略化したメッセージ拡張関数の出力を線形符号と考え、確率的最小重み探索法を使って拡張メッセージ差分の最小ハミング重みが 42 ステップまでで 35 であることを求め、それをい、フルステップで最小ハミング重み 356 と推定している。

以上この３年間の SHA-2 の安全性評価研究を総合するに、現時点の解析結果は、SHA-2 の衝突探索の成功確率は、誕生日攻撃の確率を上回るものではなく、その安全性を脅かすものではないと考えられる。

第1章　はじめに

SHA-256/-384/-512 は 2002 年に FIPS 180-2 SECURE HASH STANDARD として SHA-1 と共に、規定されたハッシュ関数群の中の 256 ビット以上のハッシュ値を出力するＭＤ型ハッシュ関数である。これらはハッシュビット長およびその構造から、SHA-1 の上位バージョンと想定され、まとめて SHA-2 とも呼ばれる。2004 年以降の Wang らの攻撃により、SHA-1 の安全性に関し、疑問が呈され、現在、活発な議論が行われている。本報告では、SHA-2 の安全性に関する Gilbert の 2003 年の論文以来、2006 年１月末までに公表された論文で、SHA-2 の安全性係わる記述が見られるものを網羅しまとめた。Wang らの一連のハッシュ関数の衝突探索の基本的な戦略は、

戦略 1. 差分パスの局部衝突の重ね合わせでメッセージ拡張関数通過後の語に対し成り立つ効果的な大域的な衝突（ハッシュ関数の衝突）を見つけること (disturbance vector 探索)

戦略 2. XOR 差分のみで無く符号付き算術差分も同時に解析し、確率高く伝播する差分パスを探索すること

戦略 3. 望ましい差分伝播が起こる内部状態に関する十分条件を解析し、差分伝播確率を高く保つメッセージ語変更法を見つけること

と考えることが出来る。　SHA-2 に関しても、現在までの研究で、同様な戦略がとられている。本報告書では、準備として、第 2 章で、ハッシュ関数の基本と SHA-2 のアルゴリズムを紹介し、それ以降の節で、SHA-2 の安全性に関する研究として Gilbert [3], Hawkes [4], 吉田 [9], Lee [5], Pramstaller [8] らの論文をまとめる。

第2章　ハッシュ関数SHA-256/384/512

2.1　ハッシュ関数の安全性

ハッシュ関数とは、与えられた任意長のデータ X から、固定長の値（ハッシュ値）$Y = H(X)$ を生成するデータ圧縮関数 $H(X)$ を言う。暗号の分野で使用されるハッシュ関数は、暗号論的ハッシュ関数と呼ばれ、データを効率よく圧縮する機能だけでなく、次の性質を満たす必要がある。

- 一方向性:ハッシュ値 $Y = H(X)$ が与えられたとき、X を求める事が計算量的に困難であること。
- 第二原像発見困難性：データ X 及び、そのハッシュ値 $Y = H(X)$ が与えられたとき、同一のハッシュ値 y を与えるデータ X' を求める事が計算量的に困難であること。
- 衝突発見困難性：同一のハッシュ値を与える $(H(X) = H(X'))$ 異なるデータ X,X' の対を求める事が計算量的に困難であること。

これらの計算は、衝突発見、第二原像発見、一方向性、の順により困難となる。ハッシュ値のビットサイズが n の理想的ハッシュ関数の場合、これらの、計算量は、一方向性、第二原像発見困難性に関し、2^n、衝突発見困難性に関し、$2^{n/2}$ である。[1] 実在のハッシュ関数の安全性は、それぞれの計算困難性が、理想的ハッシュ関数のそれ、2^n、$2^{n/2}$ より小さな具体的攻撃法があるか否かで評価される。

2.2　ハッシュ関数 SHA-256/384/512

ハッシュ関数 SHA-256、SHA-384、SHA-512 は、2000 年に米国標準技術局 (NIST) により提案されたハッシュ関数であり、2002 年にハッシュ関数 SHA-1 と共に FIPS 180-2 として制定されている。SHA-1 の上位規格であり、SHA-256,SHA-384,SHA-512 は、ひとまとめにして、通称 SHA-2 と呼ばれる。SHA-1 及び SHA-2 は、現時点 (2006 年 3 月) において、CRYPTREC 推奨暗号リストのハッシュ関数である。SHA-256、SHA-384、SHA-512 は、よく似た構造を持つので、以下では、SHA-256 のアルゴリズムを中心に紹介し、SHA-384、SHA-512 に関しては、相違点のみ簡単に補足する。

2.2.1　記号と関数の定義

SHA-256 では、1 ワード=32 ビットが、基本の演算単位である。本報告書で、SHA-256 の定義や攻撃法の説明の際に、使用する演算記号を以下に説明する。

[1] ビットサイズ n のランダムデータ Y を約 $2^{n/2}$ 個、用意すれば、その中に同一の値が見つかる確率が 1/2 以上となる。この性質を誕生日一致パラドックスといい、それを使った衝突発見攻撃を誕生日攻撃という

$+$：2^{32} を法とした算術加算
$\oplus$：ビット毎の排他的論理和
$\vee$：ビット毎の論理和
$\wedge$：ビット毎の論理積
$\|$：ビットの連結
$ROTR^y(x)$：x を右に y ビット巡回シフト
$SHR^y(x)$：x を右に y ビットシフト
$\bar{x}$：x のビット反転
ΔX：X の排他的論理和差分
δx：x の算術和差分 $X[i]$：X の i ビット目

SHA-384 及び SHA-512 に言及する場合、1 ワード=64 ビットとして、同じ記号を用いる。
SHA-256 では、以下の 6 種類の論理関数が用いられる。

$$Ch(x,y,z) = (x \wedge y) \vee (\bar{x} \wedge y) \tag{2.1}$$
$$Maj(x,y,z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z) \tag{2.2}$$
$$\begin{aligned} \Sigma_0(x) &= ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x) \\ \Sigma_1(x) &= ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x) \end{aligned} \tag{2.3}$$
$$\begin{aligned} \sigma_0(x) &= ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x) \\ \sigma_1(x) &= ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x) \end{aligned} \tag{2.4}$$
$$\tag{2.5}$$

SHA-384 及び SHA-512 では、1 ワード=64 ビットとしている他に、Σ 及び σ 関数においてシフトビット数が異なり、以下である。

$$\begin{aligned} \Sigma_0(x) &= ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x) \\ \Sigma_1(x) &= ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x) \end{aligned} \tag{2.6}$$
$$\begin{aligned} \sigma_0(x) &= ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x) \\ \sigma_1(x) &= ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x) \end{aligned} \tag{2.7}$$
$$\tag{2.8}$$

2.2.2　SHA-256 の処理

SHA-256 は、2^{64} ビット未満のメッセージから 256 ビットのハッシュ値を生成するハッシュ関数である。SHA-384/512 は、2^{128} ビット未満のメッセージから 384/512 ビットのハッシュ値を生成する。何れの、ハッシュ関数でも、メッセージを所定の長さにそろえる前処理を行った後、メッセージ圧縮関数を適用する。

2.2.3　SHA-256 の前処理

SHA-256 の前処理（パディング処理）は、以下である。

1. 入力メッセージ M に対し、メッセージ長が 512 ビットの倍数になるように、M の末尾に下記のようなデータを付加する。

$$M\|1\|0^k\|l \tag{2.9}$$

ただし、l は M のメッセージ長の二進数表現 (64 ビット)、k は $l+1+k \equiv 448 (\mathrm{mod}\ 512)$ を満たす正の最小値である。この処理をパディングという。

2. パディングされたメッセージは、N 個の 512 ビット単位のブロック $M^{(i)}$ に分割される。

$$M = M^{(1)}\|M^{(2)}\|\cdots\|M^{(i)}\|\cdots\|M^{(N)} \tag{2.10}$$

ただし、各々の $M^{(i)}$ は、16 個のワード

$$M^{(i)} = M_0^{(i)}\|M_1^{(i)}\|\cdots\|M_{15}^{(i)} \tag{2.11}$$

からなる。

3. 初期値として

$$\begin{aligned} H_0^{(0)} &= \texttt{6a09e667} \\ H_1^{(0)} &= \texttt{bb67ae85} \\ H_2^{(0)} &= \texttt{3c6ef372} \\ H_3^{(0)} &= \texttt{a54ff53a} \\ H_4^{(0)} &= \texttt{510e527f} \\ H_5^{(0)} &= \texttt{9b05688c} \\ H_6^{(0)} &= \texttt{1f83d9ab} \\ H_7^{(0)} &= \texttt{5be0cd19} \end{aligned} \tag{2.12}$$

を設定する。

SHA-384/512 の前処理では、メッセージ長が 1024 の倍数となるようにパディングが施されること、ブロック $M^{(i)}$ の長さが 1024 ビットであること、初期値 $h_0^{(0)} \ldots h_7^{(0)}$ が、SHA-384/512 の固有の 64 ビット定数であること、が SHA-256 と異なっている。

2.2.4　SHA-256 のハッシュ値計算

N 個のメッセージブロック $M^{(i)}, ..., M^{(N)}$ の $M^{(i)}$ に対して、$1 \le i \le N$ の順に以下を実行する。

メッセージ拡張

次式で定義する SHA-256 メッセージ拡張関数を用いて拡張メッセージ W_t を計算する。

$$\begin{aligned} W_t &= M_t^{(i)} && (0 \le t \le 15) \\ W_t &= \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} && (16 \le t \le 63) \end{aligned} \tag{2.13}$$

(2.14)

SHA-384/512 の場合も、メッセージ拡張関数は同じであるが、処理の単位が 64 ビットになり、σ 関数が、式 (2.7) の SHA-512 用の関数に変わり、再帰関係式 (2.13) が、$(16 \leq t \leq 79)$ の範囲で適用され、拡張メッセージ $W_t(0 \leq t \leq 79)$ が作られる。

圧縮関数

1. 8 個のバッファ変数を $(i-1)$ 番目のハッシュ値 $H^{(i-1)}$ で初期化する。

$$\begin{aligned} a_0 &= H_0^{(i-1)} \\ b_0 &= H_1^{(i-1)} \\ b_0 &= H_2^{(i-1)} \\ d_0 &= H_3^{(i-1)} \\ e_0 &= H_4^{(i-1)} \\ f_0 &= H_5^{(i-1)} \\ g_0 &= H_6^{(i-1)} \\ h_0 &= H_7^{(i-1)} \end{aligned} \tag{2.15}$$

2. $0 \leq t \leq 63$ に対して以下の計算を繰り返す。

$$\begin{aligned} T_1 &= H_t + \Sigma_1(e_t) + Ch(e_t, f_t, g_t) + K_t + W_t \\ T_2 &= \Sigma_0(a_t) + Maj(a_t, b_t, c_t) \\ h_{t+1} &= g_t \\ g_{t+1} &= f_t \\ f_{t+1} &= e_t \\ e_{t+1} &= d_t + T_1 \\ d_{t+1} &= c_t \\ c_{t+1} &= b_t \\ b_{t+1} &= a_t \\ a_{t+1} &= T_1 + T_2 \end{aligned} \tag{2.16}$$

   ただし、K_t は 32 ビットの定数である。また、ここでの 1 回の演算を、以下本稿では 1 ステップとする。

3. i 番目の中間ハッシュ値を

$$\begin{aligned} H_0^{(i)} &= H_0^{(i-1)} + a_{64} \\ H_1^{(i)} &= H_1^{(i-1)} + b_{64} \\ H_2^{(i)} &= H_2^{(i-1)} + c_{64} \\ H_3^{(i)} &= H_3^{(i-1)} + d_{64} \\ H_4^{(i)} &= H_4^{(i-1)} + e_{64} \\ H_5^{(i)} &= H_5^{(i-1)} + f_{64} \\ H_6^{(i)} &= H_6^{(i-1)} + g_{64} \\ H_7^{(i)} &= H_7^{(i-1)} + h_{64} \end{aligned} \tag{2.17}$$

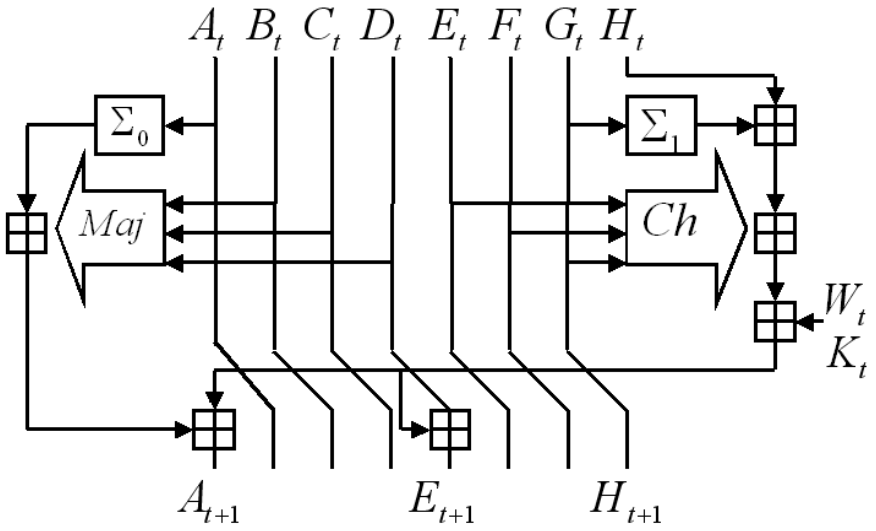

図 2.1: 1 ステップの構造

で計算する。ここまでの 1 回の処理を、1block とする。
上記手続きを N 回繰り返した最終的な 256 ビットの値

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} \tag{2.18}$$

がメッセージ M のハッシュ値である。

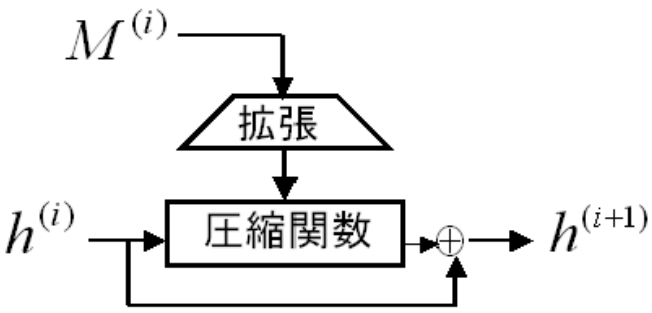

図 2　　1block の構造

SHA-384/512 の場合、全ての変数が、64 ビット変数になること、定数 K_t が SHA-512 用の定数となること、拡張メッセージ $W_t (0 \leq t \leq 79)$ に対して 80 ステップの処理が行われる事が圧縮処理の相違点である。最終的なハッシュ値に関しては、SHA-512 の場合、メッセージ M に対して、512 ビット値

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)} \tag{2.19}$$

が、ハッシュ値となるが、SHA-384 の場合、$H^{(N)}$ の左 384 ビット

$$H^{(N)} = H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \tag{2.20}$$

がハッシュ値となる。

第3章　Gilbertらの解析

Gilbert らは論文 [3] において、SHA-2 の圧縮関数を線形近似し、XOR 差分について、9 ステップで確率 2^{-66} の局部衝突を導いている。この局部衝突は最良のものであるが、大域的衝突を構成するには、SHA-256 の場合、この局部的衝突が少なくとも 3 個以上、SHA-384/-512 においては 5 個以上必要と見積もられ、SHA-2 は、戦略 1 のみを使う Joux 型の衝突発見攻撃に対し安全であると結論づけている。以下、その概要を紹介する。

3.1　局部衝突

3.1.1　XOR 差分、Ch 関数、Maj 関数

二つの 2 元ベクトル X,X' に対し、XOR 差分を、次式で表す。

$$\Delta X = X \oplus X' \tag{3.1}$$

二つのメッセージ X,X' に対し、衝突を見つける事は、$\Delta X \neq 0$ のメッセージ対に対し $\Delta H(X) = 0$ なるなる物を発見することになる。

SHA-2 においては、選択関数 $Ch(x, y, z)$(式 (2.1)) , 多数決関数 $Maj(x, y, z)$(式 (2.2)) が使われている。これらは、32 ビット (又は 64 ビット）変数を並列に処理する関数である。入力 (x, y, z) と出力 Ch、Maj において対応するビット間で、XOR 差分の伝播を調べれば、表 3.1 である。

この表に於いて、ΔCh や ΔMaj 欄が'0' は、出力差分が常に'0'、'1' は常に'1' を表す。'0/1' は、確率 1/2 で、出力差分が、'0' に（又は'1' に）なることを表す。

表 3.1: Ch,Maj 関数の差分伝播

ΔX	ΔY	ΔZ	ΔCh	ΔMaj
0	0	0	0	0
0	0	1	0/1	0/1
0	1	0	0/1	0/1
0	1	1	1	0/1
1	0	0	0/1	0/1
1	0	1	0/1	0/1
1	1	0	0/1	0/1
1	1	1	0/1	1

3.2　9 ステップ局部衝突

ハッシュ値が衝突を発見する為に、ハッシュ関数の内部状態の衝突に着目する。SHA-0 に対する Joux と Chabaud の攻撃では, 次の手順を踏む。

1. メッセージ語 W_t の 1 ビットの変化で引き起こされる圧縮関数の内部状態の変化を打ち消す可能性の高い拡張メッセージ差分系列 ΔW_i,$(i = t+1, \dots)$ を見いだす。（1 ビット擾乱パターン：1-bit perturbative patterns 又は局部衝突:local collision）[1]

2. 局部衝突の合成になっている拡張メッセージ差分系列でメッセージ拡張関数の再帰関係式を満足する物を探し、衝突する可能性の高いメッセージ差分を求める。

圧縮関数の非線形性を線形近似する為、局部衝突を与える拡張メッセージ差分系列におけるハミング重みが小さいほど、内部状態の変化を打ち消す確率が高くなる。

SHA-2 の圧縮関数内の Σ 関数で使われている算術加算も、XOR 差分に対して、非線形関数として影響する。Gilbert らは、ハミング重み 1 の差分に対して, 算術加算が XOR 加算と同じ出力差分を与える確率は 1/2,Ch 及び Maj 関数の出力差分が 0 となる確率も 1/2 として,1 ビット擾乱パターンを求めている。

結果の、9 ステップ局部衝突を示せば, 表 3.2 である。W_i に入力した 1 ビット擾乱が、内部状態 a, b, c, d, e, f, g, h に引き起こす差分のハミング重みが、表の第 1 行に示してある。この擾乱で、a_i 及び e_i に重み 1 の擾乱が発生している。この差分は、次のステップで Σ_0,Σ_1 関数により、それぞれ重み 3 の擾乱に拡散するが、それを打ち消す為に W_{i+1} に重み 6 の差分を入れれば、a_{i+1} の擾乱が打ち消され、$b_{i+1}, e_{i+1}, f_{i+1}$ に $1, 3, 1$ の重みの擾乱となることが期待できる。これが、2 行目に示されている。このように、W の列に示してある、重み $1, 6, 9, \cdots, 1$ の差分を $W_i, W_{i+1}, W_{i+2} \cdots, W_{i+8}$ を入れれば、$(i+8)$ ステップにおいて、全ての内部状態の差分が 0 になることが期待される。1 ビット擾乱 ΔW_i に対し、表 3.2 の局部衝突を与えるメッセージ差分は,$\Delta W_{i+1} = \Sigma_1(\Delta W_i) \oplus \Sigma_0(\Delta W_i)$;$DeltaW_{i+2} = \Sigma_1(\Sigma_0(\Delta W_i))$; $\Delta W_{i+3} = 0$; $\Delta W_{i+4} = \Delta W_i$; $\Delta W_{i+5} = \Delta W_{i+1}$; $\Delta W_{i+\{6,7\}} = 0$;$\Delta W_{i+8} = \Delta W_i$ である。

Gilbert らは、Σ 関数の拡散効果を考慮して、ハミング重みの低い局部衝突を与える擾乱として、1 ビット擾乱のみを解析しているが、Hawkes らは、論文 [4] において、これが、最良の局部衝突としている。

3.3　9 ステップ局部衝突の確率

表 3.2 の局部衝突が起こるためには、差分が存在するビットについて、算術和が XOR と同じ振る舞いをする事、及び Ch,Maj 関数において、出力差分が 0 になることが必要である。SHA-2 の圧縮関数において、T_1 と内部状態 d の加算で、新しい内部状態 e とするときと、T_1 と T_2 の加算で、新しい内部状態 a を作るとき算術加算が使われる。算術加算においてキャリが発生する確率の上界は 1/2 と評価される。1 ステップ当たり 2 回の算術加算があることから、1 ビットの非零差分当たり、確率は $(1/2)^2$ となる。表の $W_i, \cdots, W_{i+8}$ には、$1+6+9+1+6+1 = 24$ ビットの非零差分があるので、確率は $(2^{-24})^2 = 2^{-48}$ である。さらに、Ch,Maj に、18 ビットの非零差分があ

[1] 変化が打ち消され、内部状態差分がゼロ（即ち、内部状態が衝突）となっている事を局部衝突、1 ビットの変化で引き起こされた擾乱を 1 ビット擾乱パターン、また、その擾乱パターンを拡張メッセージ差分系列で訂正するという意味で、その差分系列を擾乱訂正パターン corective patterns と区別する場合もある。

表 3.2: 9 ステップ局部衝突

i	ΔW	Δa	Δb	Δc	Δd	Δe	Δf	Δg	Δh
i	1	1	0	0	0	1	0	0	0
$i+1$	6	0	1	0	0	3	1	0	0
$i+2$	9	0	0	1	0	0	3	1	0
$i+3$	0	0	0	0	1	0	0	3	1
$i+4$	1	0	0	0	0	1	0	0	3
$i+5$	6	0	0	0	0	0	1	0	0
$i+6$	0	0	0	0	0	0	0	1	0
$i+7$	0	0	0	0	0	0	0	0	1
$i+8$	1	0	0	0	0	0	0	0	0

り、その確率は 2^{-18} と評価される。従って、ランダムなメッセージに対し、この表の局部衝突が起こる確率は 2^{-66} である。

3.4　SHA-2 の衝突確率の評価

3.4.1　SHA-256

SHA-256 では、圧縮関数は 64 ステップの処理を行う。そのときに使う拡張メッセージ系列は、式 (2.13) で表されており、引き続く 16 個の拡張メッセージ W_i の差分が 0 の場合、拡張メッセージ全体の差分が 0 になってしまう。9 ステップ局部衝突の合成で、64 ステップの衝突を作るためには、$15+9+15+9+15 = 63 < 64$ であるため、9 ステップ局部衝突 2 個の合成では不十分であり、最低 3 個合成する必要がある。その衝突発見確率の上界は 2^{-132} と見積もられ、その計算量は、誕生日攻撃の 2^{128} を上回る。従って、この論文で述べた手法は、有効な衝突発見攻撃とは言えない。

3.4.2　SHA-384/512

SHA-384/512 において、圧縮関数は 80 ステップの処理を行う。拡張メッセージ系列の再帰関係式 (2.13) の中の *sigma* 関数を調べると、9 ステップ局部衝突の終端から、最大 7 ステップの零差分系列は、存在しうるが、8 ステップ目には、必ず、非零差分が拡張メッセージ系列に出現する。$15+9+7+9+7+9+7+9+7 = 79 < 80$ であるため、衝突を、9 ステップ局部衝突の合成で作るには、4 こでは不十分で、最低 5 個、必要である。その衝突発見確率の上界は 2^{-264} と見積もられ、その計算量は、誕生日攻撃の 2^{254} を上回る。従って、この論文で述べた手法は、有効な衝突発見攻撃とは言えない。

3.5　その他の攻撃

Gilbert らは、その他の攻撃として、Dobertin 流の攻撃、擬似衝突の発見可能性について考察し、SHA-2 の場合、何れも誕生日攻撃より、攻撃計算量が大きいであろうと推測している。

MD4 や MD5 に対する衝突を見つけた Dobertin 流の攻撃では、それらの、ハッシュ関数のメッセージ拡張関数が、メッセージ語の並び順を変えているだけの単純なものであるから、適用可能であり、SHA-2 のような、複雑な、メッセージ拡張関数にそのまま適用することは難しいとしている[2]。

擬似衝突として、Gilbert らは、同一のメッセージブロック M に対し、異なる 2 つのハッシュ連鎖変数 H,H' が、同じハッシュ値を与えるものを考察している。ハッシュ関数の圧縮関数部分を、ハッシュ連鎖変数を平文入力とメッセージブロックを鍵入力、最終内部状態を暗号文とする暗号化関数と考えれば、SHA は、Davis-Meyer 構造を持つ。そこでは、ハッシュ連鎖変数は、式 (2.17) のように、暗号化関数の出力と加算され、次のメッセージブロックの為のハッシュ連鎖変数となる。SHA-2 におけるこの暗号化関数を SHACAL-2 と呼ぶことにして、その差分特性を調べれば、衝突発見の手がかりとなる[3]。彼らは、SHACAL-2 の 4 ステップに対して、確率 2^{-8} の差分パスを見つけている。それは、複数回の接続が不可能なパスであるが、接続可能と仮定するならば、SHACAL-2 において、SHA-256 ならば 16 回、SHA-384/512 ならば 20 回接続すれば、それぞれ、2^{-128},2^{-160} の確率を持つ差分パスとなる。しかし、この差分パスが使えたとしても、誕生日攻撃よりも効率のよい衝突発見攻撃にはならないとしている。

最後に、彼らは、変形版 SHA-256 として SHA"'-256 を考え、それの衝突発見が非常に容易である事を示している。そこでは、SHA-256 に対して

1. 全ての定数 $K_0, \cdots, K_63$ が 4 ビットパターンの繰り返し[4]
2. 初期値 $H_0^{(0)}, \cdots, H_7^{(0)}$ が 4 ビットパターンの繰り返し
3. 全ての算術加算 $+$ は、排他的論理和 $\oplus$ で置き換え
4. 全てのシフト演算 $SHR(x)$ は、巡回シフト演算 $ROTR(x)$ で置き換え

の変形が成されている。このような変形をするならば、処理過程全体に 4 ビットパターンの繰り返し（対称性）が発生するので、衝突探索すべきメッセージも 4 ビットパターンの繰り返しで良いことになる。その下で、計算機探索を行い、SHA"'-256 の衝突発見が驚くほど簡単であるとしている。しかし、この評価が、SHA-256 の安全性評価にどの程度関係するかは、今後の課題であろう。

[2]Dobertin 型攻撃の精神を、圧縮関数を解析して、衝突発見確率を上場させるようなメッセージを選択すると理解するならば、その考え方は、Wang らのハッシュ関数攻撃におけるメッセージ語変更法に受け継がれている。

[3]Davis-Meyer 構造では、この暗号化関数において、入力差分＝出力差分となる高確率の差分パスがあれば、それを使って、ハッシュ連鎖変数の差分＝0、即ち、衝突発見となる可能性が高い

[4]SHA'-256,SHA"256 として、それぞれ 16 ビット、8 ビットパターンの繰り返しも述べている

第4章　Hawkesらの解析

Hawkes らは、Gilbert の局部衝突の解析に、符号付き算術差分も合わせて考察し、適切な拡張メッセージ語差分決定法を提案し、それを使うことにより、Gilbert の局部衝突確率 2^{-66} が、局部衝突の開始時点の内部状態が未知の場合 2^{-39} に、既知の場合は、2^{-9} に上昇することを論文 [4] で示している。また、彼らの手法を使えば、この局部衝突は、第二原像発見攻撃として使用可能であり、攻撃の計算量は、誕生日攻撃の $2^{n/2}$ と比較するのではなく、2^n と比較すべきと主張している。

Hawkes らの論文は、入力される拡張メッセージ語を攻撃者が自由に変更出来るという前提の上で、Gilbert の９ステップ局部衝突の確率を向上させる事を目指したものであり、SHA-2 の衝突発見を議論しているのでは無いが、その基本的考え方を以下に紹介する[1]。詳しくは、論文 [4] を参照されたい。

4.1　Gilbertらの局部衝突

第 3.2 節の 9 ステップ局部衝突を再掲する。ここでは、ハミング重みではなく、差分ベクトルとして表示する。この局部衝突では、拡張メッセージ語 W_i に 1 ビット差分 $\alpha 1$ を入れて引き起こされた内部状態の差分の伝播を 4.1 のように制御する事で、９ステップ局部衝突を構成している。表中の差分は、以下である。

$$\begin{aligned} \alpha 1 &= \Delta W_i \\ \beta 3 &= \Sigma_0(\Delta W_i) \\ \gamma 3 &= \Sigma_1(\Delta W_i) \\ \epsilon 9 &= \Sigma_1(\Sigma_0(\Delta W_i)) \end{aligned}$$

この表において、Gilbert らは、Ch,Maj 関数の出力差分を０、圧縮関数の中の算術和が、XOR 和と同じ振る舞いをすると期待して、拡張メッセージ差分系列 ΔW_{i+j} を、表の最右列にとれば、内部変数差分は表のように振る舞い、局部衝突が実現できると考えている。Gilbert らは、その確率を Ch,Maj 関数に関し 2^{-18}, 算術和に関し 2^{-48}、全体で 2^{-66} と評価している。Hawkes らは、算術差分も併用して考えることにより、目標となる内部変数差分が、この表のように伝播するように W_{i+j} の算術差分を選び、確率を向上させている。

4.2　Gilbert評価に対するHawkesの考え

Gilbert らは、SHA-2 のメッセージ拡張関数に関し、考察を行い、9 ステップ局部衝突の合成で得られる衝突の発見確率は、大きくても、SHA-256 の場合 2^{-132}、(SHA-384/512 の場合 2^{-264}) であり、2^{132} (2^{264}) の攻撃計算量が掛かるとし、誕生日攻撃の計算量 2^{128}(2^{192} 又は 2^{256}) に比べ、SHA-2 のこの攻撃に対する暗号学的安全性を確認している。

[1]本章は、Hawkes らの論文を要約では無く、本報告書作成者の責任で、彼らの基本的考え方としてまとめている

表 4.1: 9 ステップ局部衝突

j	Δa	Δb	Δc	Δd	Δe	Δf	Δg	Δh	ΔW_{i+j}
0	-	-	-	-	-	-	-	-	$\alpha 1$
1	$\alpha 1$	-	-	-	$\alpha 1$	-	-	-	$\beta 3 \oplus \gamma 3$
2	-	$\alpha 1$	-	-	$\beta 3$	$\alpha 1$	-	-	$\epsilon 9$
3	-	-	$\alpha 1$	-	-	$\beta 3$	$\alpha 1$	-	-
4	-	-	-	$\alpha 1$	-	-	$\beta 3$	$\alpha 1$	$\alpha 1$
5	-	-	-	-	$\alpha 1$	-	-	$\beta 3$	$\beta 3 \oplus \gamma 3$
6	-	-	-	-	-	$\alpha 1$	-	-	-
7	-	-	-	-	-	-	$\alpha 1$	-	-
8	-	-	-	-	-	-	-	-	$\alpha 1$

Hawkes らは、衝突発見確率が $1/N$ ならば、第二原像発見攻撃のように、対象となるメッセージ M が固定されていても、相手候補である別のメッセージ M' を N 個選べば第二原像が発見できる事が期待されるので、比較する計算量は、第二原像発見攻撃の計算量,SHA-256 で 2^{256}（SHA-384 で 2^{384},SHA-512 で 2^{512}) であり、SHA-2 の Gilbert 型攻撃に対する暗号学的安全性が確認されてはいないと主張している。

本報告書作成者は、以下のように考える。Gilbert の攻撃では、差分が固定されており、第二原像発見攻撃で一つめのメッセージ M が与えられたとき、二つめのメッセージ候補 M' は一つしか選べない。従って、第二原像が発見できる確率は無視出来るほど小さいので、第二原像攻撃と考えるのは無理がある。Hawkes らの 9 ステップ局部衝突は、ハッシュ関数の内部状態に依存した拡張メッセージ差分を使うものである。内部状態を推測した上で複数の差分候補を見かけ上、使っているが、第二原像攻撃の立場で、一つめのメッセージ M が与えられたとき、内部状態は一意に決まるため、複数の差分候補 (即ち、複数の第二原像候補）が選べる訳ではない。従って、Hawkes の提案した攻撃を第二原像発見攻撃と比較する事は適切でないと考える。

4.3　算術差分併用方式による９ステップ局部衝突

4.3.1　算術差分と XOR 差分

二つの w 次元 2 元ベクトル X,X' を 2 進数として見た物を、x,x' とする。0 から $2^w - 1$ までの数字である。算術差分 δX を

$$\delta X = x' - x(\mod 2^w) \tag{4.1}$$

で表す。ここで、w 次元ベクトル X の第 i ビットを $X[i]$ と表すことにする。

$$x = \sum_{i=0}^{w-1} X[i]2^i \tag{4.2}$$

である。このとき、次の補題が成り立つ。

補題:

XOR 差分 $\Delta X = \lambda$ が与えられたとき、$\lambda[i] = 1$ となるビット位置の X の値のみを使い、δx は、

次式で表される。

$$\delta X = x' - x = \sum_{i:\lambda[i]=1} (1 - 2X[i])2^i \tag{4.3}$$

$\because$) 次の 2 つの式を $\delta X = x' - x$ に代入すれば、式 (4.3)。

$$\begin{aligned} x &= \sum_{i:\lambda[i]=1} X[i]2^i + \sum_{i:\lambda[i]=0} X[i]2^i \\ x' &= \sum_{i:\lambda[i]=1} X'[i]2^i + \sum_{i:\lambda[i]=0} X'[i]2^i \\ &= \sum_{i:\lambda[i]=1} (1 - 2X[i])2^i + \sum_{i:\lambda[i]=0} X[i]2^i \end{aligned}$$

□

ここで、式 (4.1) が 2^w を法とする計算であることに注意するならば、$2X[w-1]2^{w-1} = 0(\mod 2^w)$ であり、最上位ビット (MSB) の値 $X[w-1]$ は知らなくても、算術差分 δX が、式 (4.3) から計算できることに注意。

4.3.2　9 ステップ局部衝突の拡張メッセージ差分 δW_{i+j}

表 4.1 の 9 ステップ局部衝突で、期待する内部状態の差分伝播を、引き起こす為に、拡張メッセージ W_{i+j} の算術差分が使用される。SHA-2 の内部状態は、式 (2.16) によって更新される。変数 W_{i+j} の変化で、直接に影響を受けるのは、変数 e 及び a である。式に算術差分を適用して考え、Hawkes らは次の基本方針で、入力すべき拡張メッセージの算術差分 δW_{i+j} を決定している。

性質 1. 次式

$$\delta W_t = -(\delta H_t + \delta\Sigma_1(e_t) + \delta(Ch(e_t, f_t, g_t)) \tag{4.4}$$

を満たすように、算術差分 δW_t を選べば、$\delta T_1 = \Delta T_1 = 0$ である。このとき、

$$\begin{aligned} \delta e_{t+1} &= \delta d_t \\ \Delta e_{t+1} &= \Delta d_t \end{aligned} \tag{4.5}$$

と成る。

性質 2. 次式

$$\delta W_t = -(\delta H_t + \delta\Sigma_1(e_t) + \delta Ch(e_t, f_t, g_t) + \delta\Sigma_0(a_t) + \delta Maj(a_t, b_t, c_t)) \tag{4.6}$$

を満たすように、算術差分 δW_t を選べば、

$$\begin{aligned} \delta a_{t+1} &= \delta(T_1 + T_2) = 0 \\ \Delta a_{t+1} &= \Delta(T_1 + T_2) = 0 \end{aligned} \tag{4.7}$$

となる。

性質 3. Maj 関数の 3 入力 a_t, b_t, c_t の 1 つのみに差分が入っても、そのビット位置で、残りの 2 つの変数が等しければ、$\Delta Maj = 0$ となる。

性質 4. Ch 関数の出力差分は、３入力 e_t, f_t, g_t の何れかのビットに差分が入っているビット位置に発生しうる。

彼らは、表 4.1 の９ステップ局部衝突のステップ番号 $j = 1$ に性質１を、$j = 2, \cdots, 8$ に性質２を適用している。Maj 関数については、性質３を適用し全てのステップで,$\Delta Maj = 0$ としている。性質１、２の適用を合わせ考えれば、結果として、$\delta a_j = 0 (j = 2, \cdots, 8)$ を保証していることになる。$\Delta Maj = 0$ が成立するために、内部変数 a_t, b_t, c_t の特定ビットに条件が必要となる。期待する局部衝突を実現するために、条件が付く内部変数を Hawkes らは、**仮定ビット**と呼んでいる。Ch 関数について、$j = 1$ ステップでは、内部変数 f_1, g_1 の特定ビットに条件を付け、$\Delta Ch_1 = 0$ としているが、他のステップでは、Ch 関数の出力に差分を許し、その算術差分 δCh を評価し、性質１により拡張メッセージ差分 δW でそれを打ち消し、差分の拡散を防いでいる。補題により、算術差分を求める時、ＸＯＲ差分 ΔX が非零のビットについては、その値 $X[i]$ が必要である。Hawkes らは、その推定すべきビットを、**推定ビット**と呼んでいる。内部状態が不明の場合、攻撃者は、推定ビットを総当たりで推定し、算術差分候補 δW_t を試し、局部衝突を見つける。しかし、仮定ビットが条件を満たしているメッセージが１つ見つかったならば、そのメッセージに対して、内部状態を攻撃者は知っている。この場合は、推定ビットを総当たりする必要はない。これが、**仮定ビット**と**推定ビット**の性格の違いである。

このような方針で Hawkes らは、9 ステップ局部衝突を引き起こす拡張メッセージ差分 δW_{i+j} を求めている。結果を表 4.2 示す。表中ではステップ番号 i を省略しているが、任意のステップ番号 i から $i + 8$ にかけて、この局部衝突を実現出来る。各行はステップ番号を表す。内部変数 $a, \cdots, h$ に関し、上段は XOR 差分、下段は算術差分を表す。δW_j の列は、そのステップで入力すべき拡大メッセージ算術差分を表す。最後の２列は、Ass で仮定ビット、Gue で推測ビットを定める内部変数 XOR 差分を表す。例えば、１行目の Ass 欄の $\hat{\alpha 1}$ は、$\hat{\alpha 1}[i] = 1$ となっているビット位置 i に関し、内部変数に関する仮定条件があり、Gue 欄の $\hat{\alpha 1}$ は、$\hat{\alpha 1}[i] = 1$ となっているビット位置 i に関し、別の内部変数ビットを推測する必要があることを示す。$\hat{X}$ は、２進数ベクトル X から最上位ビットをのぞいたものを表し、$|X|$ は、X のハミング重みを表す。

この局部衝突として、1 ビットの擾乱訂正パターンを選べば、$|\alpha 1| = |\hat{\alpha 1}| = 1, |\beta 3| = |\hat{\beta 3}| = 3$, $|\gamma 3| = |\hat{\gamma 3}| = 3, |\epsilon 9| = |\hat{\epsilon 9} = 9$ であり、仮定ビット数 u, 推定ビット数 v は、それぞれ

$$u = 4 \cdot 1 + 2 \cdot 1 + 3 + 3 = 12 \tag{4.8}$$

$$v = 5 \cdot 1 + 1 + 2 \cdot 3 + 3 + 2 \cdot 3 + 9 = 30 \tag{4.9}$$

となる。従って、合計で、$2^{u+v} = 2^{42}$ の計算量で、局部衝突を発見できる。さらに、擾乱パターンを最上位ビットに係わるようにすれば、(例えば $\alpha 1 = 0x80000000$ とすれば $|\hat{\alpha 1}| = 0$)、さらに計算量を減らすことが出来、2^{39} となる。

なお、拡張メッセージを攻撃者が自由に選ぶことができるという前提では、$2^u = 2^{12}$ の計算量で、仮定ビット条件を満たす拡張メッセージが得られたならば、そのメッセージに対して、ハッシュ関数の圧縮動作を適用して、内部変数の値を確認すれば、推定ビット数分 $2^v = 2^{30}$ の試行錯誤は必要なく、衝突発見の計算量は、2^{12} となる。

表 4.2: Hawkes の 9 ステップ局部衝突

j		a	b	c	d	e	f	g	h	δW_j	Ass	Gue
0	Δ	-	-	-	-	-	-	-	-	$\delta E_1 = \delta A_1 = \delta_0$	$\hat{\alpha}1$	$\hat{\alpha}1$
	δ	-	-	-	-	-	-	-	-			
1	Δ	$\alpha 1$	-	-	-	$\alpha 1$	-	-	-	$-(\delta\Sigma_1(e_1) + \delta\Sigma_0(a_1))$	$\alpha 1, \alpha 1$	$\hat{\alpha}1$
	δ	δ_0	-	-	-	δ_0	-	-	-	$= -(\delta_{1,1} + \delta_{1,2})$	$\hat{\beta}3$	$\hat{\gamma}3$
2	Δ	-	$\alpha 1$	-	-	$\beta 3$	$\alpha 1$	-	-	$-(\delta Ch_2 + \delta\Sigma_1(e_2))$	$\alpha 1$	$\alpha 1$
	δ	-	δ_0	-	-	$-\delta_{1,2}$	δ_0	-	-	$= -(\delta_{2,1} + \delta_{2,2})$	$\beta 3$	$\hat{\epsilon}9$
3	Δ	-	-	$\alpha 1$	-	-	$\beta 3$	$\alpha 1$	-	$-\delta Ch_3 = -\delta_3$	$\alpha 1$	$\alpha 1$
	δ	-	-	δ_0	-	-	$-\delta_{1,2}$	δ_0	-			$\beta 3$
4	Δ	-	-	-	$\alpha 1$	-	-	$\beta 3$	$\alpha 1$	$-(\delta h_4 + \delta Ch_4)$	$\hat{\alpha}1$	$\beta 3$
	δ	-	-	-	δ_0	-	-	$-\delta_{1,2}$	δ_0	$-(\delta_0 + \delta_4)$		
5	Δ	-	-	-	-	$\alpha 1$	-	-	$\beta 3$	$-(\delta h_5 + \delta Ch_5 + \delta\Sigma_1(e_5))$	-	$\alpha 1$
	δ	-	-	-	-	δ_0	-	-	$-\delta_{1,2}$	$-(-\delta_{1,2} + \delta_{5,1} + \delta_{5,2})$		$\hat{\gamma}3$
6	Δ	-	-	-	-	-	$\alpha 1$	-	-	$-\delta Ch_6 = -\delta_6$	-	$\alpha 1$
	δ	-	-	-	-	-	δ_0	-	-			
7	Δ	-	-	-	-	-	-	$\alpha 1$	-	$-\delta Ch_7 = -\delta_7$	-	$\alpha 1$
	δ	-	-	-	-	-	-	δ_0	-			
8	Δ	-	-	-	-	-	-	-	$\alpha 1$	$-\delta h_8 = -\delta 0$	-	-
	δ	-	-	-	-	-	-	-	δ_0			

仮定ビット数	$4\lvert\alpha 1\rvert + 2\lvert\hat{\alpha}1\rvert + \lvert\hat{\beta}3\rvert$
推定ビット数	$5\lvert\alpha 1\rvert + \lvert\hat{\alpha}1\rvert + 2\lvert\hat{\beta}3\rvert + \lvert\hat{\gamma}3\rvert + \lvert epsi\hat{l}on9\rvert$

第5章　Yoshidaらの解析

Yoshida らは、SHA-2 の全ての算術加算を XOR 加算で置き換えた SHA-2-XOR の安全性に関し議論をしている。SHA-2-XOR の圧縮関数部分を暗号化変換と考える SHACAL-2-XOR の差分特性を解析し, 確率 2^{-8} の 1 ステップ繰り返し差分パスを発見している。それを適用することで、15 ステップ SHA-2-XOR の擬似衝突が 2^{120} の計算量で発見できる事、同じく、31 ステップ繰り返すならば、差分特性確率 $2^{-248} > 2^{-256}$ であり、31 ステップ SHA-2-XOR は、ランダムハッシュ関数では無い事を示している。また、先頭 19 ステップの差分確率を 1 とするメッセージ変更法を示し、残り 15 ステップに、繰り返し差分特性を適用し、34 ステップ SHA-2-XOR の擬似衝突が、計算量 2^{120} で発見できると推定している。以下、その概要を述べる。

5.1　SHA-2-XOR と SHACAL-2-XOR の差分解析

5.1.1　１段繰り返し型特性差分

SHA-2 の圧縮関数のレジスタ状態 $(a_0, b_0, c_0, d_0, e_0, f_0, g_0, h_0)$ を入力平文、$(a_{64}, b_{64}, c_{64}, d_{64}, e_{64}, f_{64}, g_{64}, h_{64})$ を出力暗号文、拡張メッセージ語 $W_t (0 \leq t \leq 63)$ を段鍵と考えた暗号化変換を SHACAL-2 という。SHA-2-XOR,SHACAL-2-XOR は、それらの変換内部の算術加算を XOR 演算で置き換えたものである。以下、SHA-2-XOR の 1 段繰り返し差分特性[1]を考察し、それが満足するべき制約条件を示す。

ここでレジスタ a の t 段の値を a_t、差分を Δa_t と記す。1 段繰り返し差分では、段 t と $t+1$ での差分が同じになっていることから次式である。

$$\begin{array}{rclcrclcrcl} \Delta a_{t+1} & = & \Delta a_t, & \Delta b_{t+1} & = & \Delta b_t, & \Delta c_{t+1} & = & \Delta c_t, & \Delta d_{t+1} & = & \Delta d_t, \\ \Delta e_{t+1} & = & \Delta e_t, & \Delta f_{t+1} & = & \Delta f_t, & \Delta g_{t+1} & = & \Delta g_t, & \Delta h_{t+1} & = & \Delta h_t \end{array} \tag{5.1}$$

式 (2.16) のように、a_{t+1}, e_{t+1} 以外の 6 つのレジスタの差分は、段 t での一つのレジスタの値で決定される。段 t の条件式として整理すれば、上式の 6 つのレジスタの差分に関する条件として

$$\begin{array}{rclcrclcrcl} \Delta a_t & = & \Delta b_t & = & \Delta b_t & = & \Delta b_t, \\ \Delta e_t & = & \Delta e_t & = & \Delta g_t & = & \Delta g_t \end{array} \tag{5.2}$$

が得られる。残りの二つの制約 $\Delta a_{t+1} = \Delta a_t$ と $\Delta e_{t+1} = \Delta e_t$ に関し、考察する。段関数内で使われている関数の出力差分

$$\begin{array}{rcl} \Delta Ch & = & Ch(X,Y,Z) \oplus Ch(X',Y',Z') \\ \Delta Maj & = & Maj(X,Y,Z) \oplus Maj(X',Y',Z') \end{array} \tag{5.3}$$

を、入力差分 $\Delta X_t = X_t \oplus X'_l$ を使って書き換えれば

$$\begin{array}{rcl} \Delta Ch & = & ((Y \oplus Z) \wedge \Delta X) \oplus (X \wedge \Delta Y) \oplus (\bar{X} \wedge \Delta Z) \oplus (\Delta X \wedge \Delta Y) \oplus (\Delta X \wedge \Delta Z) \\ \Delta MJ & = & MJ(\Delta X, \Delta Y, \Delta Z) \oplus ((Y \oplus Z) \wedge \Delta X) \oplus ((Z \oplus X) \wedge \Delta Y) \oplus ((X \oplus Y) \wedge \Delta Z) \end{array} \tag{5.4}$$

[1]ブロック暗号における言い方に従い、ハッシュ関数の演算の 1 ステップを、この章では、1 段と表記する

ここで、Ch, Maj 関数の入力は、それぞれ、(e, f, g),(a, b, c) であり、一段繰り返し型差分特性において、式 (5.2) のように、入力差分が等しく $\Delta X = \Delta Y = \Delta Z$ である。このとき、

$$\begin{array}{rcl} \Delta Ch & = & (Y \oplus \bar{Z}) \wedge \Delta X \\ \Delta Maj & = & \Delta X \end{array} \tag{5.5}$$

となる。これは、Maj 関数は全ての入力差分が同じ場合、線形に振舞うという重要な特性を意味する。上式を使って、$\Delta T1_t, \Delta T2_t, \Delta a_{t+1}, \Delta e_{t+1}$ は、SHACAL-2-XOR の場合、以下となる。

$$\begin{array}{rcl} \Delta T1_t & = & \Delta e_t \oplus \Sigma_1(\Delta e_t) \oplus ((f_t \oplus \bar{g}_t) \wedge \Delta e_t) \\ \Delta T2_t & = & \Sigma_0(\Delta a_t) \oplus \Delta Maj(a_t, b_t, c_t) = \Sigma_0(\Delta a_t) \oplus \Delta a_t \\ \Delta a_{t+1} & = & \Delta T1_t \oplus \Delta T2_t \\ \Delta e_{t+1} & = & \Delta D_t \oplus \Delta T1_t = \Delta a_t \oplus \Delta T1_t \end{array} \tag{5.6}$$

従って、残った $\Delta a_{t+1} = \Delta a_t, \Delta e_{t+1} = \Delta e_t$ の制約条件は以下の二つの条件となる。

$$\begin{array}{rl} \Delta a_t & = \Delta T1_t \oplus \Delta T2_t \\ \Delta e_t & = \Delta a_t \oplus \Delta T1_t \end{array} \tag{5.7}$$

以下、レジスタ変数差分の段番号の指標 t を、$\Delta a_t = \Delta a$ などと省く。二つの条件は、以下である。

$$\begin{array}{rcl} \Delta a & = & \Delta T1_t \oplus \Delta e \\ \Delta e & = & \Delta T2_t \end{array} \tag{5.8}$$

式 (5.6) を代入し、条件式は、

$$\begin{array}{rcl} \Delta a & = & \Delta e \oplus \Sigma_1(\Delta e \oplus ((f \oplus \bar{g}) \wedge \Delta e) \oplus \Delta e \\ \Delta e & = & \Sigma_0(\Delta a) \oplus \Delta a \end{array} \tag{5.9}$$

ここで、Δa の条件式中の $f \oplus \bar{g}$ の値を, ランダムの値 X と考えることができ、この条件は次の条件となる。

$$\Delta a = \Sigma_1(\Delta e) \oplus (X \wedge \Delta e) \tag{5.10}$$

これらの条件を満たす 1 段繰り返し差分特性で高い確率を持つものを考える。SHACAL-2-XOR の内部状態更新関数の構造より、差分 $\Delta a, \Delta e$ が二つの条件式 (5.9) を満たせば、他の 6 つの条件式 (5.2) も満たされることに注意する。二つの条件式において、確率が係わるのが、Δa の条件である。そこでは、$\Delta e[j]$ が 1 でビット位置 j におけるランダム変数 $X[j]$ の値により、成立／不成立が変わってくる。従って、確率の一番高い 1 段繰り返し特性差分は Δe のハミング重みが最も小さいものとなる。以上の議論を定理としてまとめる。**定理:**
SHA-2-XOR において、入力差分 $(\Delta a, \Delta b, \Delta b, \Delta b, \Delta b, \Delta b, \Delta b, \Delta b)$ の差分特性は、ある 32 ビットの値 X に対し、入力差分 $\Delta a, \Delta e$ が以下を満たす場合に一段繰り返し特性差分となる。

$$\begin{array}{rl} \Delta a & = \Sigma_1(\Delta e) \oplus (X \wedge \Delta e) \\ \Delta e & = \Sigma_0(\Delta a) \oplus \Delta a \end{array} \tag{5.11}$$

この条件が満たされるとき、他の差分は $\Delta a, \Delta e$ によって以下の様に定まる。

$$\begin{array}{rcl} \Delta b & = & \Delta a, \Delta c \ = \ \Delta a, \Delta d \ = \ \Delta a, \\ \Delta f & = & \Delta e, \Delta g \ = \ \Delta e, \Delta g \ = \ \Delta e \end{array} \tag{5.12}$$

また、Δe のハミング重みがもっとも小さいものが一番よい特性差分確率を与える。

表 5.1: １段繰り返し特性差分（確率＝ 2^{-8}）

$\Delta a = \Delta b = \Delta c = \Delta d$	$\Delta e = \Delta f = \Delta g = \Delta h$
$3b3b3b3b$	$c0c0c0c0$
67676767	18181818
76767676	81818181
$9d9d9d9d$	60606060
$b3b3b3b3$	$0c0c0c0c$
$cecececе$	30303030
$d9d9d9d9$	06060606
$ecececec$	03030303

5.1.2　１段繰り返し型特性差分の探索

前節の定理の二つ目の条件を一つ目の条件に代入すれば、次の式となる。

$$\Delta a = \Sigma_1(\Sigma_0(\Delta a) \oplus \Delta a) \oplus (X \wedge (\Sigma_0(\Delta a) \oplus \Delta a)) \tag{5.13}$$

$I = \Delta a \oplus \Sigma_1(\Sigma_0(\Delta a) \oplus \Delta a)$,$R = \Sigma_0(\Delta a) \oplus \Delta a$ とおけば、方程式、$I = X \wedge R$ である。この式をビット単位にみれば、i ビット目の式で解 $X[i]$ が存在するのは、$R[i] = 1$ 又は、$R[i] = I[i] = 0$ の時である。Δa を 2^{32} 回総当たりし、その差分に対し、32 ビット分の方程式 $(0 \leq i \leq 32)$ が全て解を持つならば、その差分 Δa は１段繰り返し特性差分を構成しうる差分である。

このようにして、１段繰り返し特性差分を調べ上げ、その中で、Δe のハミング重みが最も小さいものを、表 5.1 に示す。

表の 8 つの繰り返しパターンは、同じパターンが循環したものとなっている。

この特性差分の確率を実験的に確認すれば、$\Delta a = b3b3b3b3$,$\Delta e = 0c0c0c0c$ の場合、$259/2^{16}$ であり、ほぼ、2^{-8} である。この確率を下げているのは、Ch 関数であり、その入力差分 $0c0c0c0c$ は、確率 2^{-8} で出力差分 08080808 を与える。これは、以下のビットごとの Ch の差分特性を使って計算される。

$$\Delta Ch(0,0,0) = 0$$
$$\Delta Ch(1,1,1) = 0/1 \quad \textit{with probability } 1/2$$

Yoshida らは、同様の手法で、２段繰り返し型の特性差分の制約条件式を導出し、２段以下の繰り返し型特性差分の範囲で最良のものは、表 5.1 であるとしている。

5.1.3　繰り返し特性差分を使った SHA-2-XOR の擬似衝突攻撃

前節の繰り返し差分特性を利用して SHA-2-XOR に対して、次の二種類の攻撃が考えられる。

定義より、擬似衝突を見つけるため、攻撃者は拡張メッセージと内部レジスタに差分を注入出来る。SHA-2-XOR が理想のハッシュ関数ならば擬似衝突を見つけ計算量は 2^{128} 演算であるが、ここで見つけた１段繰り返し特性差分を 15 段つなぐならば、特性差分確率が 2^{-120} となり、15 段の SHA-2-XOR に対して 2^{120} 演算で擬似衝突を見つける攻撃へとつながる。

理想的ハッシュ関数の場合、もし入力差分と出力差分が固定されているとき、入力差分＝出力差分となる確率は 2^{-256} である。しかしながら、31 段までの SHA-2-XOR では確率 2^{-248} であるということは、このハッシュ関数の 31 段まではランダム関数として振舞わないということを意味する。

さらに、SHACAL-2-XOR に対する差分攻撃としてみれば、この 31 段の特性差分を使って 32 段の SHACAL-2-XOR 暗号が攻撃できることを意味する。詳細は、論文 [9] を参照されたい。

5.1.4　擬似衝突攻撃の改良

前節で述べた、一段繰り返し差分特性を使い、適切なメッセージ W_t $(0 \le t \le 15)$ および内部変数初期値 e_0, g_0, h_0 を選ぶことにより、メッセージ拡張関数を含めた SHA-2-XOR について、19 段まで特性差分が確率 1 で伝わることを示す[2]。

第 5.1.2 節の条件式 (5.1.2) は、ビット単位に見て 32 本の $I[i] = X[i] \wedge R[i]$ 型の方程式である。ある $X[i] = f[i] \oplus g[\bar{i}] = f[i] \oplus g[i] \oplus 1$ に対し、この式を満足する解として、1 段繰り返し差分特性が表 5.1 のように与えられている。この式で $R[i] = 1$ の場合、$X[i]$ は制約を受けるが、$R[i] = I[i] = 0$ の場合、$X[i]$ はいかなる値でもよい。この X に対する、制約条件を、差分 $\Delta a = 3b3b3b3b$,$\Delta e = c0c0c0c0$ に対して求めれば以下である。

$$\begin{array}{lllllll} R & = & \Delta e & = & c0c0c0c0 & = & 11000000 \cdots 11000000 \\ I & = & \Delta a \oplus \Sigma_1(\Delta e) & = & 40404040 & = & 01000000 \cdots 01000000 \\ X & & & & & = & 01xxxxxx \cdots 01xxxxxx \end{array} \tag{5.14}$$

ここで x は任意の値を示す。従って、$f \oplus g = 10xxxxxx10xxxxxx10xxxxxx10xxxxxx$ である。$L = 80808080$ として、この X に関する条件は、ビット位置の組を $J = \{0, 1, 8, 9, 16, 17, 24, 25\}$ として次式で表せる。

$$f_t[i] \oplus g_t[i] = L[i], for i \in J \tag{5.15}$$

さらに、内部状態更新関数より、$g_{t+2} = f_{t+1} = e_t$ であるので、この条件は

$$\begin{array}{rcll} f_0[i] \oplus g_0[i] & = & L[i] \quad , & for(i \in J) \\ e_0[i] \oplus f_0[i] & = & L[i] \quad , & for(i \in J) \\ e_t[i] \oplus e_{t+1}[i] & = & L[i] \quad , & for(i \in J, t = 0, 1, \cdots, 16) \end{array} \tag{5.16}$$

とまとめられる。この条件を満足するように、$e_0, g_0, h_0, W_0, \cdots, W_{15}$ を定めるアルゴリズムを示せば、以下である。

Step 1. 初期値 $a_0, b_0, c_0, d_0, e_0, f_0, g_0, h_0$ をランダムに選ぶ

Step 2. メッセージ $W_0, W_1, \cdots, W_{15}$ をランダムに選ぶ

Step 3. $g_0[i]$ と $e_0[i]$ の 8 ビットを $(f_0 \oplus L)[i], (i \in J)$ に変更

Step 4. For t=0 to 15 do:
$\alpha = e_{t+1} \oplus e_t \oplus L$
$W_t[i]$ の 8 ビットを $\alpha[i], (i \in J)$ に変更
t 段目の内部状態更新関数を W_t で計算

Step 5. $W_0^{old} = W_0$

[2] ここでは、論文 [9] の考え方を用いた別の数値例で示す。

Step 6. $\beta = d_{16} \oplus h_{16} \oplus \Sigma_1(e_{16}) \oplus Ch(e_{16}, f_{16}, g_{16}) \oplus K_{16} \oplus L \oplus e_{16}$

Step 7. $W_{16} = \sigma_1(W_{14}) \oplus W_9 \oplus \sigma_0(W_1) \oplus W_0$

Step 8. $W_{16}[i]$ の 8 ビットを $\beta[i], (i \in J)$ に変更

Step 9. $W_0[i]$ の 8 ビットを $(W_{16} \oplus \sigma_1(W_{14}) \oplus W_9 \oplus \sigma_0(W_1))[i], (i \in J)$ に変更

Step 10. $h_0[i]$ の 8 ビットを $(h_0 \oplus W_0 \oplus W_0^{old})[i], (i \in J)$ に変更

このように変更することで、$f_t \oplus g_t, (0 \leq t \leq 18)$ が所定の値となり、19 段まで、特性差分が確率 1 で伝わる。

このアルゴリズムは、$e_0[i], g_0[i], h_0[i], W_0[i], W_1[i], \cdots, W_{15}[i], (j \in J)$ の 152 ビットの入力ビット $(= 19 \times 8$ ビット$)$ を修正する。擬似衝突攻撃で、攻撃者が選べるビット数は $256 + 512 = 768$ ビットであり、残りの 616 ビットで、継続する 15 段の特性差分パスを考えれば、全体で、34 段の SHA-2-XOR に対し、2^{120} の計算量の擬似衝突攻撃となる。

Yoshida らは、23 段の SHA-2-XOR に対し、計算機実験により求めた擬似衝突データを示している。さらに、彼らの求めた 1 段繰り返し特性差分のハミング重みが大きいことから、実際の SHA-2 にこの特性差分をそのまま適用してもが高い確率となる可能性は小さいとしている。

第6章　第1回CRYPTOGRAPHIC WORKSHOPにおける報告及びその他

2005年10月にNISTの第1回ハッシュワークショップが開かれた。発表論文の中で、SHA-2の解析を行っているものが1件ある。

Leeらは、メッセージ変更法による、衝突確率の向上法が存在すると仮定して、SHA-256の第24から55ステップを調査し、衝突には、15個以上の局部衝突の重ね合わせが必要である事を報告した[5]。これを、Hawkesらの局部衝突確率2^{-39}と合わせると、衝突発見の確率は、2^{-585}以下であるとしている。新しいハッシュ関数DHA-256の提案に主眼がおかれている為、詳細は不明である。彼らは、この確率は、思いの外大きいとしているが、SHA-256の安全性を脅かす結果ではない。

Pramstallerらは、予備的調査として、SHA-256のメッセージ拡張関数の効果を調べている[8]。1ビットのメッセージ差分は、SHA-1においてハミング重み107の拡張メッセージ差分を引き起こすのに対し、SHA-256の場合は、メッセージ拡張関数内の算術加算をXORで置き換えて簡略化しても、ハミング重み467の差分を引き起こし、差分拡散性に優れている。さらに、この簡略化メッセージ拡張関数の出力を線形符号と考え、確率的最小重み探索法を使って拡張メッセージ差分の最小ハミング重みが42ステップまでで35であることを求め、それを使って、フルステップで最小ハミング重み356と推定している。

これ以外、サーベイ論文として、SHA-2の解析に触れた論文[7],[10]や、実用的側面をふまえ、次期のハッシュ関数選択で、どのような戦略をとるべきかの議論のたたき台として、SHA-2に触れた論文[2]などがある。

以下、SHA-2の解析として、Leeら及びPramstallerらの論文の概要を紹介する。

6.1　Leeの解析

Leeらは、Wangらの攻撃を２種類のグループに分類し、彼らの提案するハッシュ関数DHAの安全性評価の根拠としている。ここでは、SHA-2に関する評価のみを紹介する。

Wangらの攻撃対象の１つのグループ（ Group I ）は、メッセージ語を並び替えて、圧縮関数に入力するものであり、MD4、MD5、RIPEMD、HAVALなどである。他のグループ（ Group II ）は、メッセージ語をそのまま使うのではなく、再帰関係式で定められた拡張関数の出力を圧縮関数に入力するものであり、SHA-0,SHA-1などである。SHA-2もGroup IIに属する。

Group IIのハッシュ関数に対する攻撃では、第１章で述べた、３つの戦略がとられている。このうちの戦略１を再度、述べれば以下であり、二つのステップからなる。

1. 効果的な局部衝突を発見する。

2. 局部衝突の合成として、メッセージ拡張関数通過後の語に対し成り立つ効果的な大域的な衝突（ハッシュ関数の衝突）を発見する。

SHA-2 に関しては、Gilbert らの９ステップ局部衝突が最良の局部衝突確率を与えるものとして知られている。その確率は、圧縮関数内の各ステップ関数の解析で得られ、Gilbert らの評価で 2^{-66}、Hawkes らの拡張メッセージ差分修正法では、最良で 2^{-39} である。２つめの大域的衝突の発見は、まだ成されていないが、SHA-256 の安全性を議論する場合大域的衝突を構成する局部衝突の個数（又は、その下限）を調べる事により、安全性の下界が評価ができる。この評価に於いては、メッセージ拡張関数のみを解析すれば良い。

Lee らは、Wang らの SHA-1 及び SHA-0 攻撃からの類推で、SHA-256 に関しても

1. ステップ１から 23 までは、メッセージ語変更により確率を１にできる
2. ステップ 56 から 63 までは、準衝突を使う解析法で無視できる

と推測し、ステップ 24 から 55 までの 32 ステップを結ぶのに必要な局部衝突の最小個数を disterbance vector の手法で求めている。

連続する 16 個の拡張メッセージ語を仮定し、メッセージ拡張関数を使えば、全ての拡張メッセージ語が定まるので、Lee らは、連続する拡張メッセージに、1 ビットから 3 ビットの擾乱（局部衝突の起点）を与えるパターンを総当たりし、必要な局部衝突の最小個数を求めている。結果は、最小値が 15 であり、Hawkes らの手法が適用できると仮定するならば、SHA-256 の衝突発見確率は、2^{-585} 以下であるとしている。彼らは、この確率は、思いの外大きいとしているが、SHA-256 の安全性を脅かす結果ではない。かれらの論文は、新しいハッシュ関数 DHA-256 の提案に主眼がおかれている為、解析の条件等に関し、本報告書作成者の立場からは、不明な点[1]が見られるが、彼らの結果を論文よりそのまま引用する。(表 6.1, 表 6.2) 表 6.1 は、LEE らの提案する DHA-256 と SHA-256 の安全性を（彼らの評価）で比較したものである。DHA-256 の紹介は、本報告書の趣旨をはずれるので省略する。表 6.2 において、'0' が１ビット差分を入力する語であり、局部衝突の開始位置である。

表 6.1: SHA-256 と DHA-256 の衝突発見確率

	SHA-256	DHA-256
Probability of Inner Collision Pattern	2^{-39}	2^{64}
Hamming Weight of Disterbance Vector (at Step.24-55)	15	63
Total Probability	2^{-585}	2^{-4032}

6.2　Pramstaller らの解析

Pramstaller らのグループは、SHA-256 のメッセージ拡張関数の効果に関し、論文 [6] および [8] で 2 つの方法で検討している。以下、その概要を紹介する。

[1]例えばメッセージ拡張関数の中の算術演算を XOR で近似したとの記述はないが、結果からは、近似が行われていると推測される。

表 6.2: 32 ステップ disturbance vector

Step	
24	//////////////////////////////
25	//////////////////////////////
26	//////////////////////////////
27	//////////////////////////////
28	////////////00////////////////
29	//////////////////////////////
30	//////////////////////////////
31	//////////////////////////////
32	//////////////////////////////
33	//////////////////////////////
34	//////////////////////////////
35	//////////////////////////////
36	///////0/////////0//////////
37	////////////00////////////////
38	//////////////////////////////
39	//////////////////////////////
40	//////////////////////////////
41	//////////////////////////////
42	//////////////////////////////
43	//////////////////////////////
44	//////////////////////////////
45	//////////////////////////////
46	//////////////////////////////
47	//////////////////////////////
48	//////////////////////////////
49	//////////////////////////////
50	//////////////////////////////
51	0//////////////////////////////
52	//////////////////////////////
53	////////////00////////////////
54	//////////////////////////////
55	///00////////////////////0000//

6.2.1　算術加算に対して線形な変形 SHA-256

彼らは、局部衝突の合成で、衝突を発見する戦略の有効性を確認するために、ADD-linear SHA-256 を想定している。その変形ハッシュ関数では、SHA-256 の関数 $\sigma_0, \sigma_1, \Sigma_0, \Sigma_1$ を恒等変換で置き換え、Maj, Ch 関数は、2^{32} を法とする算術加算

$$Maj(x, y, z) = Ch(x, y, z) = x + y + z \tag{6.1}$$

で置き換ている。このハッシュ関数では、全ての演算が、2^{32} を法とする算術加算の下で、線形となっている[2]。i 番目の拡張メッセージ W_i に、算術差分 $\delta_i = W'_i - W_i$ を擾乱として入れ、それに引き続く、8 個の拡張メッセージに、その擾乱を打ち消す差分を入れることにより、容易に、9 ステップの局部衝突が構成できる。この局部衝突は、次の長さ 9 の差分系列である。

$$\{\delta_i, -4\delta_i, 2\delta_i, 2\delta_i, 4\delta_i, 2\delta_i, \delta_i, 0, -\delta_i\} \tag{6.2}$$

このハッシュ関数において、メッセージ拡張関数も ADD-linear であり、メッセージから、拡張メッセージ系列への変換 $E : \mathbb{Z}_{2^{32}}^{16} \to \mathbb{Z}_{2^{32}}^{64}$ は、次の 64×16 行列

$$E = \begin{bmatrix} I_{16} \\ A \\ A^2 \\ A\ 3 \end{bmatrix} \tag{6.3}$$

で与えられる。ここで、A はメッセージ拡張関数の再帰関係式 (2.13) を表す 16×16 の行列である。従って、求める問題は、式 (6.2.1) において、$i = 0, 1, \cdots, 55$ と置いて得られる 56 本の $\mathbb{Z}_{2^{32}}^{64}$ のベクトルの線形結合で、且つ、上式の変換 E で生成される物を探す事になる。Pramstaller らは、拡張メッセージ差分の擾乱開始位置として $i = -8, -7, \cdots, -1$ 及び $i = 56, 57, \cdots, 63$ は許されないことから、条件式を導出し、局部衝突の合成を与えるメッセージ差分を導いている。そのようなメッセージ差分の中で、拡張メッセージ系列を 2 元ベクトルとして表したときのハミング重みが最小の物として、各拡張メッセージ語の最上位ビットにのみ差分を持つパターンを示し、そのハミング重みが 27 であるとしている。拡張メッセージ語 W_i の最上位ビット差分の系列を示せば、

$$10^8 1^2 0101^3 01^3 0^2 1^2 010^2 1^2 0^6 1^3 0^2 101^5 01^3 0^{11} \tag{6.4}$$

である。ここで、0^8 の表記は、0 が 8 語続くことを表す。他も同様である。Maj, Ch 関数のみ本来のものに戻した変形 SHA-256 において、この擾乱パターン (Disturbance pattern) を用いた時の、衝突発見確率を求めるならば、2^{-84} とのことであり、かなり大きな値である。

彼らが、述べているように、この解析は、SHA-256 の強度に関しては、何も言及していないと考えるが、$\sigma_0, \sigma_1, \Sigma_0, \Sigma_1$ 関数が、メッセージ差分の拡散に重要な役割を担っている事が伺える。

6.2.2　XOR 演算に対して線形な変形 SHA-256

Pramstaller らは、SHA-1 と SHA-256 のメッセージ拡張関数の差分拡散効果を調べる為に初歩的調査を行っている。メッセージに入れた 1 ビット差分が影響する拡張メッセージビット数の最小値を、SHA-1,SHA-256, 変形 SHA-1, 変形 SHA-256 の完全版 (SHA-1 は 80 ステップ、SHA-256 は

表 6.3: 変形したメッセージ拡張における 1 ビット差分が影響を与えるビット数の比較

	orig.SHA-1	mod.SHA-1	mod.SHA-256	orig.SHA-256
min(40step)	18	18	110	137
max(40step)	30	41	297	307
min(full)	107	247	467	507
max(full)	174	354	694	709

64 ステップ) と 40 ステップまで減らした縮小版に対して調べている。結果を表に示す。ここで、変形 SHA-1 のメッセージ拡張は、すべての XOR を mod2^{32} の加算に置き換えたものであり、変形 SHA-256 のメッセージ拡張は、すべての加算を XOR に置き換えたものである。mod 加算の導入と、単なるビットシフトを、σ_0 と σ_1 関数に置き換える事によりメッセージ拡張の中で影響されるビット数が、著しく増えている事が解る。

mod 加算の非線形性のため、SHA-256 のメッセージ拡張は、２元線形符号で記述することはできない。しかし、もし mod 加算を XOR に置き換えたならば、線形符号として考えることが出来る。もし、N ステップまでの SHA-256 を考えると、$512 \times 32N$ の生成行列 G を持つ２元線形符号と考えられる。

XOR で線形化することにより、二つの拡張語系列のすべての起こりえる差分は、この符号での符号語となっている。それゆえ、XOR で線形化した SHA-256 のメッセージ拡張における、差分系列の最小重みは、符号理論における確率的最小重み探索アルゴリズムで、求めることが出来る。結果を図 6.1 に論文 [8] より再掲する。見つかった最小重みを図では、実線で示してある。42 ステップまでは、小さい重みが見つかっている。40 ステップまでを考えると、重み 26 は表 6.3 の最小重み 110 と比較して低くなっている。　42 ステップ以降は、確率的アルゴリズムの動作時間が長く、適切な重みが得らていない。42 ステップ以降の突然の上昇は、SHA-256 のメッセージ拡張の本来の特性ではなく、限られた動作時間の影響と考える。

42 ステップ以降の実際の小さい重みを推測するために、以下の方法をとった。42 ステップまでの小さい重みの符号語を得た後、それをメッセージ拡張関数により 64 ステップ分のワードに拡張する。このように得られた符号語は、図の中で点線で示されている。スタートの点は、42 ステップまでの重み 35 のワードである。それを 64 ステップまで拡張することにより、重み 356 が与えられる。これは、表 6.3 で与えられる最小重みの 467 よりかなり小さい。

ここでは、本来の SHA-256 のメッセージ拡張関数の中の mod 演算を、ビットごとの XOR 演算で近似しているので、これが、本来関数に対して、必ずしも有効な差分系列である保証はない。また、得られたベクトルは、Chabaud と Joux が SHA-0 に行ったような衝突を生み出す disturbance パターンとして直接使うことはできない。理由は、拡張より前の非ゼロワードによって生じる truncated 局部衝突があるためである。

[2] このハッシュ関数は、算術差分に関し線形であり、高々17 回のハッシュ関数計算で、衝突を発見できるが、彼らの主張を説明するために、導入した変形ハッシュ関数と考えられる

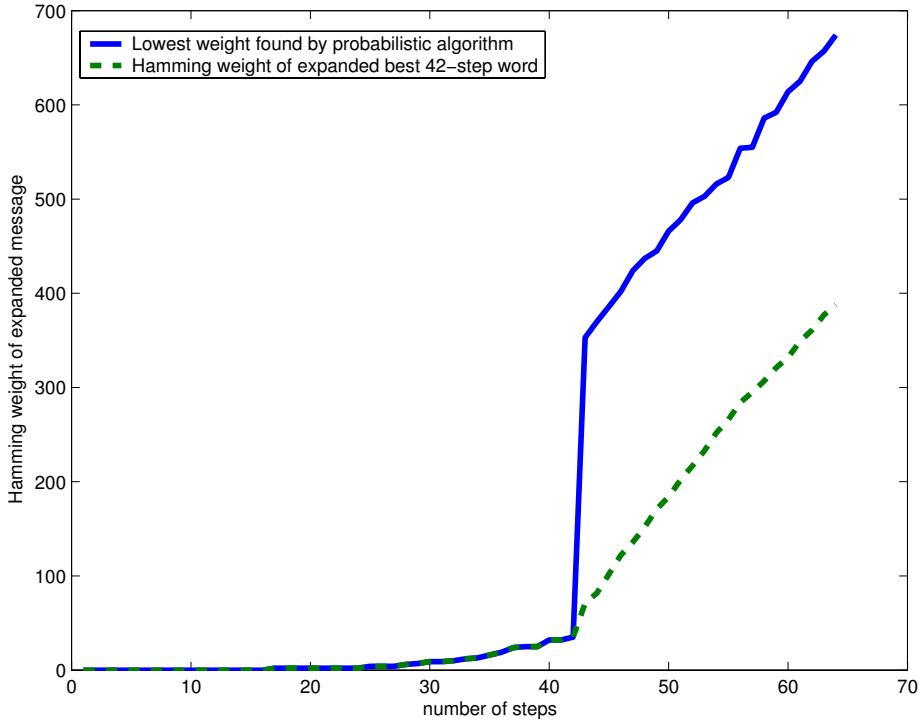

図 6.1: XOR で線形とした SHA-256 のメッセージ拡張をステップごとにみた最小ハミング重み

第7章　おわりに

本報告では、SHA-2 の安全性に関する Gilbert の 2003 年の論文以来、2006 年１月末までに公表された論文で、SHA-2 の安全性係わる記述が見られるものを網羅しまとめた。Wang らの一連のハッシュ関数の衝突探索の基本的な戦略は、

(1.) 差分パスの局部衝突の重ね合わせでメッセージ拡張関数通過後の語に対し成り立つ効果的な大域的な衝突（ハッシュ関数の衝突）を見つけること (disturbance vector 探索)

(2.) ＸＯＲ差分のみで無く符号付き算術差分も同時に解析し、確率高く伝播する差分パスを探索すること

(3.) 望ましい差分伝播が起こる内部状態に関する十分条件を解析し、差分伝播確率を高く保つメッセージ語変更法を見つけること

と考えることが出来る。

SHA-2 に関しても、現在までの研究で、同様な戦略がとられている。Gilbert らは、SHA-2 の圧縮関数を線形近似し、XOR 差分について、9 ステップで確率 2^{-66} の局部衝突を導いている。この局部衝突は最良のものであるが、大域的衝突を構成するには、SHA-256 の場合、この局部的衝突が少なくとも３個以上、SHA-384/-512 においては５個以上必要と見積もられ、SHA-2 は (1.) の戦略のみを持つ Joux 型の攻撃に対し安全であると結論づけている。

Hawkes らは、この局部衝突のみに,(2.)(3.) の戦略を適用している。符号付き算術差分も合わせて考察し、適切な拡張メッセージ語差分変更法を提案し、それを使うことにより、確率が 2^{-39} から 2^{-42} に上昇することを示している。

Lee らは、(3.) のメッセージ変更法が存在すると仮定して、SHA-256 の第 24 から 55 ステップを調査し、衝突には、15 個以上の局部衝突の重ね合わせが必要であり、Hawkes らの結果と合わせることにより、衝突発見の確率は、2^{-585} 以下であるとしている。新しいハッシュ関数 DHA-256 の提案に主眼がおかれている為、詳細は不明である。彼らは、この確率は、思いの外大きいとしているが、SHA-256 の安全性を脅かす結果ではない。

Yoshida らは、SHA-2 の全ての算術加算を XOR 加算で置き換えた SHA-2-XOR の差分特性を調査している。その圧縮関数部分を暗号化変換と考える SHACAL-2-XOR の差分特性を解析し, 確率 2^{-8} の１ステップ繰り返し差分パスを発見している。それを用いて、15 ステップ SHA-2-XOR の擬似衝突が 2^{120} の計算量で発見できる事、同じく、31 ステップ SHA-2-XOR は、ランダムハッシュ関数では無い事を示している。また、先頭 19 ステップの差分確率を 1 とするメッセージ変更法を提案し、残り 15 ステップに繰り返し差分特性を適用する事により、34 ステップ SHA-2-XOR の擬似衝突が、計算量 2^{120} で発見できると推定している。

Pramstaller らは、予備的調査として、SHA-256 のメッセージ拡張関数の効果を調べている。1 ビットのメッセージ差分は、SHA-1 においてハミング重み 107 の拡張メッセージ差分を引き起こすのに対し、SHA-256 の場合は、メッセージ拡張関数内の算術加算を XOR で置き換えて簡略化しても、ハミング重み 467 の差分を引き起こし、差分拡散性に優れている。さらに、この簡略化

メッセージ拡張関数の出力を線形符号と考え、確率的最小重み探索法を使って拡張メッセージ差分の最小ハミング重みが 42 ステップまでで 35 であることを求め、それを使って、フルステップで最小ハミング重み 356 と推定している。

以上この 3 年間の SHA-2 の安全性評価研究を総合するに、現時点の解析結果は、SHA-2 の衝突探索の成功確率は、誕生日攻撃の確率を上回るものではなく、その安全性を脅かすものではないと考えられる。

関連図書

[1] "Secure Hash Standard," National Institute of Standards and Technology, FIPS 180-2, Aug. 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf

[2] Bill Burr:,"Where should we go from here?",NIST Hash Function Workshop 2005,http://www.csrc.nist.gov/pki/HashWorkshop/2005/Nov1_Presentations/Burr_Future Strategy_Panel.pdf

[3] H.Gilbert and H.Handschuh: "Security Analysys of SHA-256 and Sisters", SAC 2003, LNCS 3006, pp.175-193, 2003

[4] Philip Hawkes, Michael Paddon, and Gregory G. Rose. On corrective patterns for the SHA-2 family. Cryptology ePrint Archive, Report 2004/207, August 2004. http://eprint.iacr.org/.

[5] Jesang Lee, Donghoon Chang, Hyun Kim, Eunjin Lee, Deukjo Hong, Jaechul Sung, Seokhie Hong, Sangjin Lee,:"A New 256-bit Hash Function DHA-256 : Enhancing the Security of SHA-256", NIST Hash Function Workshop 2005, http://www.csrc.nist.gov/pki/HashWorkshop/2005/Nov1_Presentations/ChangD_DHA256.pdf

[6] Krystian Matusiewicz, Josef Pieprzyk, Norbert Pramstaller, Christian Rechberger, Vincent Rijmen, "Analysis of simplified variants of SHA-256", WEWoRC 2005

[7] Bart Preneel:"Hash Functions: Past, Present and Future" ASCACRYPT 2005, http://www.cs.iitm.ernet.in/ ac05/Lectures/Bart_Preneel.pdf 2005

[8] Norbert Pramstaller, Christian Rechberger, Vincent Rijmen:"Preliminary Analysis of the SHA-256 Message Expansion",NIST Hash Function Workshop 2005,http://www.csrc.nist.gov/pki/HashWorkshop/2005/Oct31_Presentations/Rechberger_PreliminaryAnalysisOfSHA256.pdf

[9] Hirotaka Yoshida and Alex Biryukov.: " Analysis of a SHA-256 variant ", SAC 2005, 2005,

[10] 金子敏信："[招待論文] 共通鍵暗号に対する近年の攻撃－ CRYPTREC 暗号リストから２年－ "、電子情報通信学会技術研究報告、IT2005-36,pp.1-8,(2005.7)