

ハッシュ関数構成法のモデル化と安全性解析

平野敬之, 岸本渡

千葉大学大学院 融合科学研究科 情報科学専攻 知能情報コース 情報処理工学領域

概要 暗号理論では、任意長の入力を一定長に圧縮する関数(ハッシュ関数)を署名・認証・鍵配送などに応用し、その際に衝突対計算困難性や第2原像計算困難性などの安全性を要求する。固定長の入力を一定長に圧縮する関数(圧縮関数)を繰り返し使うことで反復型ハッシュ関数を構成できる。反復型ハッシュ関数の安全性は圧縮関数の安全性と繰り返し方に依存する。ある構成法に基づく反復型ハッシュ関数が圧縮関数の安全性を保存するかどうか、多数の安全性に関して解析することを Multi-Property-Preserving(MPP) 特性解析という。具体的な構成法に関して MPP 特性を解析した研究が存在するのに対し、本研究はモデル化した構成法に関して MPP 特性を解析する。

キーワード 反復型ハッシュ関数、衝突対計算困難性、第2原像計算困難性、Multi-Property-Preserving 特性解析、構成法のモデル

Modeling and safety analysis of hush function composition method

Takayuki Hirano, Wataru Kishimoto

Chiba University, Graduate School of Advanced Integration Science, Information Sciences
Division, Information Processing and Computer Sciences Department, Information
Processing and Computer Engineering Area

Abstract A repetitive type hush function is composed of repeated usages of a compression function that compresses a fixed length input into a fixed shorter output. In cryptology, some security requirements such as Col-Secure and Sec-Secure are needed for hush functions. The security of a repetitive type hush function depends on the security of the compression function and on how to repeat it. If a hush function, which is composed of a compression function with some security requirements preserves the same security requirements, the hush function is Multi-Property-Preserving (MPP). This research suggests a model of a repetitive type hush function and analyzes the MPP property of this model.

Key Words Repetitive type hush function, Col-Secure, Sec-Secure, Multi-Property-Preserving, Model of a repetitive type hush function.

1 はじめに

ハッシュ関数とは、任意長の入力を一定長に圧縮する関数である。暗号理論では、署名・認証・鍵配送などにハッシュ関数を応用する。ハッシュ関数は入力空間が出力空間より広い場合、衝突が必ず存在する。衝突はハッシュ関数を応用する署名・認証・鍵配送な

どに影響を及ぼすため、暗号理論ではハッシュ関数に安全性を要求する。Rogaway らは文献 [2] で衝突計算困難性や第2原像計算困難性などの安全性を定めた。

本研究では、固定長の入力を一定長に圧縮する関数を圧縮関数、圧縮関数を繰り返し使って構成するハッシュ関数を反復型ハッシュ関数と呼ぶ。反復型

ハッシュ関数の安全性は圧縮関数の安全性と繰り返し方に依存する。ある構成法に基づく反復型ハッシュ関数が圧縮関数の安全性を保存するかどうか、多数の安全性に関して解析することを MPP 特性解析という。

文献 [3] は Dedicated-Key という鍵で圧縮関数を鍵付き関数に拡張して Dedicated-Key Setting と呼んでいる。文献 [1][3][4] などは、具体的な構成法に対し、Dedicated-Key Setting において MPP 特性を構成法ごとに解析している。本研究では、反復型ハッシュ関数の安全性を解析することと安全な反復型ハッシュ関数の構成法を提案することを目的とし、反復型ハッシュ関数の構成法が圧縮関数をどのように繰り返し使うかをモデル化し、Dedicated-Key Setting において衝突計算困難性や第 2 原像計算困難性に関する MPP 特性を解析する。

本稿では、 \mathbb{N} は自然数の集合、 \mathbb{B} は 1 ビット値の集合、 \mathbb{K} は鍵をまとめた集合、 $u^{(v)}$ はビット列 u の最上位から v [ビット目]、 $u^{(v_1, v_2)}$ はビット列 u のうち $u^{(v_1)}$ から $u^{(v_2)}$ までのビット列、 u^v はビット値 u が v [個] だけ連続したビット列、 $u \| v$ はビット列 u, v の結合、 $u \oplus v$ はビット列 u, v の排他的論理和、 $|u|$ はビット列 u の長さ、 $|A|$ は集合 A の要素数、 $\Pr(u)$ は u が成り立つ確率、 $\max_{\Pr}(u)$ は $\Pr(u)$ の最大値、 $u \xleftarrow{\$} A$ は集合 A からの一様ランダムな u の撰択、 $v \xleftarrow{\$} A(u)$ は確率的アルゴリズム A が入力 u に対して v を出力することを表わす。

2 ハッシュ関数の構成法

任意長の入力を一定長に圧縮する関数をハッシュ関数、固定長の入力を一定長に圧縮する関数を圧縮関数、圧縮関数を繰り返し使って構成するハッシュ関数を反復型ハッシュ関数と呼ぶ。例えば、反復型ハッシュ関数の構成法である Markle-Damgård は、入力 M をパディングして M_1, \dots, M_m に m [分割] し、 K が Dedicated-Key の圧縮関数 f を繰り返し使って $H_1 = f_K(M_1 \| H_0), \dots, H_m = f_K(M_m \| H_{m-1})$ を計算し、 H_m を出力する。ここで、 H_0 は初期値である。

なお、本研究ではモデル化した構成法を対象とするため、その他の具体的な構成法の記載は省略する。詳しくは文献 [1][4] などを参照のこと。

3 ハッシュ関数の安全性

ある構成法に基づく反復型ハッシュ関数が圧縮関数の安全性を保存するかどうか、多数の安全性に関して解析することを MPP 特性解析という。本研究では、Dedicated-Key Setting における反復型ハッシュ関数の MPP 特性を、モデル化した構成法に対して解析する。モデルが表わすすべての構成法について圧縮関数が持つある安全性を反復型ハッシュ関数も必ず持つならば、**保存すると定義する**。モデルが表わすすべての構成法について圧縮関数が持つある安全性を反復型ハッシュ関数が持たない実例が存在するならば、**保存しないと定義する**。

文献 [2] を参考に、本研究が対象とするハッシュ関数の安全性を説明する。他にも、原像計算困難性や偽造不可能性などの安全性が存在する。

衝突計算困難性 (Col-Secure): 計算量が高々 t な任意のアルゴリズム A が式 (1) を満たすとき、Dedicated-Key を K とするハッシュ関数 φ は (t, ϵ) -Col-Secure と定義する。

$$\begin{aligned} \Pr \max [K \xleftarrow{\$} \mathbb{K}, (M, M') \xleftarrow{\$} A(K) : \\ M \neq M' \text{ かつ } \varphi_K(M) = \varphi_K(M')] < \epsilon. \quad (1) \end{aligned}$$

第 2 原像計算困難性 (Sec-Secure): 計算量が高々 t な任意のアルゴリズム A が式 (2) を満たすとき、Dedicated-Key を K とするハッシュ関数 φ は (t, ϵ, λ) -Sec-Secure と定義する。

$$\begin{aligned} \Pr \max [\lambda \in \mathbb{N}, K \xleftarrow{\$} \mathbb{K}, M \xleftarrow{\$} \mathbb{B}^\lambda, M' \xleftarrow{\$} A(K, M) : \\ M \neq M' \text{ かつ } \varphi_K(M) = \varphi_K(M')] < \epsilon. \quad (2) \end{aligned}$$

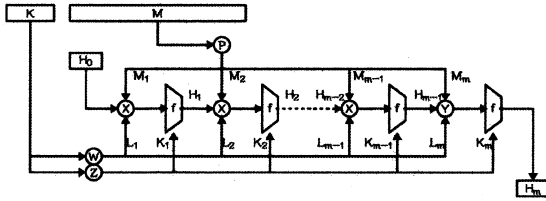
4 ハッシュ関数構成法のモデル化

文献 [1][3][4] などが MPP 特性を具体的な構成法ごとに解析したのに対し、本研究はモデル化した構成法に対して MPP 特性を解析する。モデル化により、モデルが表わす構成法の MPP 特性をまとめて解析する。また、安全性に優れた新しい構成法の発見を期待する。

反復型ハッシュ関数構成法のモデルを Hash-Composition-Method (HCM) と名付ける。 HCM は、初期値 H_0 、入力分割アルゴリズム P 、鍵スケジューラ Z, W 、圧縮関数 f 、前処理関数 X, Y によって反復型ハッシュ関数 F を構成する。 F は $K \in \mathbb{K}$ を鍵とし、任意長の入力 M の出力 H_m を以下の手順で計算する。①: 入力分割アルゴリズム P に M を与え、

m [個]の変数 M_1, \dots, M_m に分割させる。②: 鍵スケジューラ Z に K を与えて圧縮函数 f の Dedicated-Key として K_1, \dots, K_m を、鍵スケジューラ W に K を与えて前処理函数 X の鍵 L_1, \dots, L_{m-1} と前処理函数 Y の鍵 L_m とを生成させる。③: 定数 H_0 を初期値とし、 $i \in [1, m-1]$ について $H_i = f_{K_i}(X_{L_i}(M_i, H_{i-1}))$ を計算する。④: $H_m = f_{K_m}(Y_{L_m}(M_m, H_{m-1}))$ を計算し、出力する。

ハッシュ函数 F が鍵 K と入力 M とから出力 H_m を計算する概要を下図に示す。



初期値 H_0 、入力分割アルゴリズム P 、圧縮函数 f 、前処理函数 X, Y 、鍵スケジューラ Z, W は、以下のように設定する。 H_0 の設定: H_0 は初期値とし、公開な可変値とする。 H_0 が公開な固定値のときは、公開な可変値の特別な場合とする。 P の設定: P は M の末尾にパディングを施し、 m [分割] して M_1, \dots, M_m を生成するアルゴリズムとする。 P のパディングは、パディングの長さで一意に決まるタイプ(タイプ①)と $|M|$ と 1 対 1 に決まるタイプ(タイプ②)とのどちらかで決める。 f の設定: f は圧縮函数とする。 $i \in [1, m]$ に対し、 i [番目] の f は K_i を Dedicated-Key とし、 X や Y の出力を一定長に圧縮して H_i を出力する。 Z, W の設定: Z は f の Dedicated-Key として K_1, \dots, K_m を生成するスケジューラとする。 W は X の鍵として L_1, \dots, L_{m-1} を、 Y の鍵として L_m を生成するスケジューラとする。 K_1, \dots, K_m や L_1, \dots, L_m の値は F の鍵 K に基づき、 $K_1 = \dots = K_m$ かつ $L_1 = \dots = L_m$ を満たすパターン(パターン①)、入力 M' に生成される鍵である $K_1, \dots, K_{m'}$ や $L_1, \dots, L_{m'}$ に対し $K_m = K_{m'}$ かつ $L_m = L_{m'}$ を満たすパターン(パターン②)、 K_1, \dots, K_m が互いに独立で鍵 L_1, \dots, L_m も互いに独立となるパターン(パターン③)のいずれかで決める。 X の設定: X は、1 番目から $m-1$ [番目] までの f の入力を前処理する全単射な函数とする。 X の入力を M_i, H_{i-1} 、鍵を L_i とする。特に、任意の鍵 L_i

に対して $X_{L_i}(M_i, H_{i-1}) = M_i \| H_{i-1}$ ならば、 X は単純結合とする。 Y の設定: Y は、 m [番目] の f の入力を前処理する全単射な函数とする。 Y の入力を M_m, H_{m-1} 、鍵を L_m とする。文献 [1] が扱う構成法 MDP のように、 m [番目] の f の入力だけ置換するような構成法をモデル化すべく、 m [番目] の f の入力は X でなく Y に前処理させる。特に、任意の鍵 L_m に対して $Y_{L_m}(M_m, H_{m-1}) = M_m \| H_{m-1}$ ならば、 Y は単純結合とする。

5 モデルに対する MPP 特性解析

P, X, Y, Z, W を具体的に設定して構成法のモデル HCM_1, \dots, HCM_{24} を定義し、それぞれについて Col-Secure と Sec-Secure とに関する MPP 特性を解析した結果を下表に記す。

構成法	X	Y	Z, W	P	Col	Sec
HCM_1	単	単	①	①	×	×
HCM_2	単		①	①	×	?
HCM_3		単	①	①	?	×
HCM_4			①	①	?	?
HCM_5	単	単	②	①	×	×
HCM_6			②	①	×	?
HCM_7		単	②	①	?	×
HCM_8			②	①	?	?
HCM_9	単	単	③	①	×	×
HCM_{10}	単		③	①	?	?
HCM_{11}		単	③	①	?	×
HCM_{12}			③	①	?	?
HCM_{13}	単	単	①	②	○	×
HCM_{14}	単		①	②	○	?
HCM_{15}		単	①	②	○	×
HCM_{16}			①	②	○	?
HCM_{17}	単	単	②	②	○	×
HCM_{18}	単		②	②	○	?
HCM_{19}		単	②	②	○	×
HCM_{20}			②	②	○	?
HCM_{21}	単	単	③	②	×	×
HCM_{22}	単		③	②	?	?
HCM_{23}		単	③	②	?	×
HCM_{24}			③	②	?	?

単:単純結合、○:保存する、×:保存しない、?:不明

文献 [1] が扱う構成法のうち、CS は HCM_2 の、NI は HCM_5 の、MDP, KMDP', PMDP', sCS は HCM_{14} の、sNI は HCM_{17} の、ESh は HCM_{20} の具体例である。文献 [4] が扱う構成法のうち、Strengthened MD は HCM_{13} の、Enveloped MD は HCM_{14} の、Linear は HCM_{21} の具体例である。なお、文献 [4] が扱う構成法のうち、XOR-Linear, Prefix-free MD, Rndomized, HAIFA, Strengthened Merkle Tree, Tree Hash, XOR Tree は HCM で表せず、ROX はランダムオラクルを使うので対象外とした。モデルを改良し、これらを表すことを今後の課題としたい。

以降、解析に使った定理と解析結果の証明とを記す。ただし、以下のように特別な場合を考えた。

①: HCM_6 が Col-Secure を保存しないことを示し、 HCM_6 の特別な場合である HCM_1, HCM_2, HCM_5 が Col-Secure を保存しないとした。②: HCM_{20} が Col-Secure を保存することを示し、 HCM_{20} の特別な場合である $HCM_{13}, \dots, HCM_{19}$ が Col-Secure を保存するとした。③: HCM_7 が Sec-Secure を保存しないことを示し、 HCM_7 の特別な場合である HCM_1, HCM_3, HCM_5 が Sec-Secure を保存しないとした。④: HCM_{11} が Sec-Secure を保存しないことを示し、 HCM_{11} の特別な場合である HCM_9 が Sec-Secure を保存しないとした。⑤: HCM_{19} が Sec-Secure を保存しないことを示し、 HCM_{19} の特別な場合である $HCM_{13}, HCM_{15}, HCM_{17}$ が Sec-Secure を保存しないとした。⑥: HCM_{23} が Sec-Secure を保存しないことを示し、 HCM_{23} の特別な場合である HCM_{21} が Sec-Secure を保存しないとした。

定理 1 $K \in \mathbb{K}$ が Dedicated-Key の圧縮関数 $g : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^{n-1}$ の存在を仮定し、 K が Dedicated-Key の圧縮関数 $f : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^n$ を式 (3) のように定めると、 g が $(t+\delta, \epsilon)$ -Col-Secure ならば f は (t, ϵ) -Col-Secure となる。ただし、 δ は $d+n$ に対して定数オーダーである。 $U_0 \in \mathbb{B}^d, V_0 \in \mathbb{B}^n$ は任意の定数である。

$$f_K(R) = \begin{cases} V_0, & R = U_0 \| V_0 \\ g_K(R) \| \overline{V_0}^{(n)}, & R \neq U_0 \| V_0 \end{cases} \quad (3)$$

証明: 対偶を考え、 f の Col-Secure を破るアルゴリズム A_f の存在を仮定し、 g の Col-Secure を破るアルゴリズム A_g の存在を示す。 A_g は Dedicated-Key として K を受け取り、 A_f に K を与えて (R, R') を獲得し、獲得した (R, R') を出力する。このとき、 A_g の計算量は $t+\delta$ となる。 δ は $d+n$ に対して定数オーダーである。また、 R と R' に関して場合分けすると、 $R \neq U_0 \| V_0$ かつ $R' \neq U_0 \| V_0$ だけが矛盾なく生起する。従って、 $\Pr(A_g) \geq \Pr(A_f, A_g) = \Pr(A_f) \geq \epsilon$ より A_g の成功確率は ϵ 以上になる。

定理 2 $K \in \mathbb{K}$ が Dedicated-Key の圧縮関数 $g : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^{n-1}$ の存在を仮定し、 K が Dedicated-Key の圧縮関数 $f : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^n$ を式 (4) のように定めると、 $\epsilon \geq |\mathbb{D}|^{-1}$ のとき、 g が $(t+\delta, \epsilon - |\mathbb{D}|^{-1}, d+n)$ -Sec-Secure ならば f は $(t, \epsilon, d+n)$ -Sec-Secure となる。ただし、 δ は $d+n$ に対して定数オーダーである。 $h : \mathbb{B}^{d+n} \rightarrow \mathbb{D}$ は一様ランダムな入力に対して出力が一様ランダムに分布する任意の関数、 $B_0 \in \mathbb{B}^n, D_0 \in$

\mathbb{D} は任意の定数である。

$$f_K(R) = \begin{cases} B_0, & h(R) = D_0 \\ g_K(R) \| \overline{B_0}^{(n)}, & h(R) \neq D_0 \end{cases} \quad (4)$$

証明: 対偶を考え、 f の Sec-Secure を破るアルゴリズム A_f の存在を仮定し、 g の Sec-Secure を破るアルゴリズム A_g の存在を示す。 A_g は入力 R と Dedicated-Key として K とを受け取り、 A_f に R と K とを与えて R' を獲得し、獲得した R' を出力する。このとき、 A_g の計算量は $t+\delta$ となる。 δ は $d+n$ に対して定数オーダーである。また、 R と R' に関して場合分けすると、 $h(R) = D_0$ かつ $h(R') = D_0$ と $h(R) \neq D_0$ かつ $h(R') \neq D_0$ とが矛盾なく生起する。従って、 $\Pr(A_g) = \Pr(A_f, h(R) \neq D_0) \geq \epsilon - |\mathbb{D}|^{-1}$ より A_g の成功確率は $\epsilon - |\mathbb{D}|^{-1}$ 以上となる。

定理 3 $K \in \mathbb{K}$ が Dedicated-Key の圧縮関数 $g : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^{n-2}$ の存在を仮定し、 K が Dedicated-Key の圧縮関数 $f : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^n$ を式 (5) のように定めると、 g が $(t+\delta, \epsilon)$ -Col-Secure ならば f は (t, ϵ) -Col-Secure となる。ただし、 δ は $d+n$ に対して定数オーダーである。 $V_0 \in \mathbb{B}^n, U_0 \in \mathbb{B}^d$ は任意の定数、 $U_1 \in \mathbb{B}^d$ は $U_1 \neq U_0$ を満たす任意の定数、 $U_2 \in \mathbb{B}^d$ は $U_2 \neq U_0$ を満たす任意の定数である。

$$f_K(R) = \begin{cases} \textcircled{1} & V_0, \quad R = U_0 \| V_0 \\ \textcircled{2} & 0^{n-1} \| \overline{V_0}^{(n)}, \quad K^{(1)} = 0 \text{ かつ } R = U_1 \| V_0 \\ \textcircled{3} & 0^{n-1} \| \overline{V_0}^{(n)}, \quad K^{(1)} = 1 \text{ かつ } R = U_2 \| V_0 \\ \textcircled{4} & g_K(R) \| 1 \| \overline{V_0}^{(n)}, \quad \text{上記を除く場合} \end{cases} \quad (5)$$

証明: 対偶を考え、 f の Col-Secure を破るアルゴリズム A_f の存在を仮定し、 g の Col-Secure を破るアルゴリズム A_g の存在を示す。 A_g は Dedicated-Key として K を受け取り、 A_f に K を与えて (R, R') を獲得し、獲得した (R, R') を出力する。このとき、 A_g の計算量は $t+\delta$ となる。 δ は $d+n$ に対して定数オーダーである。また、 R と R' に関して場合分けすると、 R が $\textcircled{4}$ かつ R' が $\textcircled{4}$ だけが矛盾なく生起する。従って、 $\Pr(A_g) \geq \Pr(A_f, A_g) = \Pr(A_f) \geq \epsilon$ より A_g の成功確率は ϵ 以上になる。

5.1 HCM_6 は Col-Secure を保存しない

$f : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^n$ は Col-Secure だが F は Col-Secure ではない実例を示す。ただし、Col-Secure な圧縮関数 $g : \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^{n-1}$ の存在を仮定する。

定理 1 において、 $U_0 \in \mathbb{B}^d$ を任意の定数、 $V_0 \in \mathbb{B}^n$ をハッシュ関数の初期値 H_0 とし、式 (6) のように f

を定義すると、 f は Col-Secure となる。

$$f_K(R) = \begin{cases} H_0, & R = U_0 \| H_0 \\ g_K(R) \| \overline{H_0}^{(n)}, & R \neq U_0 \| H_0 \end{cases} \quad (6)$$

HCM_6 に基づいて式 (6) の f から F を構成し、入力 $M = U_0$ と入力 $M' = U_0 \| U_0$ とを同じ鍵 K のもとで F に与えたとする。 HCM_6 の Z, W はパターン②だから、 Z は $K_2 = K_3 = K_0$ を満たすように K_1, K_2, K_3 を、 W は $L_2 = L_3 = L_0$ を満たすように L_1, L_2, L_3 を生成したとする。 HCM_6 の P はタイプ①だから、 M は $M_1 = U_0$ とパディングだけからなる M_2 とに、 M' は $M'_1 = M'_2 = U_0$ とパディングだけからなる M'_3 とに分割される。さらに、パディングの長さが等しいため $M_2 = M'_3$ となり、 $Y_{L_0}(M_2, H_0) = Y_{L_0}(M'_3, H_0)$ が成り立つ。 HCM_6 の X は単純結合だから、式 (7) と式 (8) とより $H_2 = H'_3$ を導ける。

$$\begin{cases} H_1 = f_{K_1}(M_1 \| H_0) = H_0 \\ H_2 = f_{K_0}(Y_{L_0}(M_2, H_1)) = f_{K_0}(Y_{L_0}(M_2, H_0)) \end{cases} \quad (7)$$

$$\begin{cases} H'_1 = f_{K_1}(M'_1 \| H_0) = H_0 \\ H'_2 = f_{K_2}(M'_2 \| H'_1) = H_0 \\ H'_3 = f_{K_0}(Y_{L_0}(M'_3, H'_2)) = f_{K_0}(Y_{L_0}(M'_3, H_0)) \end{cases} \quad (8)$$

以上より、 $M \neq M'$ かつ $H_2 = H'_3$ だから $M = U_0$ と $M' = U_0 \| U_0$ とは F の衝突対となり、 f は Col-Secure だが F は Col-Secure ではない。

5.2 HCM_7 は Sec-Secure を保存しない

$f: \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^n$ は Sec-Secure だが F は Sec-Secure ではない実例を示す。ただし、Sec-Secure な圧縮関数 $g: \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^{n-1}$ の存在と、 P が結合するパディングが充分に長いことを仮定する。

長さ λ の入力 M に対して P が M_m に結合するパディングを ξ とする。定理 2 において、 $D_0 \in \mathbb{B}^{|\xi|}$ を ξ 、 $h(R) = R^{(d-|\xi|, d)}$ とし、 $B_0 \in \mathbb{B}^n$ を任意に定め、式 (9) のように f を定めると、 f は Sec-Secure となる。

$$f_K(R) = \begin{cases} B_0, & R^{(d-|\xi|, d)} = \xi \\ g_K(R) \| \overline{B_0}^{(n)}, & R^{(d-|\xi|, d)} \neq \xi \end{cases} \quad (9)$$

HCM_7 に基づいて式 (9) の f から F を構成する。与えられた $K \stackrel{\$}{\leftarrow} \mathbb{K}, M \stackrel{\$}{\leftarrow} \mathbb{B}^\lambda$ に対し、 $M \neq M'$ かつ $|M| = |M'|$ を満たすように M' を定めると、 M' は F における M の第 2 原像となる。 $|M| = |M'|$ だから、 Z は M にも M' にも K_1, \dots, K_m を、 W は M にも M' にも L_1, \dots, L_m を生成する。 HCM_7 の Y は単純結合だから、入力 M の出力は式 (10) のように、入力 M' の出力は式 (11) のように計算される。

$$\begin{cases} H_1 = f_{K_1}(X_{L_1}(M_1, H_0)) \\ \dots \\ H_{m-1} = f_{K_{m-1}}(X_{L_{m-1}}(M_{m-1}, H_{m-2})) \\ H_m = f_{K_m}(M_m \| H_{m-1}) \end{cases} \quad (10)$$

$$\begin{cases} H'_1 = f_{K_1}(X_{L_1}(M'_1, H_0)) \\ \dots \\ H'_{m-1} = f_{K_{m-1}}(X_{L_{m-1}}(M'_{m-1}, H'_{m-2})) \\ H'_m = f_{K_m}(M'_m \| H'_{m-1}) \end{cases} \quad (11)$$

$h(M_m \| H_{m-1}) = M_m^{(d-|\xi|, d)} = \xi$ より $H_m = B_0$ が、 $h(M'_m \| H'_{m-1}) = M'_m^{(d-|\xi|, d)} = \xi$ より $H'_m = B_0$ が成り立つ。従って、 $M \neq M'$ かつ $H_m = H'_m$ だから M' は F における M の第 2 原像となり、 f は Sec-Secure だが F は Sec-Secure ではない。

5.3 HCM_9 は Col-Secure を保存しない

$f: \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^n$ は Col-Secure だが F は Col-Secure ではない実例を示す。ただし、Col-Secure な圧縮関数 $g: \mathbb{K} \times \mathbb{B}^{d+n} \rightarrow \mathbb{B}^{n-2}$ の存在を仮定する。

長さ d の入力 M に対して P が生成する M_m の値を ξ 、長さ $2d$ の入力 M' に対して P が生成する M'_m の値を ξ' とする。定理 3 において、 $V_0 \in \mathbb{B}^n$ をハッシュ関数の初期値 H_0 、 $U_1 \in \mathbb{B}^d$ を ξ 、 $U_2 \in \mathbb{B}^d$ を ξ' 、 $U_0 \in \mathbb{B}^d$ を $U_0 \neq \xi$ かつ $U_0 \neq \xi'$ を満たす任意の定数とし、式 (12) のように f を定めると、 f は Col-Secure となる。

$$f_K(R) = \begin{cases} H_0, R = U_0 \| H_0 \\ 0^{n-1} \| \overline{H_0}^{(n)}, K^{(1)} = 0 \text{ かつ } R = \xi \| H_0 \\ 0^{n-1} \| \overline{H_0}^{(n)}, K^{(1)} = 1 \text{ かつ } R = \xi' \| H_0 \\ g_K(R) \| 1 \| \overline{H_0}^{(n)}, \text{ 上記を除く場合} \end{cases} \quad (12)$$

HCM_9 に基づいて式 (12) の f から F を構成し、入力 $M = U_0$ と入力 $M' = U_0 \| U_0$ とを同じ鍵 K のもとで F に与えたとする。 $K_2^{(1)} = 0$ かつ $K_3^{(1)} = 1$ ならば、 HCM_9 の X, Y は単純結合だから、式 (13) と式 (14) とより $H_2 = H'_3$ を導ける。

$$\begin{cases} H_1 = f_{K_1}(M_1 \| H_0) = H_0 \\ H_2 = f_{K_2}(M_2 \| H_1) = 0^{n-1} \| \overline{H_0}^{(n)} \end{cases} \quad (13)$$

$$\begin{cases} H'_1 = f_{K_1}(M'_1 \| H_0) = H_0 \\ H'_2 = f_{K_2}(M'_2 \| H'_1) = H_0 \\ H'_3 = f_{K_3}(M'_3 \| H'_2) = 0^{n-1} \| \overline{H_0}^{(n)} \end{cases} \quad (14)$$

もし、 $K_2^{(1)} = 0$ かつ $K_3^{(1)} = 1$ ならば、 $M \neq M'$ かつ $H_2 = H'_3$ だから $M = U_0$ と $M' = U_0 \| U_0$ とは F の衝突対となり、 f は Col-Secure だが F は Col-Secure とならない。 HCM_9 の Z はパターン③だから K_2 と K_3 とは互いに独立であり、 $K_2^{(1)} = 0$ かつ $K_3^{(1)} = 1$ となる確率は $\frac{1}{4}$ であると期待する。

5.4 HCM_{11} は Sec-Secure を保存しない

HCM_7 の場合と同様に示せる。

5.5 HCM₁₉ は Sec-Secure を保存しない

HCM₇ の場合と同様に示せる。

5.6 HCM₂₀ は Col-Secure を保存する

待遇を考え、 t の計算量と ϵ 以上の確率とで F の衝突対を出力するアルゴリズム A_F の存在を仮定し、 $t + \delta$ の計算量と ϵ 以上の確率とで f の衝突対を出力するアルゴリズム A_f の存在を示す。ただし、 δ は F の入力長に対して多項式オーダーである。

$|M| \leq |M'|$ を満たすように命名した M と M' とを同じ鍵 K のもとで F に入力したとする。HCM₂₀ の Z, W はパターン②だから、 Z は $K_m = K_{m'} = K_0$ を満たすように $K_1, \dots, K_m, \dots, K_{m'}, \dots, K_{m'}$ を、 W は $L_m = L_{m'} = L_0$ を満たすように $L_1, \dots, L_m, \dots, L_{m'}$ を生成したとする。このとき、 F は K, M に対して式 (15) のように出力 H_m を、 K, M' に対して式 (16) のように出力 $H'_{m'}$ を計算するものとする。

$$\begin{cases} H_1 = f_{K_1}(X_{L_1}(M_1, H_0)) \\ \dots \\ H_{m-1} = f_{K_{m-1}}(X_{L_{m-1}}(M_{m-1}, H_{m-2})) \\ H_m = f_{K_0}(Y_{L_0}(M_m, H_{m-1})) \end{cases}, (15)$$

$$\begin{cases} H'_1 = f_{K_1}(X_{L_1}(M'_1, H_0)) \\ \dots \\ H'_{m'-1} = f_{K_{m'-1}}(X_{L_{m'-1}}(M'_{m'-1}, H'_{m'-2})) \\ H'_{m'} = f_{K_0}(Y_{L_0}(M'_{m'}, H'_{m'-1})) \end{cases} (16)$$

A_f は鍵 K を受け取り、 K を A_F に与えて (M, M') を獲得し、式 (15) と式 (16) とより H_1, \dots, H_m と $H'_1, \dots, H'_{m'}$ とを計算する。このとき、以下のように動作すれば A_f は確率 ϵ 以上で f の衝突対を出力できる。場合①: $|M| < |M'|$ ならば、 $Y_{L_0}(M_m, H_{m-1})$ と $Y_{L_0}(M'_{m'}, H'_{m'-1})$ とを出力する。HCM₂₀ の P はタイプ②なので、 $|M| < |M'|$ より $M_m \neq M'_{m'}$ となり、同じ鍵 L_0 のもとでは $Y_{L_0}(M_m, H_{m-1}) \neq Y_{L_0}(M'_{m'}, H'_{m'-1})$ となる。従って、 $Y_{L_0}(M_m, H_{m-1}) \neq Y_{L_0}(M'_{m'}, H'_{m'-1})$ かつ $H_m = H'_{m'}$ が成り立っている。場合②: $|M| = |M'|$ かつ $Y_{L_0}(M_m, H_{m-1}) \neq Y_{L_0}(M'_{m'}, H'_{m'-1})$ ならば、 $Y_{L_0}(M_m, H_{m-1})$ と $Y_{L_0}(M'_{m'}, H'_{m'-1})$ とを出力する。このとき、 $Y_{L_0}(M_m, H_{m-1}) \neq Y_{L_0}(M'_{m'}, H'_{m'-1})$ かつ $H_m = H'_{m'}$ が成り立っている。場合③: $|M| = |M'|$ かつ $Y_{L_0}(M_m, H_{m-1}) = Y_{L_0}(M'_{m'}, H'_{m'-1})$ ならば、 $M_{m-i} \neq M'_{m'-i}$ となる最小の i に対し、 $X_{L_{m-i}}(M_{m-i}, H_{m-i-1})$ と $X_{L_{m'-i}}(M'_{m'-i}, H'_{m'-i-1})$ とを出力する。 $|M| = |M'|$

より m' を m とすると、 $M \neq M'$ だから $M_{m-i} \neq M'_{m'-i}$ となる i が区間 $[1, m-1]$ に必ず存在し、 $X_{L_{m-i}}(M_{m-i}, H_{m-i-1}) \neq X_{L_{m-i}}(M'_{m'-i}, H'_{m'-i-1})$ となる。 $M_{m-i} \neq M'_{m'-i}$ となる最小の i に対し、 $H_{m-i} = H'_{m-i}, \dots, H_{m-1} = H'_{m-1}$ かつ $M_{m-i+1} = M'_{m-i+1}, \dots, M_m = M'_m$ となるから、 $X_{L_{m-i}}(M_{m-i}, H_{m-i-1}) \neq X_{L_{m-i}}(M'_{m'-i}, H'_{m'-i-1})$ かつ $H_{m-i} = H'_{m-i}$ が成り立っている。

A_f の計算量は $t + \delta$ となり、場合③を考慮すると δ は F の入力の長さに対して多項式オーダーとなる。

5.7 HCM₂₁ は Col-Secure を保存しない

HCM₉ の場合と同様に示せる。

5.8 HCM₂₃ は Sec-Secure を保存しない

HCM₇ の場合と同様に示せる。

6 おわりに

本研究では、反復型ハッシュ関数の安全性を解析することと安全な構成法を提案することとを目的に、反復型ハッシュ関数の構成法が圧縮関数をどのように繰り返し使うかをモデル化し、いくつかの構成法の MPP 特性を Dedicated-Key Setting において解析した。今後は、解析できなかった MPP 特性を解析したい。また、モデルにおける P, Z, W, X, Y の設定を改良し、より多くの構成法を表せるようにしたい。

本研究の遂行や本稿の作成にあたり、千葉大学の松葉育雄教授と須鎗弘樹准教授とをはじめ、多くの方々々に貴重な助言や指導をいただいた。ここに厚く謝意を表したい。

参考文献

- [1] 中村俊吾, 岩田哲; "反復型ハッシュ関数の MPP 特性解析"; SCIS2008.
- [2] P. Rogaway, T. Shrimpton; "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance"; FSE 2004.
- [3] Mihir Bellare, Thomas Ristenpart; "Hash Functions in the Dedicated-Key Setting: Design Choices and MPP Transforms"; ICALP '07.
- [4] Elena Andreeva, Gregory Neven, Bart Preneel, Thomas Shrimpton; "Seven-Property-Preserving Iterated Hashing: ROX"; ASIACRYPT 2007.