論 文 の 内 容 の 要 旨

論文題目　Design and Evaluation of Cryptographic Hash Functions
(和訳　暗号学的ハッシュ関数の構成と評価に関する研究)

氏名　　李 善 英

ハッシュ関数は現代暗号における基本的な構成要素の１つである。特にユーザやメッセージの正当性を認証する認証コード（MAC)・ディジタル署名などの認証技術におけるハッシュ関数の利用は認証の安全性・効率性と密接な関係がある。すなわち、安全なハッシュ関数の構成は安全な暗号システムの構築を可能にする。したがって、ハッシュ関数に関する研究の目的は安全なハッシュ関数の構成にある。そこで注目すべきことはハッシュ関数の安全性の評価法である。ハッシュ関数の安全性は与えられたメッセージに対してそれと同じハッシュ値を持つ別のメッセージ、すなわち、衝突を探すのに必要な計算量で評価される。衝突を探すのに全探索意外の方法が見つからなかった場合、そのハッシュ関数は安全であると考えられる。

ハッシュ関数をユニバーサルハッシュ関数と衝突困難ハッシュ関数の２つのクラスに分けて考える。ユニバーサルハッシュ関数はその安全性が衝突確率で証明できるが、鍵という要素が必要であるためコストがかかる。また、MD4 に基づいて構成される衝突困難ハッシュ関数は高速であるが、その安全性の評価は困難である。本論文ではこの２つのクラスのハッシュ関数に対し安全なハッシュ関数を構成し、評価する。安全性が衝突確率で証明できるユニバーサルハッシュ関数の問題点はハッシュ値の計算に時間がかかることである。そのため最近は高速なユニバーサルハッシュ関数の構成に関する研究が盛んに行われている。ここでは現実的に使用可能な高速かつ安全なユニバーサル

ハッシュ関数としてSquare Hashに基づいたハッシュ関数SNH(Square and Non-linear Hash)を構成し、その安全性を衝突確率で評価する。提案ハッシュ関数SNHはSquare Hashに比べ計算量が多少多いが、現在のコンピューターでは実行時間はほぼ同じである。ハッシュ値の計算がwの長さ単位で行われる時、提案ハッシュ関数の衝突確率は2^{-w}である。これはSquare Hashの3分の1の衝突確率である。

現在、もっとも幅広く使われている衝突困難ハッシュ関数はMD4に基づいたMDx族ハッシュ関数である。MDx族ハッシュ関数は簡単な演算の繰り返しでハッシュ値を計算する。しかし、MDx族ハッシュ関数は"trial-and-error" procedureによって構成されるため、その理論的な安全性の評価が困難である。MDx族ハッシュ関数は非線形関数・法2^{32}での加算・巡回シフトの演算を用いて各ステップで連鎖変数の値を計算し、その変数の値を用いてハッシュ値を計算する。各ステップで定義される関数をステップ関数(Step function)と呼ぶ。また、ステップ関数の繰り返しで構成される部分を圧縮関数(Compression function)、最後にハッシュ値を出力部分を出力関数(Output function)と呼ぶ。差分攻撃を仮定し場合、入力の小さな変化がすべての出力に影響を及ぼすように構成されたハッシュ関数は安全であると考えられている。そこでこのような入力の差分と出力のハッシュ値の差分の間の関係を調べるためにステップ関数の演算である非線形関数と法2^{32}での加算を排他的論理和として近似するモデルを提案し、MD5のステップ関数の評価を行う。入力の14番目を少し変化させた異なる入力メッセージでMD5の連鎖の特徴を評価するとハッシュ値を計算するための4つの連鎖変数の2つだけが入力の差分によって変化される。これは、変化された2つの変数に対し、ある操作を加えることで衝突が発見できることを示している。また、衝突が発見できなくても、全段階のMD5におけるalmost-collisionの存在可能性を示している。almost-collisionを防ぐためにはハッシュ関数の入力に多重置換を用いる方法が提案されているが、その場合、余分なメモリーや演算が必要である。我々はMD5の変数の連鎖に注目し、入力の差分を広げる方法を提案する。MD5の連鎖変数(A,B,C,D)は$A \to D \to C \to B$の順で計算していくが、提案方式は各変数を複数の連鎖パターンを用いて計算することで入力の差分を広げることができる。実際の評価は実験により示す。 提案した複数の連鎖パターンを用いるハッシュ関数の圧縮関数を表 1で示す。提案方式はステップ関数で使われる変数の順番だけを変えるため、余分のメモリも作業も必要ない。

差分攻撃に耐性を持たせるもう1つの方法として、入力の差分を拡散するように新しい入力メッセージを生成する方法を提案する。入力の差分は新しいメッセージを構成する過程で拡散され、圧縮関数によっての差分がある程度減少しても、新しい入力が常に

表 1: 複数の連鎖パターンを含む圧縮関数

$B^{(i+1)}$	$:= C^{(i)} + (B^{(i)} + f^{(i+1)}(C^{(i)}, D^{(i)}, A^{(i)}) + X^{(j)} + K^{(i+1)})$
$D^{(i+2)}$	$:= A^{(i+1)} + (D^{(i+1)} + f^{(i+2)}(A^{(i+1)}, B^{(i+1)}, C^{(i+1)}) + X^{(k)} + K^{(i+2)})$
$A^{(i+3)}$	$:= B^{(i+2)} + (A^{(i+2)} + f^{(i+3)}(B^{(i+2)}, C^{(i+2)}, D^{(i+2)}) + X^{(l)} + K^{(i+3)})$
$C^{(i+4)}$	$:= D^{(i+3)} + (C^{(i+3)} + f^{(i+4)}(D^{(i+3)}, A^{(i+3)}, B^{(i+3)}) + X^{(n)} + K^{(i+4)})$
$C^{(i+1)}$	$:= D^{(i)} + (C^{(i)} + f^{(i+1)}(D^{(i)}, A^{(i)}, B^{(i)}) + X^{(j)} + K^{(i+1)})$
$A^{(i+2)}$	$:= B^{(i+1)} + (A^{(i+1)} + f^{(i+2)}(B^{(i+1)}, C^{(i+1)}, D^{(i+1)}) + X^{(k)} + K^{(i+2)})$
$B^{(i+3)}$	$:= C^{(i+2)} + (B^{(i+2)} + f^{(i+3)}(C^{(i+2)}, D^{(i+2)}, A^{(i+2)}) + X^{(l)} + K^{(i+3)})$
$D^{(i+4)}$	$:= A^{(i+3)} + (D^{(i+3)} + f^{(i+4)}(A^{(i+3)}, B^{(i+3)}, C^{(i+3)}) + X^{(n)} + K^{(i+4)})$
$A^{(i+1)}$	$:= B^{(i)} + (A^{(i)} + f^{(i+1)}(B^{(i)}, C^{(i)}, D^{(i)}) + X^{(j)} + K^{(i+1)})$
$C^{(i+2)}$	$:= D^{(i+1)} + (C^{(i+1)} + f^{(i+2)}(D^{(i+1)}, A^{(i+1)}, B^{(i+1)}) + X^{(k)} + K^{(i+2)})$
$D^{(i+3)}$	$:= A^{(i+2)} + (D^{(i+2)} + f^{(i+3)}(A^{(i+2)}, B^{(i+2)}, C^{(i+2)}) + X^{(l)} + K^{(i+3)})$
$B^{(i+4)}$	$:= C^{(i+3)} + (B^{(i+3)} + f^{(i+4)}(C^{(i+3)}, D^{(i+3)}, A^{(i+3)}) + X^{(n)} + K^{(i+4)})$
$D^{(i+1)}$	$:= A^{(i)} + (D^{(i)} + f^{(i+1)}(A^{(i)}, B^{(i)}, C^{(i)}) + X^{(j)} + K^{(i+1)})$
$B^{(i+2)}$	$:= C^{(i+1)} + (B^{(i+1)} + f^{(i+2)}(C^{(i+1)}, D^{(i+1)}, A^{(i+1)}) + X^{(k)} + K^{(i+2)})$
$C^{(i+3)}$	$:= D^{(i+2)} + (C^{(i+2)} + f^{(i+3)}(D^{(i+2)}, A^{(i+2)}, B^{(i+2)}) + X^{(l)} + K^{(i+3)})$
$A^{(i+4)}$	$:= B^{(i+3)} + (A^{(i+3)} + f^{(i+4)}(B^{(i+3)}, C^{(i+3)}, D^{(i+3)}) + X^{(n)} + K^{(i+4)})$

$(A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)})$	: i ステップ後の連鎖変数,
$+$	: 2^{32} 法での加算,
$f^{(i)}$	: i 段階の非線形関数

オリジナル入力の差分を保つため、最後まで入力の差分を出力に残すことができる。このハッシュ関数の入力として使われる新しいメッセージはマルコフ連鎖に基づいて構成されるので、衝突を探索するためには同じ推移確率行列をもつメッセージを探さなければならない。ハッシュ関数の出力が n ビットとする時、提案ハッシュ関数は衝突の探索に $2^{\frac{n}{2}}$ の計算量が必要である。この計算量は理想的な安全性を示している。