パスワード認証に用いるハッシュ関数の比較

1090324　越智 敦司　【 福本研究室 】

1　はじめに

ユーザの認証を行う方法の一つとして，パスワードがある．一般にユーザ認証は，入力されたパスワードをハッシュ化し，あらかじめハッシュ化し保存していたハッシュ値と比較を行う．比較結果が同一であれば，正規のユーザであることになる．このハッシュ関数としてMD5 が広く使われていたが，MD5 には脆弱性が見つかっており他のハッシュ関数への移行が進んでいる [1]．本研究では，移行によりハッシュ関数の高速性，ランダム性について変化が見られるかを調べるため，現在使用されているハッシュ関数を，生成時間および特徴ある入力に対する出力の偏りに関して比較を行う．

2　ハッシュ関数の比較

ハッシュ関数の基本構造を図 1 に示す．入力メッセージ M が m ビットの倍数になるように M の末尾にデータが付加される (パディング)．次にメッセージを m ビット毎に分割する．そのときのメッセージを $M = (M^{(1)}||M^{(2)}||\cdots||M^{(N)})$ とする．m ビットの固定長に分割されたメッセージ $M^{(i)}$ が順次，圧縮関数 f へ入力され，最後に出力関数 g によりハッシュ値 H が得られる．本研究ではハッシュ関数である MD5，MD4，SHA1，RIPEMD160 をハッシュ値の生成時間と出力結果の偏りについて比較を行う．

2.1　生成時間

ユーザ認証をする際にワンタイムパスワードのS/KEY という手法が用いられる [2]．この手法では，パスワードを任意の回数ハッシュ化を行う．そのためS/KEY にはハッシュ関数の高速性が要求される．そこで，任意の回数繰り返しハッシュ化するためにかかる実行時間を計測し，比較を行った．繰り返し回数を，10,000,000 回とし，1 回当たりの生成時間を平均値から算出する．RIPEMD160 は 6.7μsec となり，他のハッシュ関数に比べ実行に時間を要し，繰り返し回数が多くなるほどその差は大きくなったが，現実時間で見た場合，実行時間は十分短く，いずれの方式も生成時間では問題はない．

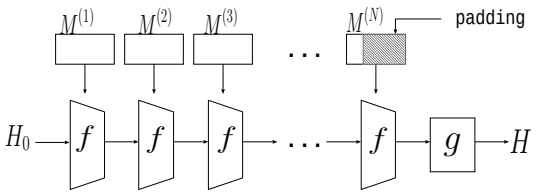

図 1　ハッシュ関数の基本構造

表 1　出力の分散

ハッシュ関数	3 文字	4 文字
MD5	1234	31826
MD4	1070	27850
SHA1	751	22038
RIPEMD160	924	19819

2.2　出力の偏り

ハッシュ関数の出力は，入力の値が少しでも変化した場合，全ての出力結果に変化が現れる，つまり似たような入力に対しても出力結果の偏りがないことが望まれる．ハッシュ関数に任意の文字列の入力を行い，その出力の偏りを調べ，比較を行う．

本研究ではパスワード認証を対象としているため，ハッシュ関数への入力はアルファベットと記号と数字の組合せとする．安全性という観点では，パスワードはランダムな文字の組合せが理想的ではあるが，実際には，名前や英単語など，意味のある文字列や，覚えやすくするために短いパスワードが利用されることが多く，入力には偏りが生じやすい．このことを考慮し，本実験では，小文字のアルファベット “a”から “z”までの 3 文字の組合せと 4 文字の組合せを入力とする．得られた出力値を 8 ビット毎に分け，16 進数 00 から FF の 256 種類に分類し出現頻度を求める．出力の長さが，MD5 と MD4 は 128 ビット，SHA1 と RIPEMD160 は 160 ビットであるため，SHA1 と RIPEMD160 には係数 0.8 を掛け，出力値の範囲を 128 ビットに揃える．出力値の平均値を求め，平均値からの出力値の散らばり具合を示すために分散を求める．分散は値が 0 に近いほど出力の偏りは小さい．実験結果を表 1 に示す．3 文字の組合せと 4 文字の組合せの入力をし，ハッシュ関数 MD5 の分散が他に比べ大きくなり出力に偏りがあることが確認できる．

3　まとめ

生成時間は，RIPEMD160 は他のハッシュ関数に比べ時間を要したが，現実時間で見た場合はこの生成時間で十分であり，いずれの方式も生成時間では問題はない．3 文字または，4 文字の小文字のアルファベットを組合せた入力の場合，MD5 は他のハッシュ関数にくらべ分散値が大きくなり，出力に偏りがあるのを確認した．

参考文献

[1] 電子情報通信学会，“情報セキュリティハンドブック,” オーム社，2004.

[2] ブルース・シュナイアー，山形浩生，“暗号技術大全集,” ソフトバンクパブリッシング株式会社，2005.