■1 群（信号・システム） - 3 編（暗号理論）

4 章 ハッシュ関数

（執筆者：廣瀬勝一）

■概要■

【本章の構成】

■1 群 - 3 編 - 4 章

4-1 性　質

（執筆者：廣瀬勝一）[2008 年 12 月 受領]

ハッシュ関数（hash function）は，任意長あるいは実用上十分な長さまでの 2 値系列を固定長の 2 値系列に変換する関数である．更に，ハッシュ関数 H は通常，以下の性質を満たす関数であると定義される．

原像計算困難性（preimage resistance）

与えられた出力 v について，$H(M) = v$ を満たす M を計算するのが困難である．

第二原像計算困難性（second-preimage resistance）

与えられた入力 M について，$H(M) = H(M')$ かつ $M \neq M'$ を満たす M' を計算するのが困難である．

衝突計算困難性（collision resistance）

$H(M) = H(M')$ を満たす相異なる M, M' を計算するのが困難である．

原像計算困難性と第二原像計算困難性はともに，定められた一つの出力に対応する入力の計算困難性であり，これらはハッシュ関数の一方向性を表している．

ハッシュ関数の入力を無作為に選択して出力を計算することを繰り返すとき，ハッシュ関数の出力長を n とすると，（第二）原像計算困難性について，定められた出力に対応する入力の計算時間は $O(2^n)$ である．一方，衝突計算困難性について，衝突，すなわち同じ出力に対応する相異なる入力の組の計算時間は $O(2^{n/2})$ である．このように，（第二）原像計算困難性に関する計算時間と衝突計算困難性に関する計算時間の違いが感覚に反して大きく異なることは，誕生日のパラドクス（birthday paradox）と呼ばれる．

■1 群 - 3 編 - 4 章

4-2 構　造

（執筆者：廣瀬勝一）［2008 年 12 月 受領］

4-2-1 反復形ハッシュ関数

ハッシュ関数 $H:\{0,1\}^* \to \{0,1\}^n$ は，通常，入出力共に固定長の圧縮関数 $h:\{0,1\}^n \times \{0,1\}^b \to \{0,1\}^n$ を用いて構成される．ハッシュ関数 H の入力 M はまず，パディングと呼ばれる処理により b の倍数の長さの系列に変換され，b ビットのブロック $M_1, M_2, \ldots, M_N$ に分割される．次に，$1 \le i \le N$ について，連鎖値 $v_i = h(v_{i-1}, M_i)$ が順次計算される．なお，$v_0 = IV$ はあらかじめ定められた初期値である．v_N が出力 $H(M)$ となる．このようにして構成されるハッシュ関数は反復形ハッシュ関数（iterated hash function）と呼ばれる．これを図 4･1 に示す．

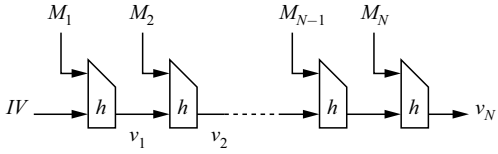

図 4･1　反復形ハッシュ関数 H．h は圧縮関数，IV は初期値である．

パディングでは，通常，M の後に 10^k を付加し，その後に更に，M の長さ $|M|$ の w ビット 2 進数表記が付加される．ここで，0^k は k 個の 0 の連接を表し，k は $|M|+1+k+w$ が b の倍数となるような最小の非負整数である．なお，このとき，入力 M の最大長は 2^w-1 となる．このように，M の長さを付加するパディングはマークル・ダンガード強化法（Merkle-Damgård strengthening）と呼ばれる．

反復形ハッシュ関数の特長は，圧縮関数 h が衝突計算困難性を満たせば，ハッシュ関数 H も衝突計算困難性を満たすことである．これは次のように対偶を考えることにより確認できる．まず，反復形ハッシュ関数 H について，衝突，すなわち，$H(M)=H(M')$ かつ $M \neq M'$ を満たす M, M' が得られたと仮定する．このとき，図 4･1 に示すような $H(M), H(M')$ それぞれの計算を考えると，まず，M, M' の長さが異なる場合は，マークル・ダンガード強化法により，$H(M)$ と $H(M')$ の計算の最終段の h で衝突が生じていることになる．一方，M, M' の長さが等しい場合も，$H(M)$ と $H(M')$ の計算を出力側から遡行することにより h の衝突を得ることができる．

圧縮関数の構成法は，専用構成法，ブロック暗号を利用した構成法，剰余系演算を利用した構成法に分類される．専用構成法によるハッシュ関数の例として，MD4 [10], MD5 [11], RIPEMD-160 [3], Secure Hash Standard （SHS）[8] のアルゴリズムがあげられる．SHS のアルゴリズムについては次節に述べる．ブロック暗号を利用した構成法は，更に，ブロック暗号のブロック長に等しい出力長の圧縮関数を構成する単ブロックモードと，ブロック長の 2 倍に等しい出力長の圧縮関数を構成する倍ブロックモードに分類される．単ブロックモードの例として，マティアス・メイヤ・オーシーズ（Matyas-Meyer-Oseas）法 [4]，デイビス・メイヤ（Davies-Meyer）法，宮口・プレネール（Miyaguchi-Preneel）法があげられる．また，

これらを含むモデルがプレネールらにより提案されている [9]．倍ブロックモードの例として，MDC-2 [4], MDC-4 があげられる．剰余系演算を利用した構成法によるハッシュ関数の例としては，MASH-1, MASH-2 [5]があげられる．

4-2-2 反復形ハッシュ関数に対する攻撃

(1) 伸長攻撃

伸長攻撃（length-extension attack）は，反復形ハッシュ関数 H について，$H(M\|M')$ が $H(M)$ と M' のみから計算できるという性質を利用する攻撃である．ここで，$M\|M'$ は M と M' との連接を表す．

ハッシュ関数 H にぜい弱性がない場合，秘密鍵 K について，$H(K\|\cdot)$ は擬似ランダム関数，すなわち，K が秘密である限り真のランダム関数と識別不能な関数となる．しかし，H が反復形ハッシュ関数である場合，$H(K\|M\|M')$ は，K を用いることなく，$H(K\|M)$ と M' のみから計算可能である．このことから，$H(K\|\cdot)$ は擬似ランダム関数でないことが分かる．

(2) 多衝突攻撃

T 衝突攻撃は，ハッシュ関数について，同じ出力に対応する T 個の入力を得ることを目的とする攻撃である．ハッシュ関数にぜい弱性がない場合，T 衝突攻撃の計算時間は $O((T!)^{\frac{1}{T}} 2^{\frac{T-1}{T}n})$ である．一方，反復形ハッシュ関数 H について，計算時間 $O((\log_2 T)2^{n/2})$ の T 衝突攻撃が提案されている[6]．この攻撃法は以下のとおりである．$t = \lceil \log_2 T \rceil$ とする．$1 \le i \le t$ について順に，圧縮関数 h の衝突，すなわち，$h(v_{i-1}, M_i^{(0)}) = h(v_{i-1}, M_i^{(1)})$ を満たす相異なる $M_i^{(0)}$, $M_i^{(1)}$ を計算する．各 i について，この計算時間は $O(2^{n/2})$ である．任意の $b_i \in \{0, 1\}$ について，$H(M_1^{(b_1)}\|M_2^{(b_2)}\|\cdots\|M_t^{(b_t)})$ はすべて等しい値をとるので，同じ出力に対応する 2^t 個の入力が得られたことになる．

■1 群 - 3 編 - 4 章

4-3 Secure Hash Standard（SHS）

（執筆者：廣瀬勝一）［2008 年 12 月 受領］

4-3-1 概　要

Secure Hash Standard（SHS）は, 米国立標準技術研究所（NIST: National Institute of Standards and Technology）の FIPS PUB 180-3 [8] に規定されている米国標準のハッシュアルゴリズムであり，五つのハッシュ関数 SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 からなる．これらは，リベスト（Rivest）が提案した MD4, MD5 に基づいて，米国家安全保障局（NSA: National Security Agency）により設計されたアルゴリズムである．

SHS の履歴は以下のとおりである．まず，1993 年に SHA（Secure Hash Algorithm）が FIPS PUB 180 に規定された．このアルゴリズムはしばしば SHA-0 と呼ばれる．その後，SHA-0 のアルゴリズムに若干の修正が加えられ，1995 年に SHA-1 が FIPS PUB 180-1 に規定された．2002 年，SHA-1 と共に新たに SHA-256, SHA-384, SHA-512 が FIPS PUB 180-2 に規定された．2004 年には，FIPS PUB 180-2 の変更通知として，SHA-224 が付加された．FIPS PUB 180-3 は FIPS PUB 180-2 の改訂版であり，SHA-224 の記述が本文に取り込まれた．また，攻撃法の進展により安全性の記述が削除された．

SHA-1, SHA-224, SHA-256 の入力長は $2^{64}-1$ ビット以下であり，SHA-384, SHA-512 の入力長は $2^{128}-1$ ビット以下である．また，SHA-1 の出力長は 160 ビットであり，SHA-224/256/384/512 については末尾の整数が出力ビット長を表す．

2004 年に王（Wang）らにより MD4, MD5 などに対する非常に強力な衝突攻撃が公表されて以来[12]，その改良及び SHS に対する適用の研究が活発に行われてきた．その結果，MD4, MD5, SHA-0 については，通常のパーソナルコンピュータでもたかだか 1 時間程度の計算で衝突の得られることが報告されている．また，SHA-1 については，およそ 2^{63} 回の圧縮関数の計算に相当する時間で衝突が得られると見積もられている．一方，SHA-224/256/384/512 については，安全性を脅かすような衝突攻撃は報告されていない．このため，NIST は，ディジタル署名など衝突計算困難性を要求する応用に関して，SHA-224/256/384/512 への早急な移行を推奨し，特に，米国政府機関に対しては 2010 年末までの SHA-1 の使用停止を勧告している．

4-3-2 SHA-256

SHA-256 では，まず，ℓ ビットのメッセージ M に対して，マークル・ダンガード強化法によるパディングが行われる．この処理ではまず，M の末尾に 10^k を付加する．k は $\ell+1+k \equiv 448 \pmod{512}$ を満たす最小の非負整数である．次に，その後ろに ℓ の 64 ビット 2 進数表記を付加する．パディング後の系列長は 512 の倍数である．この系列は，長さが 512 ビットの N 個のブロック $M_1, M_2, \ldots, M_N$ に分割される．

以下では，x, y, z を 32 ビットの 2 値系列とし，$\wedge, \oplus, \neg$ をそれぞれ，ビットごとの論理積，排他的論理和，否定とする．次の関数を定義する．

$$\mathrm{Ch}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) \tag{4・1}$$

$$\mathrm{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \tag{4・2}$$

更に，$\mathrm{ROTR}^b(x)$ を x の b ビット右巡回シフト，$\mathrm{SHR}^b(x)$ を x の b ビット右シフトとし，以下の関数を定義する．

$$\Sigma_0(x) = \mathrm{ROTR}^2(x) \oplus \mathrm{ROTR}^{13}(x) \oplus \mathrm{ROTR}^{22}(x) \tag{4·3}$$

$$\Sigma_1(x) = \mathrm{ROTR}^6(x) \oplus \mathrm{ROTR}^{11}(x) \oplus \mathrm{ROTR}^{25}(x) \tag{4·4}$$

$$\sigma_0(x) = \mathrm{ROTR}^7(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{SHR}^3(x) \tag{4·5}$$

$$\sigma_1(x) = \mathrm{ROTR}^{17}(x) \oplus \mathrm{ROTR}^{19}(x) \oplus \mathrm{SHR}^{10}(x) \tag{4·6}$$

メッセージブロック M_i，連鎖値 v_{i-1} を入力とする SHA-256 の圧縮関数の処理は以下のとおりである．なお，+ は 2^{32} を法とする加算を表す．

1. M_i を 16 個の 32 ビットのブロック $M_{i,0}, M_{i,1}, \ldots, M_{i,15}$ に分割し，以下に従い，$W_0, W_1, \ldots, W_{63}$ を計算する．

$$W_j = \begin{cases} M_{i,j} & 0 \le j \le 15 \\ \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16} & 16 \le j \le 63 \ . \end{cases} \tag{4·7}$$

2. v_{i-1} を 8 個の 32 ビットのブロック $v_{i-1,0}, v_{i-1,1}, \ldots, v_{i-1,7}$ に分割し，$0 \le l \le 7$ について，$a_l = v_{i-1,l}$ とする．

3. $0 \le j \le 63$ について，以下を計算する．

$$T_1 = a_7 + \Sigma_1(a_4) + \mathrm{Ch}(a_4, a_5, a_6) + K_j + W_j \tag{4·8}$$

$$T_2 = \Sigma_0(a_0) + \mathrm{Maj}(a_0, a_1, a_2) \tag{4·9}$$

$$a_7 = a_6 \qquad a_6 = a_5 \qquad a_5 = a_4 \qquad a_4 = a_3 + T_1 \tag{4·10}$$

$$a_3 = a_2 \qquad a_2 = a_1 \qquad a_1 = a_0 \qquad a_0 = t_1 + T_2 \tag{4·11}$$

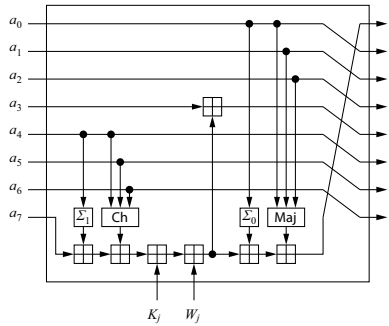

図 4·2　SHA-256 のステップ関数

各 j に関するこの計算を図で表すと図 4･2 のようになる．なお，K_j は小さい方から $(j+1)$ 番目の素数の立方根の小数部上位 32 ビットである．

4. 圧縮関数の出力 $v_j = v_{j,0}\|v_{j,1}\|\cdots\|v_{j,7}$ を以下のように定める．

$$v_{j,l} = a_l + v_{j-1,l} \qquad (0 \le l \le 7). \tag{4･12}$$

初期値 $IV = IV_0\|IV_1\|\cdots\|IV_7$ は 16 進数表記で以下のとおりである．

IV_0 = 6a09e667　IV_1 = bb67ae85　IV_2 = 3c6ef372　IV_3 = a54ff53a
IV_4 = 510e527f　IV_5 = 9b05688c　IV_6 = 1f83d9ab　IV_7 = 5be0cd19

■1 群 - 3 編 - 4 章

4-4 HMAC

（執筆者：廣瀬勝一）［2008 年 12 月 受領］

HMAC[2]はハッシュ関数を用いて構成されるメッセージ認証コード（MAC: Message Authentication Code）関数であり，NIST の FIPS PUB 198-1 [7] に規定されている．HMAC はまた擬似ランダム関数として利用される．

ハッシュ関数を H，秘密鍵を K とし，メッセージを M とするとき，HMAC は以下のように定義される．

$$H((K \oplus \mathtt{opad}) \| H((K \oplus \mathtt{ipad}) \| M)) \ . \tag{4･13}$$

HMAC では通常，H は反復形ハッシュ関数であると仮定される．H の圧縮関数のメッセージブロックの入力長を B バイトとするとき，ipad はバイト 36 の B 回の繰り返し，opad はバイト 5c の B 回の繰り返しである．また，K の長さは B バイト以下であり，B バイト未満の場合は，長さが B バイトになるまで末尾に 0 が付加される．圧縮関数が擬似ランダム関数であれば，HMAC も擬似ランダム関数であることが示されている[1]．

■参考文献

1) M. Bellare, “New proofs for NMAC and HMAC: Security without collision-resistance,” CRYPTO 2006, Lecture Notes in Computer Science 4117, pp.602-619, 2006.
2) M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” CRYPTO ’96, Lecture Notes in Computer Science 1109, pp.1-15, 1996.
3) H. Dobbertin, A. Bosselaers, and B. Preneel, “RIPEMD-160: A strengthened version of RIPEMD,” FSE ’96, Lecture Notes in Computer Science 1039, pp.71-82, 1996.
4) ISO/IEC 10118-2, “Information technology – security techniques – hash-functions – part 2: Hash-functions using an *n*-bit block cipher,” 2000.
5) ISO/IEC 10118-4, “Information technology – security techniques – hash-functions – part 4: Hash-functions using modular arithmetic,” 1998.
6) A. Joux, “Multicollisions in iterated hash functions. Application to cascaded constructions,” CRYPTO 2004, Lecture Notes in Computer Science 3152, pp.306-316, 2004.
7) National Institute of Standards and Technology（NIST）, “The keyed-hash message authentication code（HMAC）,” Federal Information Processing Standards Publication, vol.198, no.1, 2008.
8) National Institute of Standards and Technology（NIST）, “Secure hash standard（SHS）,” Federal Information Processing Standards Publication, vol.180, no.3, 2008.
9) B. Preneel, R. Govaerts, and J. Vandewalle, “Hash functions based on block ciphers: A synthetic approach,” CRYPTO ’93, Lecture Notes in Computer Science 773, pp.368-378, 1994.
10) R. Rivest: “The MD4 message-digest algorithm,” Request for Comments 1321（RFC 1320）, The Internet Engineering Task Force, 1992.
11) R. Rivest: “The MD5 message-digest algorithm”, Request for Comments 1321（RFC 1321）, The Internet Engineering Task Force, 1992.
12) X. Wang, D. Feng, X. Lai, and H. Yu, “Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD,” Cryptology ePrint Archive, Report 2004/199, 2004. http://eprint.iacr.org/.