暗号ハッシュ関数

廣瀬勝一

暗号ハッシュ関数の分類

暗号ハッシュ関数 (Cryptographic Hash Functions)

任意長の入力系列を固定長の出力系列に変換する関数

$$H : \{0,1\}^* \to \{0,1\}^\ell$$

- 鍵なしハッシュ関数 (Unkeyed hash function)
  - 改ざん検知コード (MDC: Manipulation Detection Code)
- 鍵つきハッシュ関数 (Keyed hash function)
  - メッセージ認証コード (MAC: Message Authentication Code)

鍵なしハッシュ関数の応用と性質

暗号方式で最も良く用いられる構成要素

- ディジタル署名のためのメッセージダイジェスト
- 公開鍵暗号の平文の前処理（OAEP など）
- メッセージ認証
- ハッシュ木（ディジタル署名や時刻印のための）
- 共通鍵暗号
- …

性質

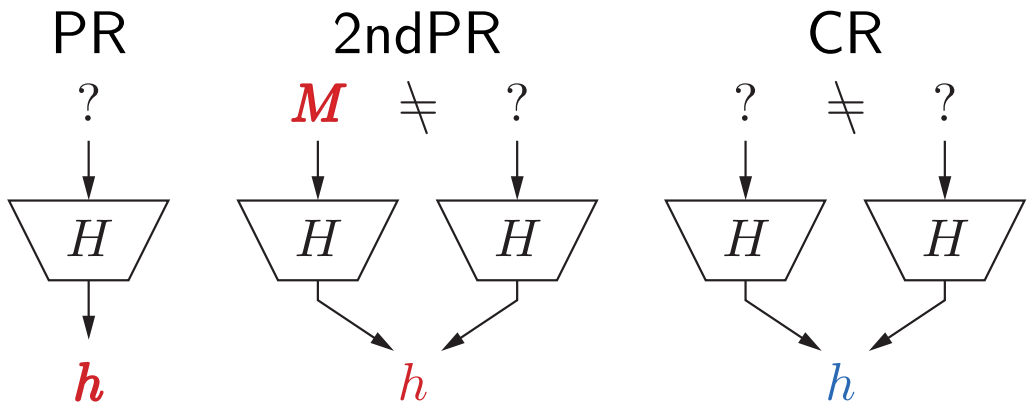
- 一方向性 (One-wayness)
  - 原像計算困難性 (Preimage resistance)
  - 第二原像計算困難性 (Second-preimage resistance)
- 衝突計算困難性 (Collision resistance)

性質

原像計算困難性 (Preimage resistance) 与えられた出力 h について，
$H(M) = h$ なる入力 M の計算が困難

第二原像計算困難性 (Second-preimage resistance) 与えられた入力 M について，$M \neq M'$ かつ $H(M') = H(M)$ を満たす M' の計算が困難

衝突計算困難性 (Collision resistance) $M \neq M'$ かつ $H(M) = H(M')$ を満たす入力の組 M, M' の計算が困難



性質	PR	2ndPR	CR
攻撃計算量	$O(2^{\ell})$	$O(2^{\ell})$	$O(2^{\ell/2})$

所望の結果が得られるまで，入力を選択して出力の計算を繰返す場合
（ハッシュ関数の内部構造を一切利用しない場合）

誕生日のパラドクス

N 個の要素から無作為に 1 個を選択する試行を繰り返す

q 回の試行で 2 回以上選択される要素の存在する確率を P とすると

$$1 - P = \frac{N-1}{N} \cdot \frac{N-2}{N} \cdots \frac{N-(q-1)}{N} = \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$$

x が小さいとき

$$\mathrm{e}^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots \approx 1 - x$$

したがって

$$1 - P \approx \prod_{i=1}^{q-1} \mathrm{e}^{-\frac{i}{N}} = \mathrm{e}^{-\sum_{i=1}^{q-1} \frac{i}{N}} = \mathrm{e}^{-\frac{q(q-1)}{2N}} \approx \mathrm{e}^{-\frac{q^2}{2N}}$$

$$P \approx 1 - \mathrm{e}^{-\frac{q^2}{2N}}$$

誕生日のパラドクス

さらに

$$\mathrm{e}^{-\frac{q^2}{2N}} \approx 1 - P$$

$$-\frac{q^2}{2N} \approx \ln(1 - P)$$

$$q \approx \sqrt{2N \ln \frac{1}{1 - P}}$$

$P = 1/2$ とすると $q \approx \sqrt{(2 \ln 2)\, N} \approx 1.17\sqrt{N}$

（約 $1.17\sqrt{N}$ 回の試行で 2 回以上選択される要素の存在する確率は $1/2$）

【例】23 人集まれば，誕生日の同じ人が存在する確率は $1/2$

23 人はあまりに少なく感じられるのでパラドクスと呼ばれる

誕生日攻撃

ハッシュ関数の衝突を見つける自明な攻撃
ハッシュ関数の内部の構造は一切利用しない

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
ハッシュ関数の出力長を ℓ ビットとすると
およそ $1.17 \times 2^{\ell/2}$ 回の計算で衝突の生じる確率は $1/2$

ハッシュ関数の出力長は 160 ビット以上でなければならない
現在は 256 ビット以上とすることが推奨されている

ハッシュ関数の構成

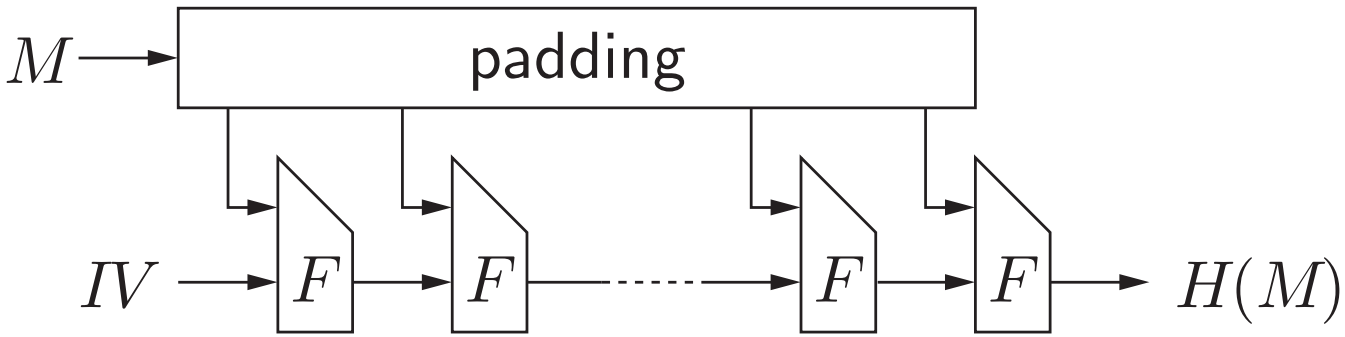
ハッシュ関数 = 圧縮関数 + 定義域拡大

圧縮関数 (compression function) 固定長入出力で，入力長 > 出力長

定義域拡大 圧縮関数による任意長入力の処理法

反復型ハッシュ関数 (iterated hash function)

- 圧縮関数 $F : \{0,1\}^n \times \{0,1\}^b \to \{0,1\}^n$
- 初期値 $IV \in \{0,1\}^n$
- パディング　入力を b の倍数の長さの系列に変換

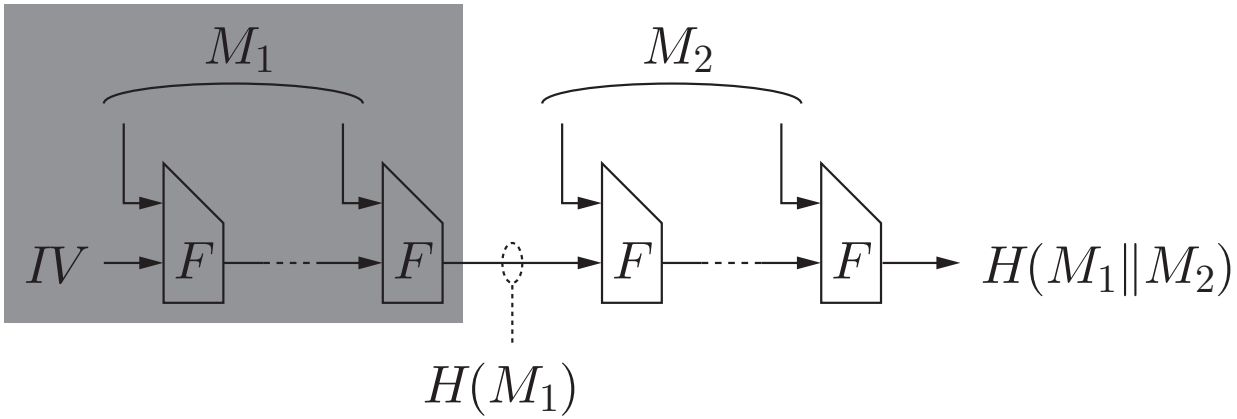


反復型ハッシュ関数

利点 圧縮関数 F が CR $\Rightarrow$ ハッシュ関数 H は CR

欠点 Length-Extension

$H(M_1 \| M_2)$ は $H(M_1)$ と M_2 から計算できる．M_1 は不要.



パディング (padding)

入力 $M = (M_1, M_2, \ldots, M_m)$, $M_i \in \{0,1\}^b$

- 簡易な曖昧さのない方法

M_1	$\cdots$	M_{m-1}	$M_m 10 \cdots 0$

- MD-strengthening (named after Merkle & Damgård)

M_1	$\cdots$	M_{m-1}	$M_m 00 \cdots 0$	$\lvert M \rvert$

圧縮関数の構成法

- 専用構成法
  - MDx 族
    MD4, MD5; RIPEMD-160;
    SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
- ブロック暗号を用いた構成法
  - 単ブロック長 (single block length)
    Preneel-Govaerts-Vandewalle のモデル
  - 倍ブロック長 (double block length)
    MDC-2, MDC-4, abreast/tandem Davies-Meyer

Secure Hash Standard (SHS) の変遷

FIPS 180 (Federal Information Processing Standards) (1993 年 5 月)

- SHA (Secure Hash Algorithm, SHA-0 とも呼ばれる)

FIPS 180-1 (1995 年 4 月)

- SHA-1 (メッセージ拡大に 1 ビット左巡回シフトを付加)

FIPS 180-2 (2002 年 8 月)

- SHA-1, SHA-256/384/512

FIPS 180-2, Change Notice (2004 年 2 月)

- SHA-224

FIPS 180-3 (2008 年 10 月)

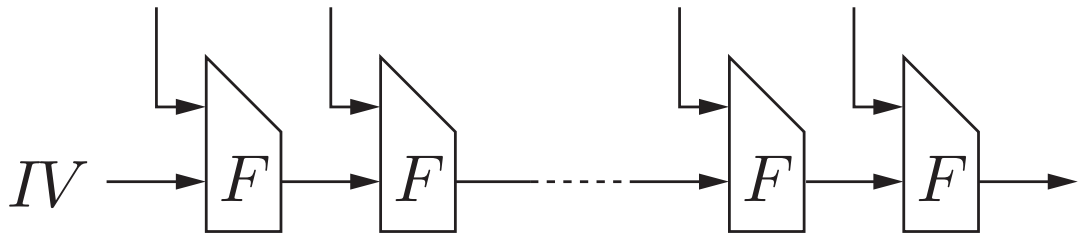
- SHA-1, SHA-224/256/384/512

Secure Hash Standard (SHS)

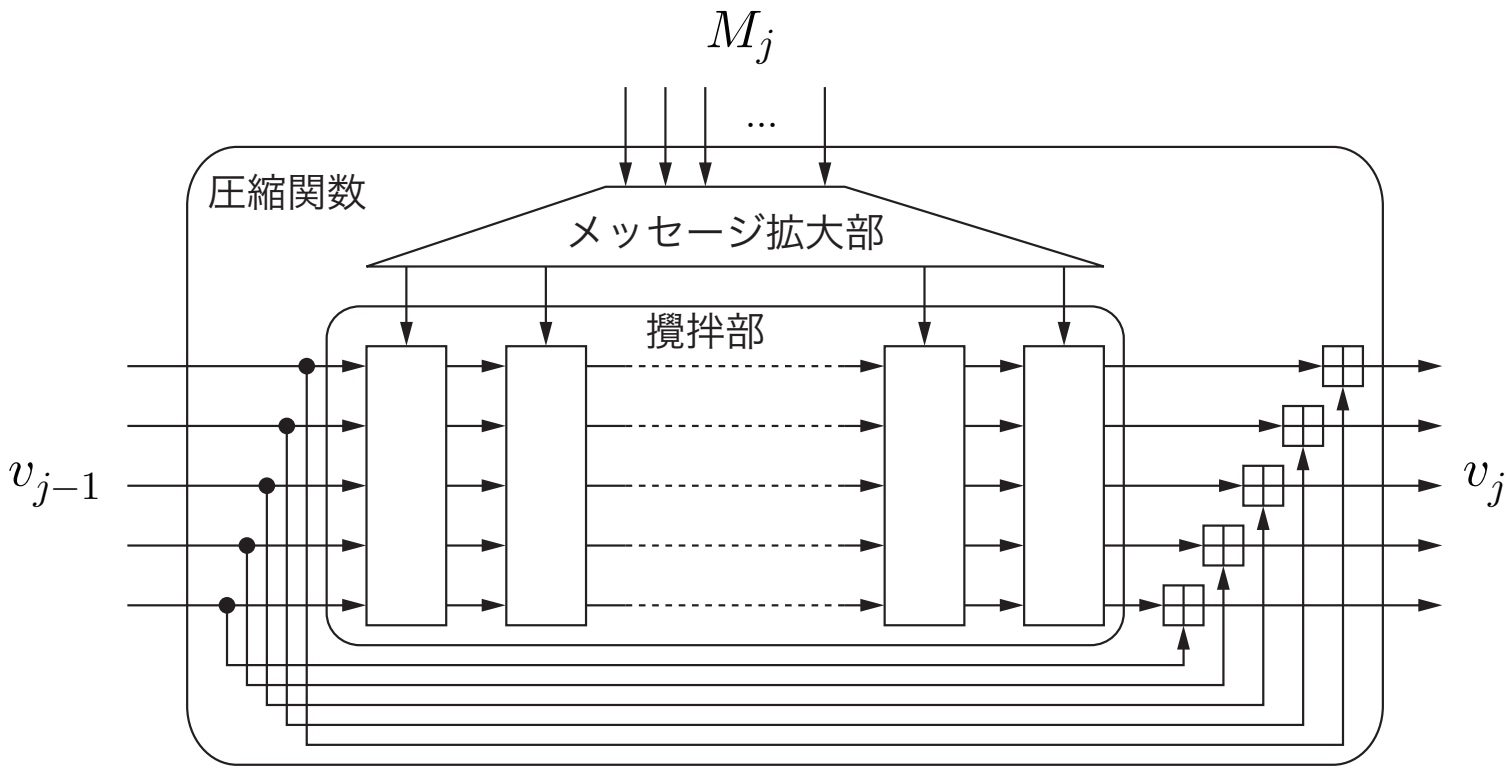
アルゴリズム	入力長	ブロック長	ワード長	出力長
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512

長さの単位はビット．ブロック長は圧縮関数のメッセージブロック長.

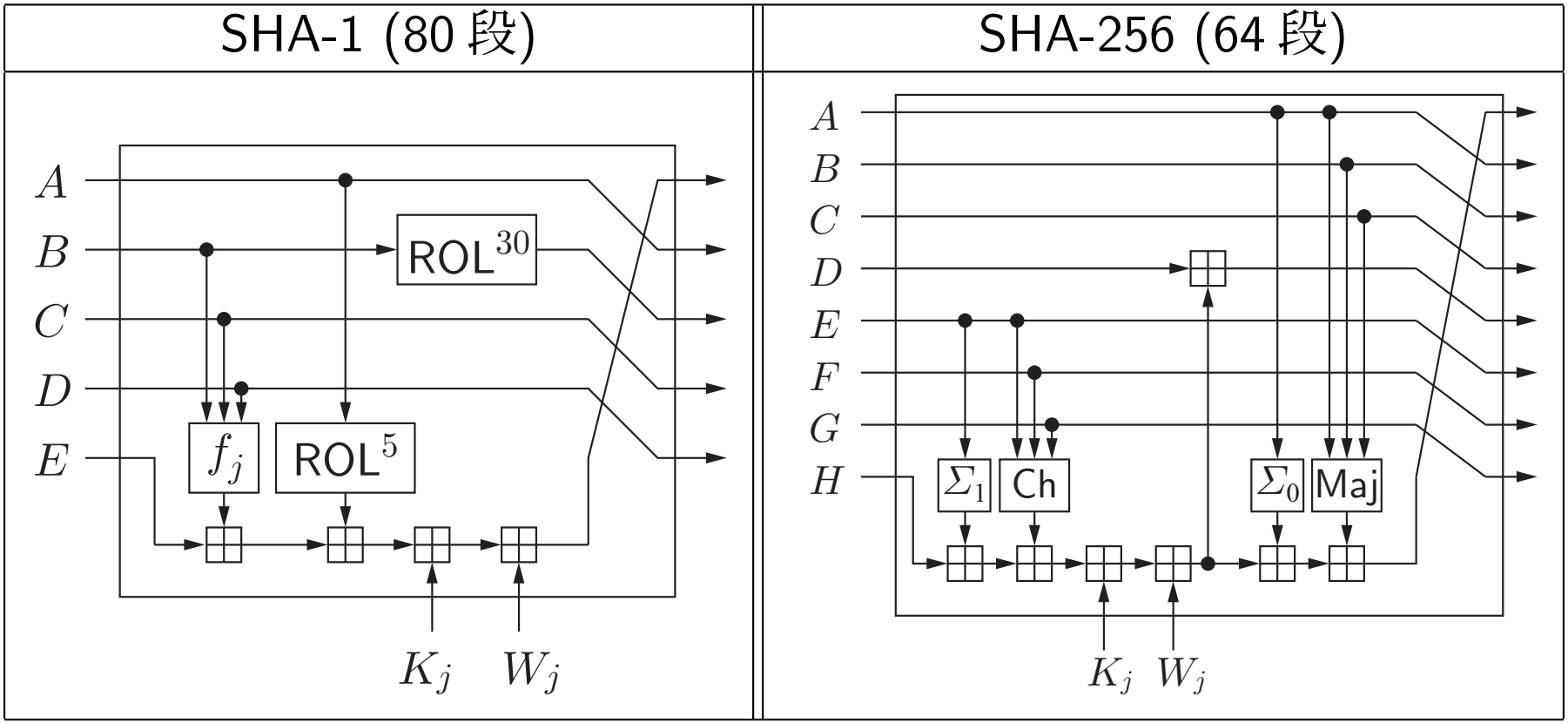
- SHA-256 と SHA-224 の相違は，初期値と出力の切り捨てのみ.
- SHA-512 と SHA-384 の相違も同様.



SHS の圧縮関数の概略



圧縮関数の攪拌部



K_j は定数

f_j は 20 段ごとに，Ch, Parity, Maj, Parity

$$\Sigma_0(x) = \mathrm{ROR}^2(x) \oplus \mathrm{ROR}^{13}(x) \oplus \mathrm{ROR}^{22}(x)$$

$$\Sigma_1(x) = \mathrm{ROR}^6(x) \oplus \mathrm{ROR}^{11}(x) \oplus \mathrm{ROR}^{25}(x)$$

SHA-1 について

f_j はビットごとの演算

$$f_j(u,v,w) = \begin{cases} \mathsf{Ch}(u,v,w) = u\,v \vee \overline{u}\,w & (0 \le j \le 19) \\ \mathsf{Parity}(u,v,w) = u \oplus v \oplus w & (20 \le j \le 39) \\ \mathsf{Maj}(u,v,w) = u\,v \vee u\,w \vee v\,w & (40 \le j \le 59) \\ \mathsf{Parity}(u,v,w) & (60 \le j \le 79) \end{cases}$$

$$K_j = \begin{cases} \texttt{5a827999} & \text{for } 0 \le j \le 19 \\ \texttt{6ed9eba1} & \text{for } 20 \le j \le 39 \\ \texttt{8f1bbcdc} & \text{for } 40 \le j \le 59 \\ \texttt{ca62c1d6} & \text{for } 60 \le j \le 79 \end{cases}$$

初期値

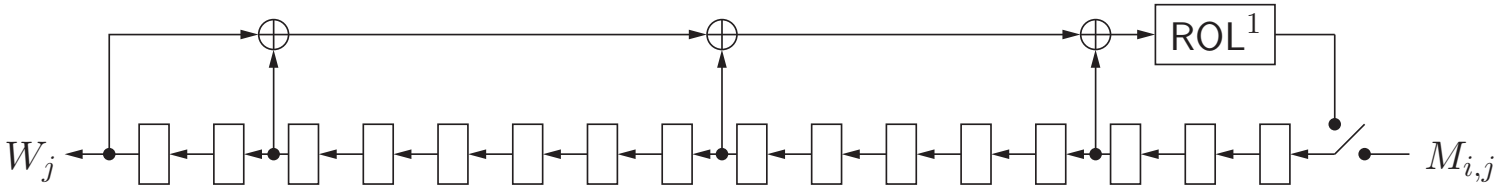
```
67452301  efcdab89  98badcfe  10325476  c3d2e1f0
```

圧縮関数のメッセージ拡大

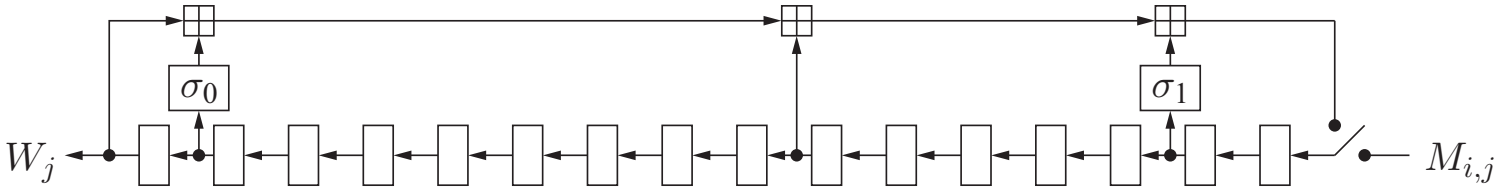
入力 $M_i = (M_{i,0}, M_{i,1}, \ldots, M_{i,15})$, $M_{i,j} \in \{0,1\}^{32}$

$$(W_0, W_1, \ldots, W_r) \leftarrow (M_{i,0}, M_{i,1}, \ldots, M_{i,15}) \quad W_j \in \{0,1\}^{32}$$

SHA-1



SHA-256

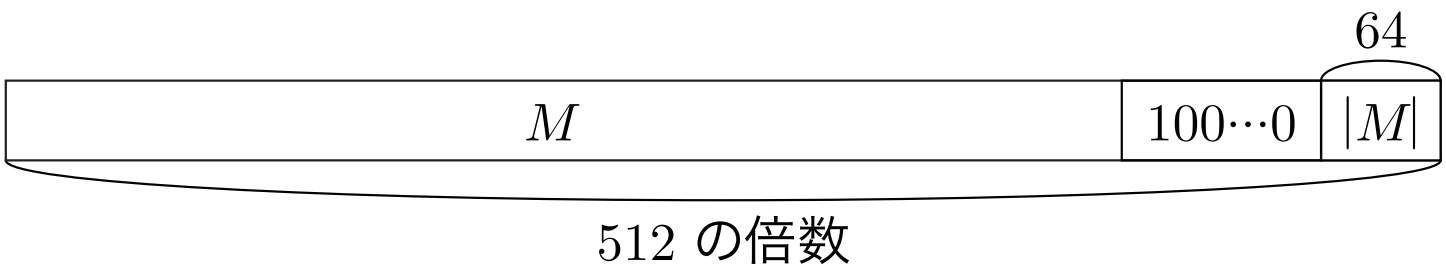


$$\sigma_0(x) = \mathsf{ROR}^7(x) \oplus \mathsf{ROR}^{18}(x) \oplus \mathsf{SHR}^3(x)$$
$$\sigma_1(x) = \mathsf{ROR}^{17}(x) \oplus \mathsf{ROR}^{19}(x) \oplus \mathsf{SHR}^{10}(x)$$

SHA-1, SHA-224/256 のパディング

入力 M のパディング

1. $z = M \| 10^d$ とする.
   d は $|M| + 1 + d + 64$ が 512 の倍数となる最小の非負整数.
2. $z = z \| \alpha$ とする.
   α は $|M|$ の 2 進数表現で $|\alpha| = 64$.



SHA-0/1 に対する強力な衝突攻撃

ハッシュ関数 H に対する衝突攻撃

$H(M) = H(M')$ を満たす相異なる M, M' を得ようとする攻撃

Wang, et. al. (1997, 1998, 2004–)

衝突攻撃の計算量 (単位は圧縮関数の計算回数)

SHA-0	$\lesssim 2^{33}$	
SHA-1	$\lesssim 2^{63}$	← 衝突はまだ得られていない.

SHA-224/256/384/512 に対して有効な攻撃法は発見されていない.

SHS の現状

NIST's Policy on Hash Functions (3/15/2006)

http://csrc.nist.gov/groups/ST/hash/policy.html

米国政府機関に対し

- SHA-2 (SHA-224/256/384/512) への早急な移行を推奨.
- 衝突計算困難性を要求する応用に関して，2010 年末までの SHA-1 の使用停止を勧告.
  - ディジタル署名，タイムスタンプなど
- 以下に限り，SHA-1 の使用継続を容認
  - メッセージ認証，鍵導出，擬似乱数生成

NIST Cryptographic Hash Algorithm Competition

`http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`

- 公募要項案に対するコメントの募集 (2007 年 1 月 23 日)
- 公募開始 (2007 年 11 月 2 日)
- 締切 (2008 年 10 月 31 日)

最小必須要件

- 特許権，知的財産権等の制約なく利用可能であること.
- 多様なハードウェア・ソフトウェアで実装可能であること.
- 入出力長について以下の要件を満たすこと.
  - 出力長：224, 256, 384, 512 ビットのサポート
  - 最小入力長 $\geq 2^{64} - 1$

NIST Hash Competition: 安全性要件

必須

- 応用の安全性の保証
  - ディジタル署名 (FIPS 186-2)
  - 鍵導出 (NIST SP 800-56A)
  - HMAC (FIPS 198)
  - DRBG (NIST SP 800-90)
  - …
- ランダム化ハッシュモードの安全性
- 衝突計算困難性，(第二) 原像計算困難性
- Length-extension 攻撃に対する安全性

オプション

- HMAC 以外の擬似ランダム関数モードの提供
- Joux 多衝突攻撃，Kelsey-Schneier 第二原像攻撃への対策

NIST Hash Competition: 安全性の定量的要件

要件	度合い
HMAC	$n/2$
ランダム化ハッシュ	$n-k$
衝突計算困難性	$n/2$
原像計算困難性	n
第二原像計算困難性	$n-k$

- 度合い s は，攻撃計算量 $\ll 2^s$ とならないことを表す.
- k は，与えられるメッセージ長が 2^k ビットであることを表す.

NIST Hash Competition: 推移

- 応募総数 64 件 (2008/10/31)
- ラウンド 1 候補 (51 件) の公開 (2008/12/10)
- ラウンド 2 候補 (14 件) を選出 (2009/07/24)
- ラウンド 3 候補 (5 件) を選出 (2010/12/09)
- winner を選出 (2012/10/02)

ラウンド 3 候補（ファイナリスト）

- BLAKE (CHE)
- Grøstl (DNK)
- JH (SGP)
- Keccak (CHE) ← winner!
- Skein (USA)

日本からの提案

- AURORA（ソニー，名古屋大）ラウンド 1
- Lesamnta（日立，神戸大，福井大）ラウンド 1
- Luffa（日立）ラウンド 2

反復型ハッシュ関数の衝突計算困難性

圧縮関数 $F : \{0,1\}^{n+t} \to \{0,1\}^n$ について $t \geq 2$ とする

ハッシュ関数 $H : \{0,1\}^* \to \{0,1\}^n$ は以下のように定義される

H の入力 x について

1. $|x| > 0$ のとき，$x = (x_1, x_2, \ldots, x_m)$ とする．ここで

$$|x_i| = \begin{cases} t-1 & \text{for } 1 \leq i \leq m-1 \\ t-1-d & \text{for } i = m \text{ and } 0 \leq d \leq t-2 \end{cases}$$

$$z_i = \begin{cases} x_i & \text{for } 1 \leq i \leq m-1 \\ x_i \| 0^d & \text{for } i = m \\ d \text{ の 2 進数表現} & \text{for } i = m+1 \end{cases}$$

2. $|x| = 0$ のとき，$m = 0$ とし，

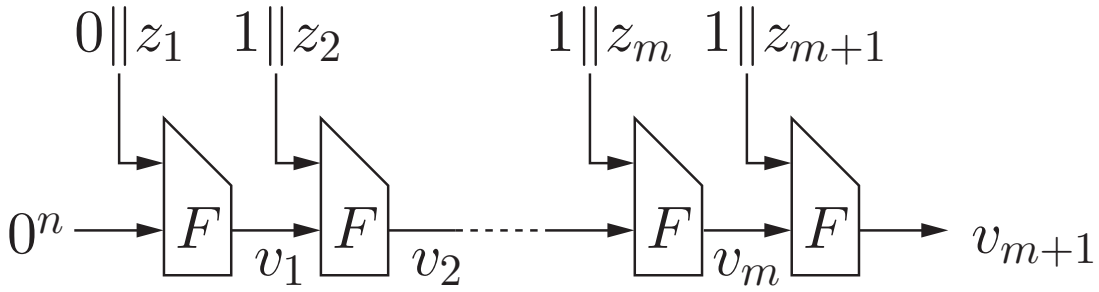
$$z_1 = 0^{t-1} \quad (0 \text{ の 2 進数表現})$$

反復型ハッシュ関数の衝突計算困難性

$H(x)$ の計算は以下の通り

$$v_1 = F(0^n \| 0 \| z_1) \ , \qquad v_i = F(v_{i-1} \| 1 \| z_i) \text{ for } 2 \leq i \leq m+1$$

$$H(x) = v_{m+1}$$



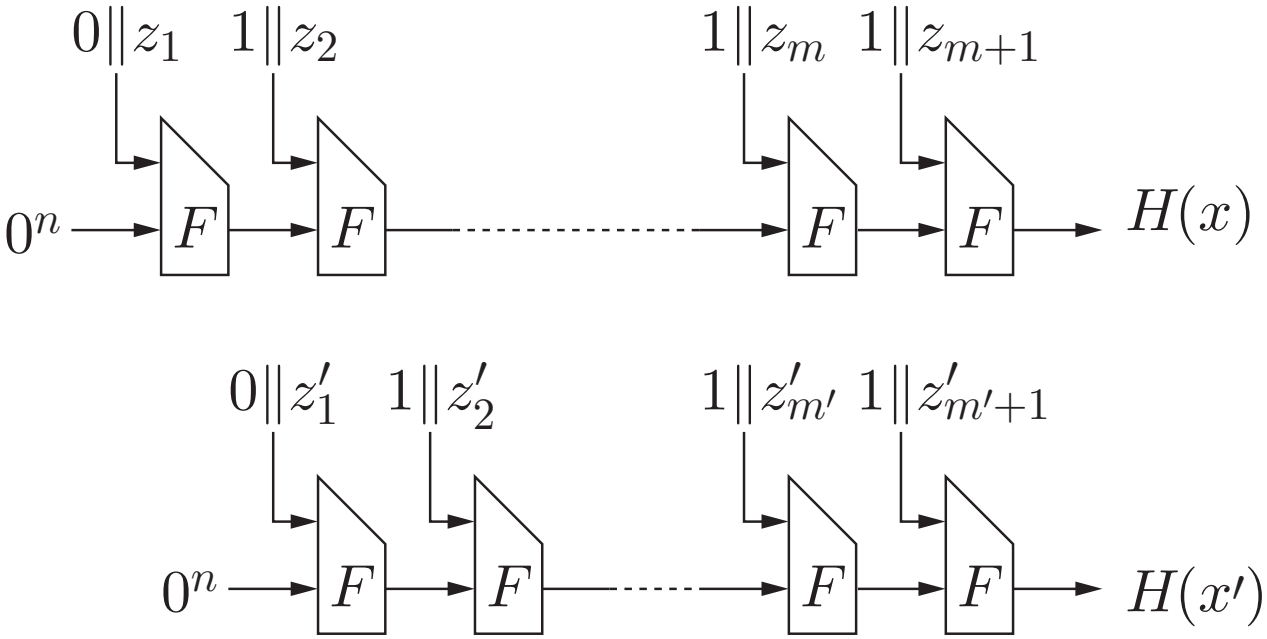
【定理 1】F が衝突計算困難 $\Rightarrow$ H が衝突計算困難

定理 1 の証明 (1/3)

H の衝突が計算できれば F の衝突が計算できることを示す.

$x \neq x'$, $H(x) = H(x')$, $|x| \geq |x'|$ とする.

$(z_1, \ldots, z_{m+1}), (z'_1, \ldots, z'_{m'+1})$ を各々x, x' のパディング後の入力とする.



定理 1 の証明 (2/3)

$|x'| = 0$ のとき，$|x| > 0$ である．したがって

$$F(v_m \| 1 \| z_{m+1}) = F(0^n \| 0 \| z'_1)$$

であり，これは F の衝突である．

定理 1 の証明 (3/3)

$|x'| > 0$ のとき

Case 1. $|x| \not\equiv |x'| \pmod{t-1}$ のとき
$z_{m+1} \neq z'_{m'+1}$ かつ $F(v_m \| 1 \| z_{m+1}) = F(v'_{m'} \| 1 \| z'_{m'+1})$ であり，これは F の衝突である．

Case 2. $|x| \equiv |x'| \pmod{t-1}$ のとき

Case 2a. $|x| = |x'|$ のとき $m = m'$ かつ $z_{m+1} = z'_{m+1}$ である．

- $v_m \neq v'_m$ ならば，$F(v_m \| 1 \| z_{m+1}) = F(v'_m \| 1 \| z'_{m+1})$ は F の衝突である．
- $v_m = v'_m$ ならば，$z_m \neq z'_m$ または $v_{m-1} \neq v'_{m-1}$ ならば
  $F(v_{m-1} \| 1 \| z_m) = F(v'_{m-1} \| 1 \| z'_m)$ は F の衝突である．
- すべての $2 \leq i \leq m+1$ について，$z_i = z'_i$ かつ $v_{i-1} = v'_{i-1}$ ならば，
  $z_1 \neq z'_1$ より，$F(0^n \| 0 \| z_1) = F(0^n \| 0 \| z'_1)$ は F の衝突である．
  （$z_1 = z'_1$ ならば $x = x'$ となるため）

Case 2b. $|x| \neq |x'|$ のとき
演習問題

□

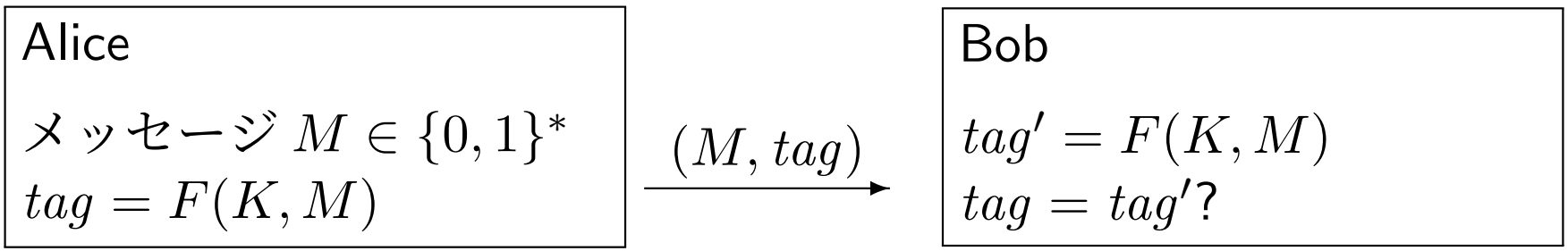
MAC 関数

MAC（メッセージ認証コード）

MAC 関数 $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$

F は改ざん検知に利用される.

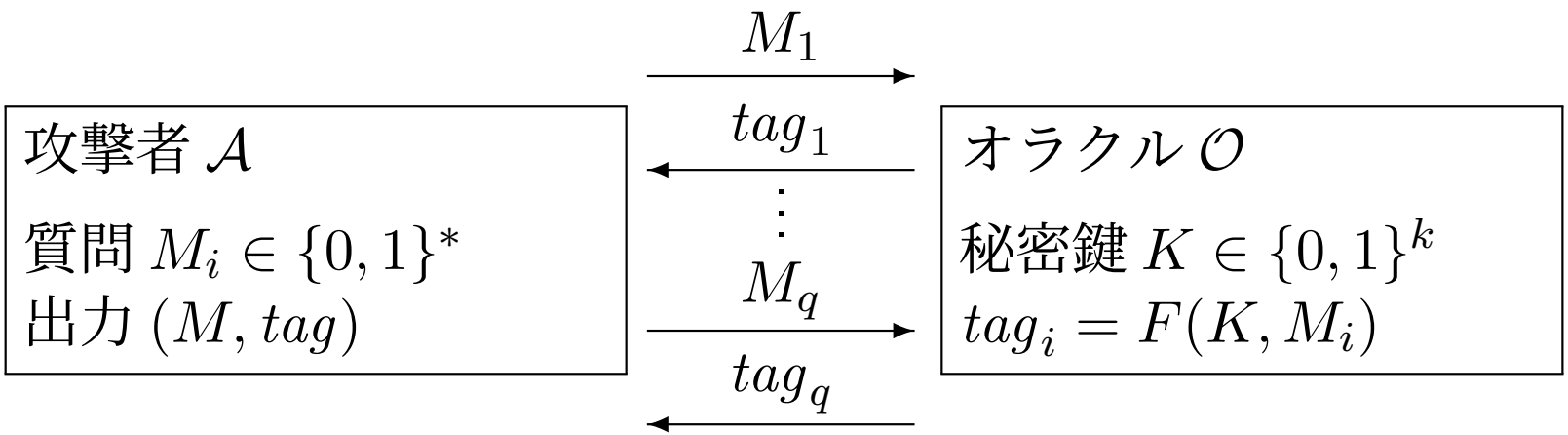
Alice と Bob が秘密鍵 $K \in \{0,1\}^k$ を共有している.



$tag \neq tag'$ ならば伝送中に改ざんが行われた.

安全な MAC 関数

MAC の安全性 = 偽造不可能性



$tag = F(K, M)$ かつ $M \notin \{M_1, \ldots, M_q\}$ ならば，$\mathcal{A}$ は偽造に成功

任意の現実的な攻撃者 $\mathcal{A}$ の成功確率が無視できる位小さいとき，F は安全な MAC 関数である．

MAC 関数の構成法

ブロック暗号を用いた構成法

- CBC-MAC
- ...

ハッシュ関数を用いた構成法

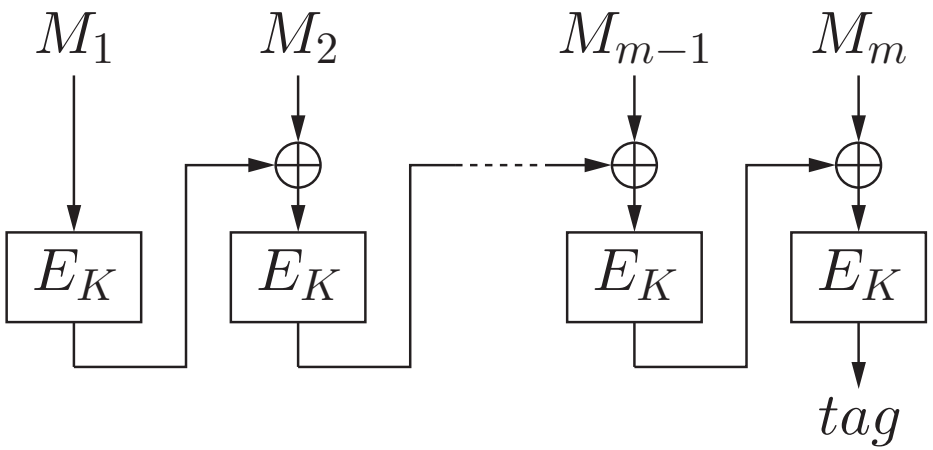
- HMAC

CBC-MAC

$E_K : \{0,1\}^n \to \{0,1\}^n$ をブロック暗号の暗号化関数とする．
K は秘密鍵

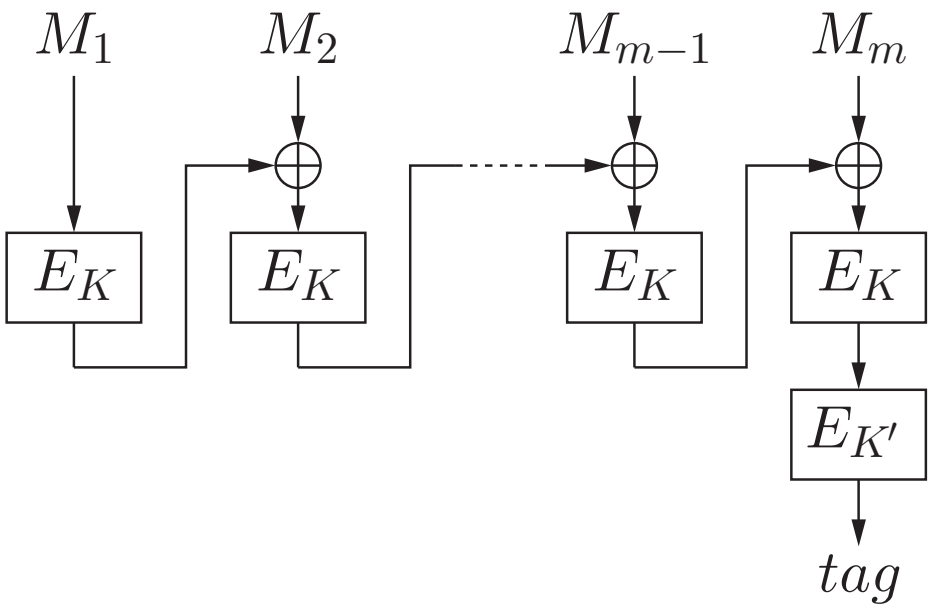
入力を $M = (M_1, \ldots, M_m)$ とする．ここで，$M_i \in \{0,1\}^n$.

$$
\begin{aligned}
v_0 &= 0^n \\
v_i &= E_K(M_i \oplus v_{i-1}) \quad \text{for } 1 \le i \le m \\
tag &= v_m
\end{aligned}
$$



Length-Extension による攻撃で容易に偽造可能

Two Key CBC-MAC



$$v_i = \begin{cases} E_K(M_i) & (i = 1) \\ E_K(M_i \oplus v_{i-1}) & (2 \leq i \leq m) \end{cases}$$

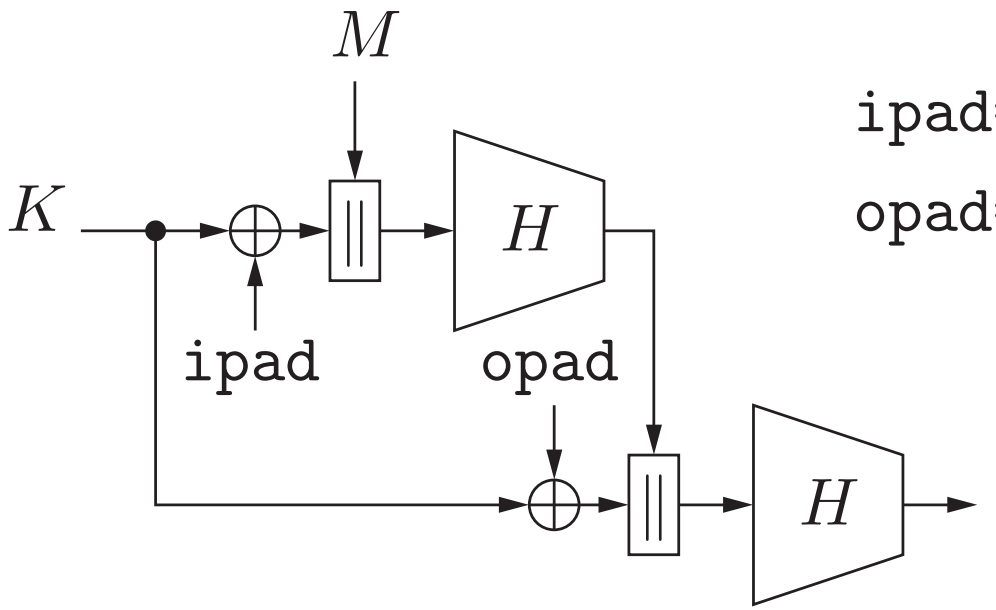
$$tag = E_{K'}(v_m)$$

Length-Extension が回避されている.

【定理 2】ブロック暗号 E が安全 $\Rightarrow$ Two Key CBC-MAC は偽造不能

HMAC

ハッシュ関数によるメッセージ認証 (MAC) 関数



ipad=0x363636...

opad=0x5c5c5c...

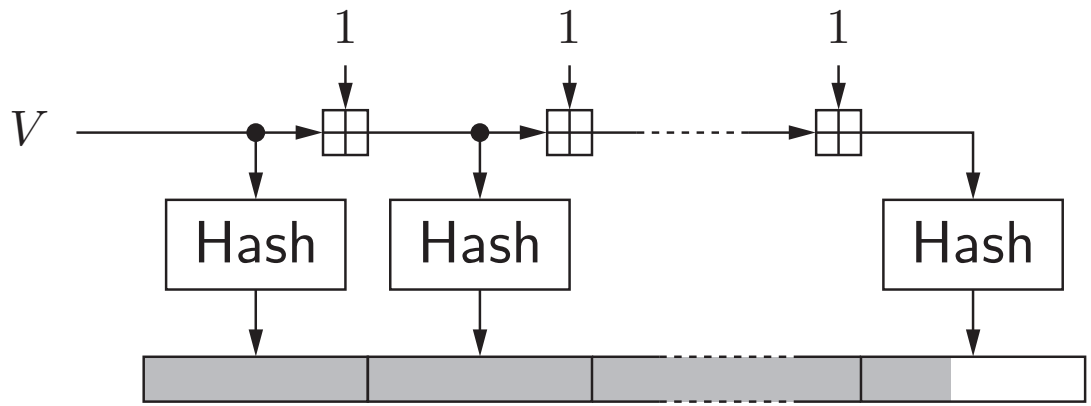
Length-Extension が回避されている.

欠点　短いメッセージに対して効率が悪い.

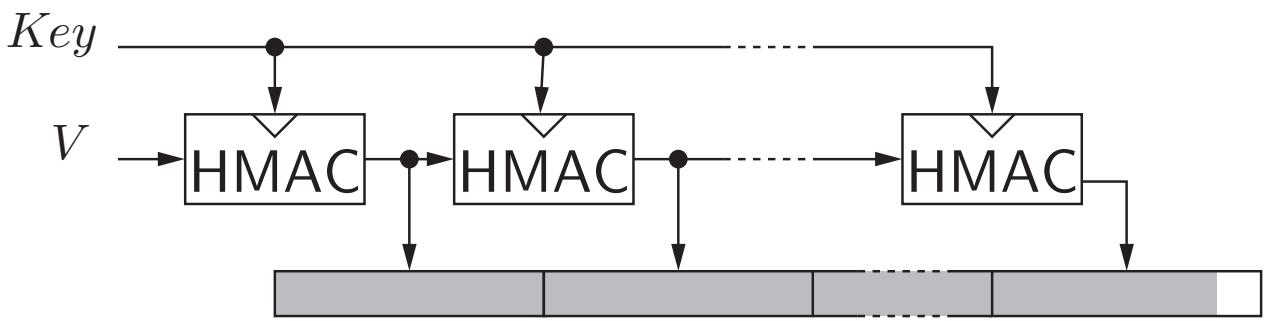
ハッシュ関数を用いた擬似乱数生成器

NIST SP 800-90

Hash_DRBG



HMAC_DRBG



Key, V は秘密鍵

演習問題

1. 定理 1 の証明を完成せよ.
2. 擬似ランダム関数が安全な MAC 関数であることを証明せよ.