暗号講座 (2007/7/14-2008/3/15, 於中央大学)

ハッシュ関数とその応用

廣瀬勝一

福井大学工学研究科電気・電子工学専攻

2007 年 9 月 29 日

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

$H : \{0,1\}^* \to \{0,1\}^\ell$　　任意長入力，固定長出力の関数

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
- …

ハッシュ関数の性質

原像計算困難性 (preimage resistance, PR)

与えられた出力 y について，$H(x) = y$ を満たす入力 x を計算するのが困難

第二原像計算困難性 (second-preimage resistance, 2ndPR)

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

衝突計算困難性 (collision resistance, CR)

$H(x) = H(x')$ を満たす相異なる入力 x, x' を計算するのが困難

ハッシュ関数攻撃の計算量

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）

原像計算	$O(2^{\ell})$
第二原像計算	$O(2^{\ell})$
衝突計算	$O(2^{\ell/2})$

ℓ は出力長

誕生日のパラドクス

23 人集まれば，誕生日の同じ人が存在する確率はおよそ 1/2

どの二人の誕生日も異なる確率は

$$\frac{365-1}{365} \times \frac{365-2}{365} \times \cdots \times \frac{365-22}{365} \approx \frac{1}{2}$$

一般に

> 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と，
> およそ $1.17\sqrt{N}$ 回で，
>
> 2 回以上選択される要素の存在する確率 $\approx 1/2$

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 $1/2$

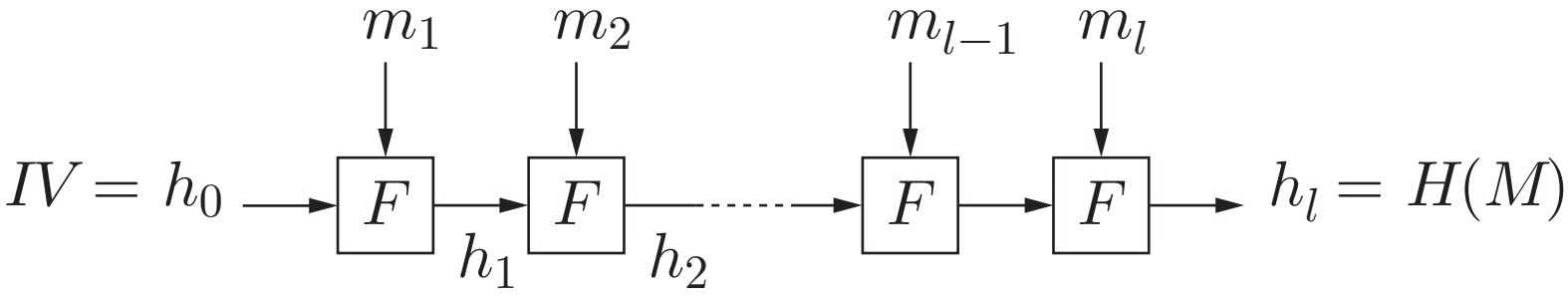
反復型ハッシュ関数

圧縮関数 $F : \{0,1\}^{\ell} \times \{0,1\}^{\ell'} \to \{0,1\}^{\ell}$

初期値 $IV \in \{0,1\}^{\ell}$

パディング 入力を ℓ' の倍数の長さの系列に変換する処理

入力 M のパディング後の系列 $(m_1, m_2, \ldots, m_l)$ について



パディング

入力 $M = (m_1, m_2, \ldots, m_l)$

- $|m_i| = \ell'$ $(i = 1, 2, \ldots, l-1)$
- $1 \leq |m_l| \leq \ell'$

曖昧さのない簡易な方法

m_1	$\cdots$	m_{l-1}	$m_l 10 \cdots 0$

MD 強化法 (MD-strengthening, by Merkle & Damgård)

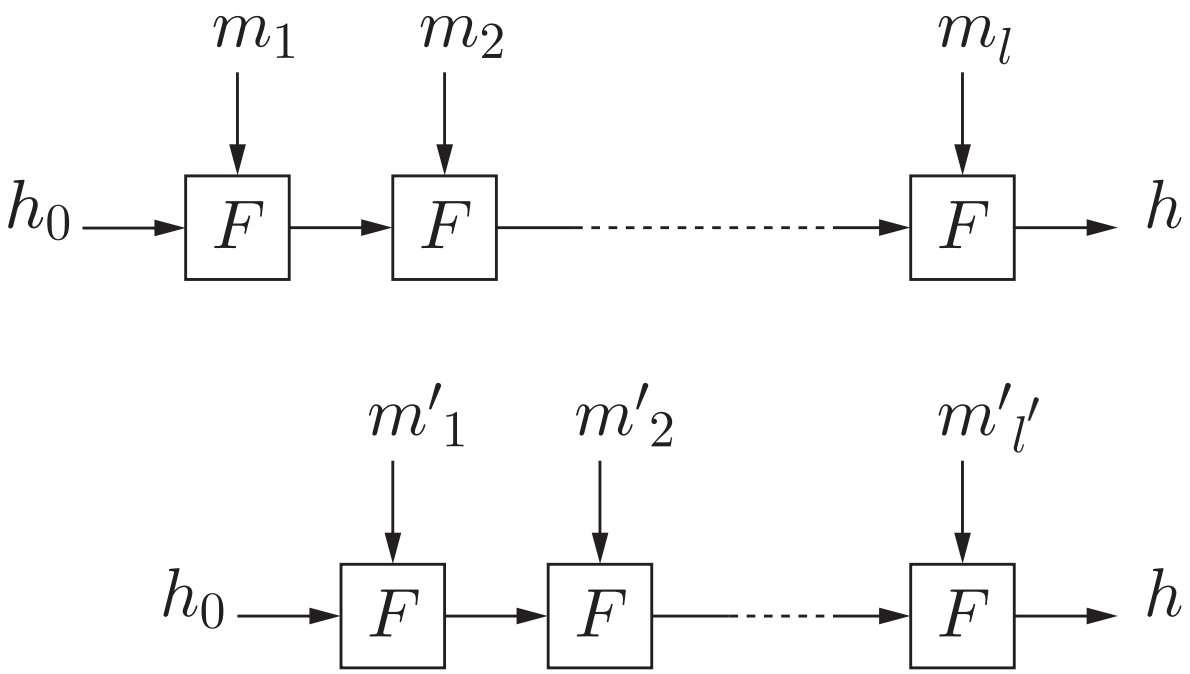
m_1	$\cdots$	m_{l-1}	$m_l 00 \cdots 0$	$\lvert M \rvert$

通常，MD 強化法に基づく方法が利用されている．

反復型ハッシュ関数の衝突計算困難性

定理 (Merkle, Damgård 89)

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

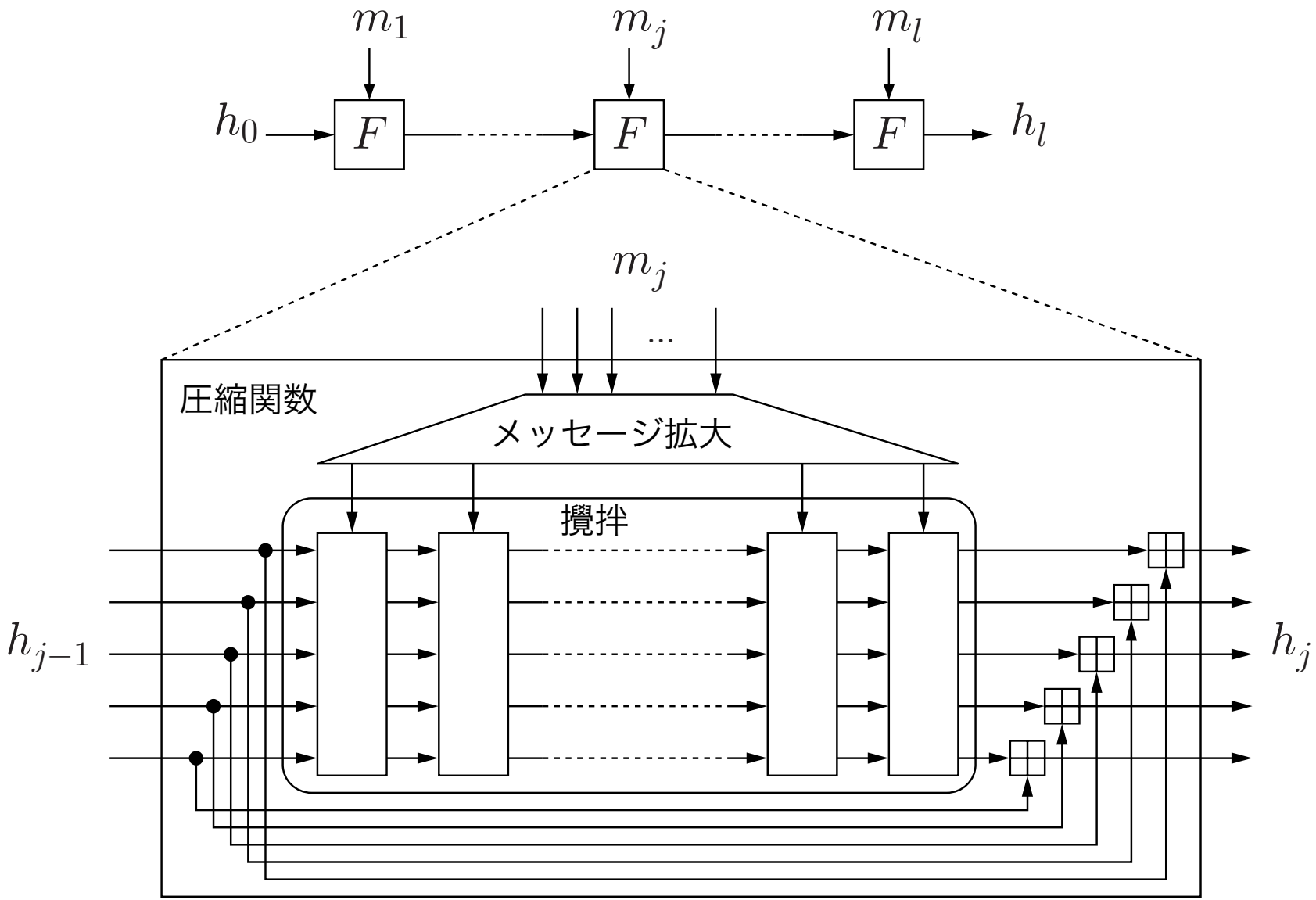


ハッシュ関数 H に衝突が見つかれば，圧縮関数 F にも衝突が見つかる

圧縮関数の構成法

- ブロック暗号を用いた構成法 (1990 年半ば以前)
  - 単ブロック長 (出力長＝ブロック長)
    Davies-Meyer, Matyas-Meyer-Oseas, Miyaguchi-Preneel
  - 倍ブロック長 (出力長＝ 2 ×ブロック長)
    MDC-2, MDC-4, abreast/tandem Davies-Meyer
- 専用構成法 (1990 年以降)
  - MDx 族
    MD4, MD5, RIPEMD-160, HAVAL,
    SHA-1, SHA-224/256/384/512
  - Whirlpool
  - …

MDx 族の圧縮関数の概略



MDx 族の圧縮関数の概略

- メッセージ拡大 (message expansion, message schedule)
  - ハミング距離を大きくする．依存関係を持たせる
    - 順序を変えて繰り返す
    - 誤り訂正符号を利用する
- 攪拌 (confusion & diffusion)
  - 1 対 1 写像
  - 非線形変換と置換の逐次的な適用
    - 簡単な非線形ブール関数を使う
    - S ボックスを使う

SHA-1 圧縮関数のメッセージ拡大

$$f_{\text{SHA-1}} : \{0,1\}^{160} \times \{0,1\}^{512} \to \{0,1\}^{160}$$

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

$$(m_{i,0}, m_{i,1}, \ldots, m_{i,15}) \to (w_0, w_1, \ldots, w_{79}) \quad w_j \in \{0,1\}^{32}$$

$$w_j = \begin{cases} m_{i,j} & \text{for } 0 \le j \le 15 \\ (w_{j-16} \oplus w_{j-14} \oplus w_{j-8} \oplus w_{j-3})^{\hookleftarrow 1} & \text{for } 16 \le j \le 79 \end{cases}$$

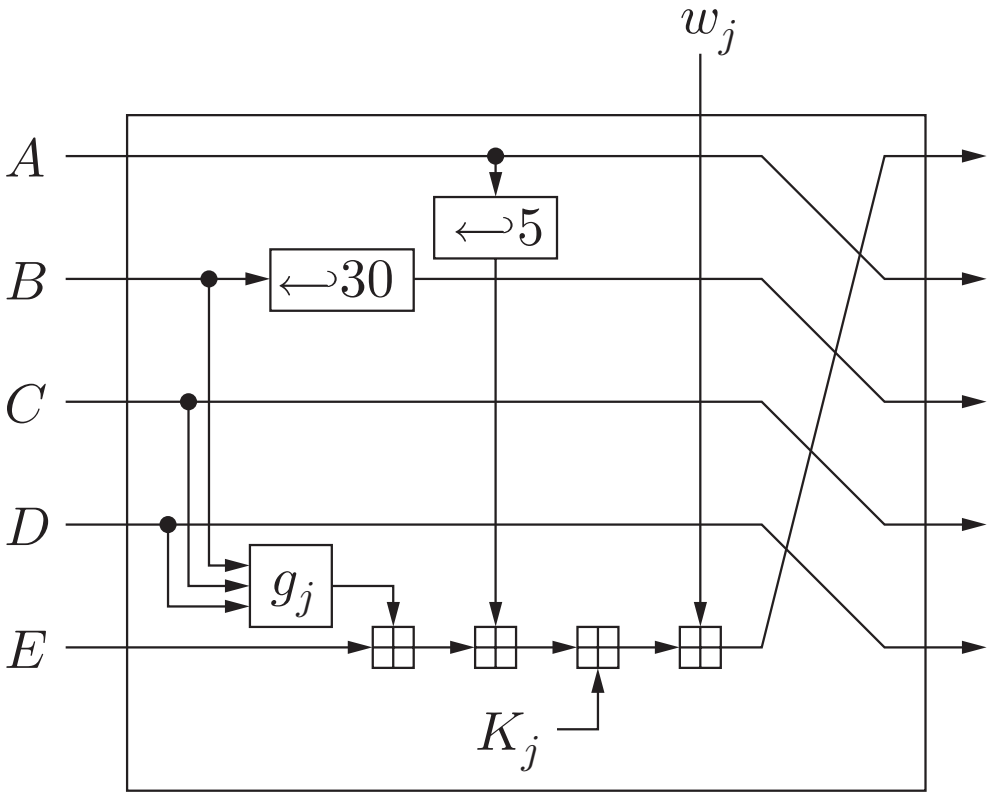
「$\hookleftarrow k$」は k ビット左巡回シフト

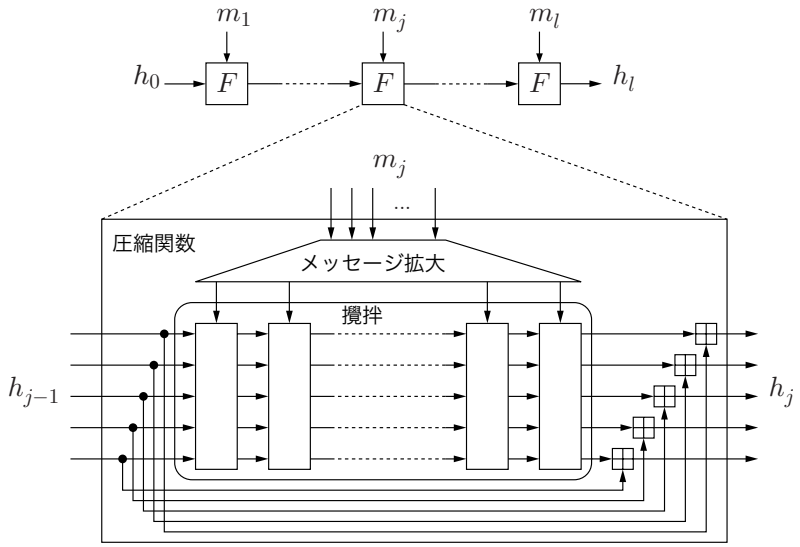
備考）$f_{\text{SHA-0}}$ では

$$w_j = w_{j-16} \oplus w_{j-14} \oplus w_{j-8} \oplus w_{j-3} \quad \text{for } 16 \le j \le 79$$

SHA-1 圧縮関数の攪拌部

$$A = A^{\hookleftarrow 5} + g_j(B, C, D) + E + K_j + w_j \bmod 2^{32}$$





この処理が 80 回行われる $(0 \leq j \leq 79)$

SHA-1 圧縮関数の攪拌部

$$g_j(u,v,w) = \begin{cases} u\,v \vee \overline{u}\,w & \text{for } 0 \le j \le 19 \quad \text{(if-then)} \\ u \oplus v \oplus w & \text{for } 20 \le j \le 39 \\ u\,v \vee u\,w \vee v\,w & \text{for } 40 \le j \le 59 \quad \text{(majority)} \\ u \oplus v \oplus w & \text{for } 60 \le j \le 79 \end{cases}$$

g_j はビットごとの演算

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

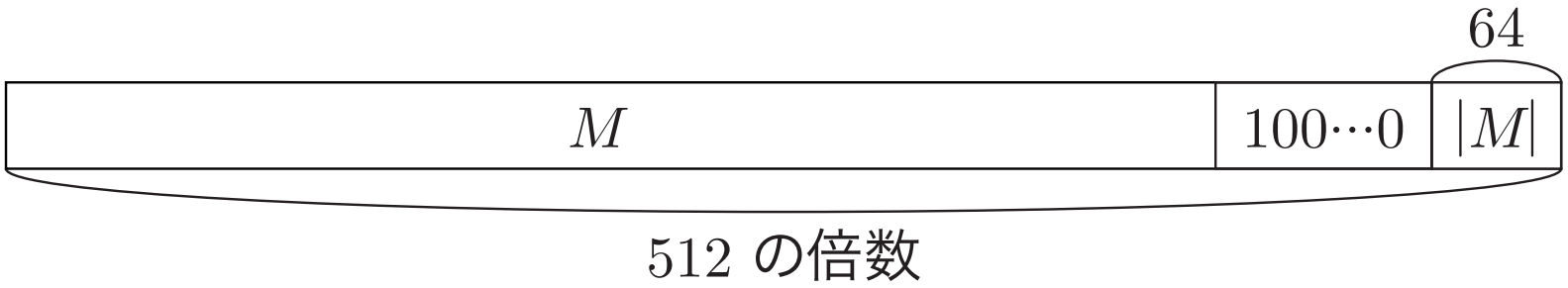
SHA-1 の初期値

$$
\begin{aligned}
h_{0,0} &= \texttt{67452301} \\
h_{0,1} &= \texttt{efcdab89} \\
h_{0,2} &= \texttt{98badcfe} \\
h_{0,3} &= \texttt{10325476} \\
h_{0,4} &= \texttt{c3d2e1f0}
\end{aligned}
$$

SHA-1 のパディング

入力 M のパディング

1. $z = M \| 10^r$ とする.
   r は $|z| + 64$ 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 $|\alpha| = 64$.



MDx 族ハッシュ関数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

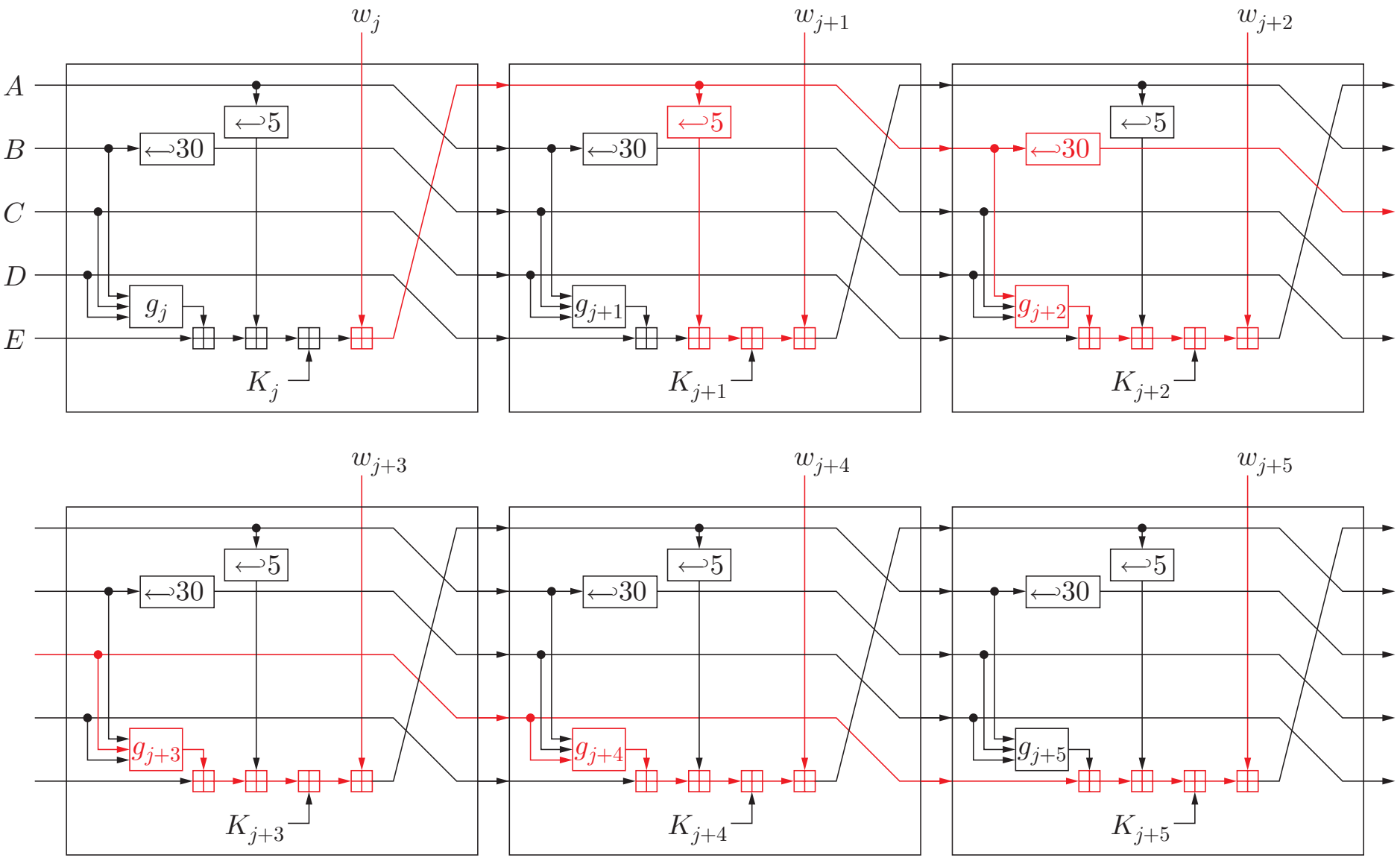
- Dobbertin (1996)
  MD4, MD5
- Chabaud & Joux (1998)
  SHA-0
- Wang, et. al. (1997, 1998, 2004–)
  MD4, MD5, HAVAL, SHA-0, SHA-1 など

衝突攻撃の計算量 (単位は圧縮関数の計算回数)

MD4	手計算でも可能
MD5	$\lesssim 2^{30}$
SHA-0	$\lesssim 2^{36}$
SHA-1	$\lesssim 2^{63}$

RIPEMD-160, SHA-224/256/384/512 に対して有効な攻撃法は発見されていない.

SHA-0/1 の局所衝突


w_j
w_{j+1}
w_{j+2}
A
B
C
D
E
$\hookleftarrow 5$
$\hookleftarrow 30$
g_j
g_{j+1}
g_{j+2}
K_j
K_{j+1}
K_{j+2}
w_{j+3}
w_{j+4}
w_{j+5}
g_{j+3}
g_{j+4}
g_{j+5}
K_{j+3}
K_{j+4}
K_{j+5}

SHA-0/1 の局所衝突

j 段目の差分の影響は後続の 5 段で解消できる.

但し,

- 各段のメッセージブロックに独立かつ任意に差分を設定することは不可能
  - メッセージ拡大法に依存して決まる
- いつも同じパタンで解消できるとは限らない
  - g_j の非線形性. g_j が 20 段ごとに変わる.
  - 加算の桁上げ

Wang の着想

- 初めの方の差分の影響は，メッセージをうまく選んで確実に解消
  - 80 段のうち 16 段の入力は自由に選択できる．
- 残りの差分の影響は，メッセージの残りの自由度で確率的に解消

ブロック暗号を用いたハッシュ関数の構成法

利点

- 極小規模のハードウェアで有効
- AES を用いた場合，MDx 族より衝突攻撃に強い (？)

欠点

- 低速

倍ブロック長ハッシュ関数を考える動機

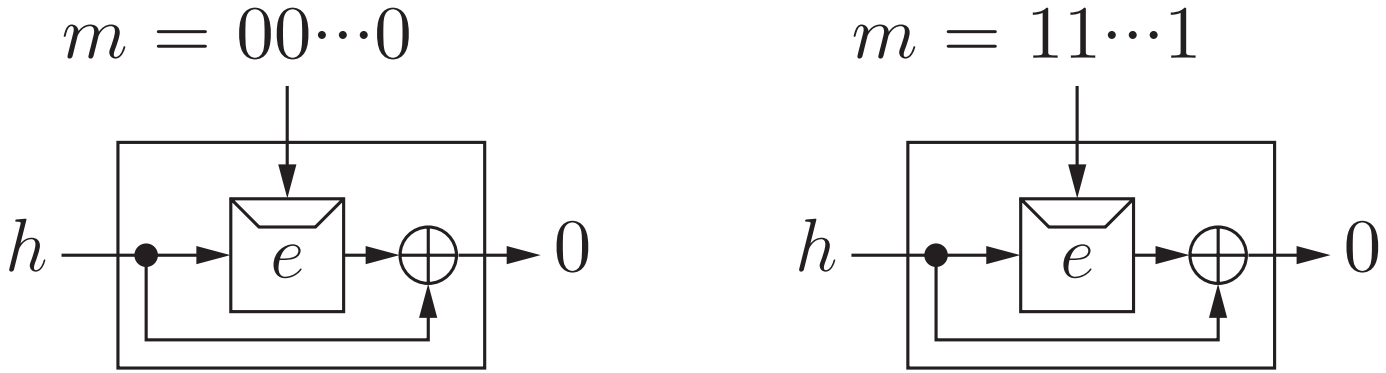
- AES で構成される単ブロック長ハッシュ関数は安全でない
  - 出力長は 128 ビット
  - 誕生日攻撃の計算量 $\approx 2^{64}$

ブロック暗号を用いたハッシュ関数の構成に関する注意点

一般的には，安全なブロック暗号から安全なハッシュ関数を構成できるとは限らない

$$e_k(x) = \begin{cases} x & k = 00\cdots0 \text{ または } 11\cdots1 \text{ のとき} \\ \mathrm{AES}_k(x) & \text{上記以外のとき} \end{cases}$$

e は安全なブロック暗号だが，以下の Davies-Meyer 圧縮関数には自明な衝突が存在



理想暗号モデル

各鍵について，ブロック暗号の暗号化関数は**可逆のランダム置換**

暗号化，復号はそれぞれ**オラクルへの質問**によって計算される．

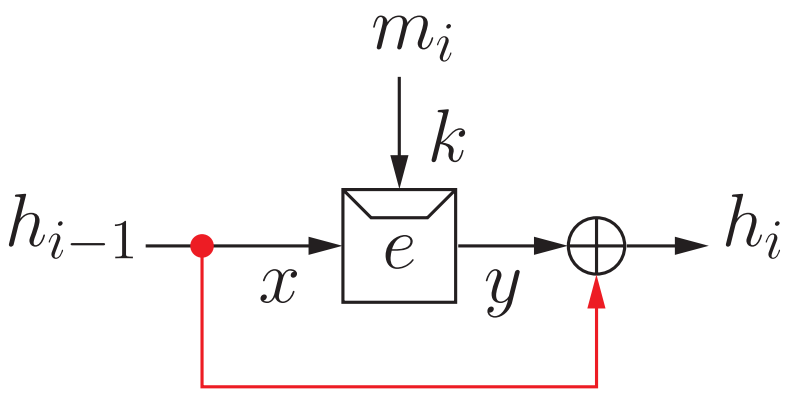
オラクル	質問	返答
暗号化 e	(鍵, 平文)	暗号文
復号 e^{-1}	(鍵, 暗号文)	平文

- 各鍵について，e, e^{-1} は 1 対 1 関数
- e, e^{-1} に不一致のないように

攻撃の計算量はオラクルへの**質問回数**

単ブロック長ハッシュ関数の安全な構成法

定理 **(Merkle 89)** 理想暗号モデルで Davies-Meyer 圧縮関数は CR



出力 h_i を得るためには，e か e^{-1} かを計算しなければならない

$$h_i = e_k(x) \oplus x \quad \text{または} \quad h_i = y \oplus {e^{-1}}_k(y)$$

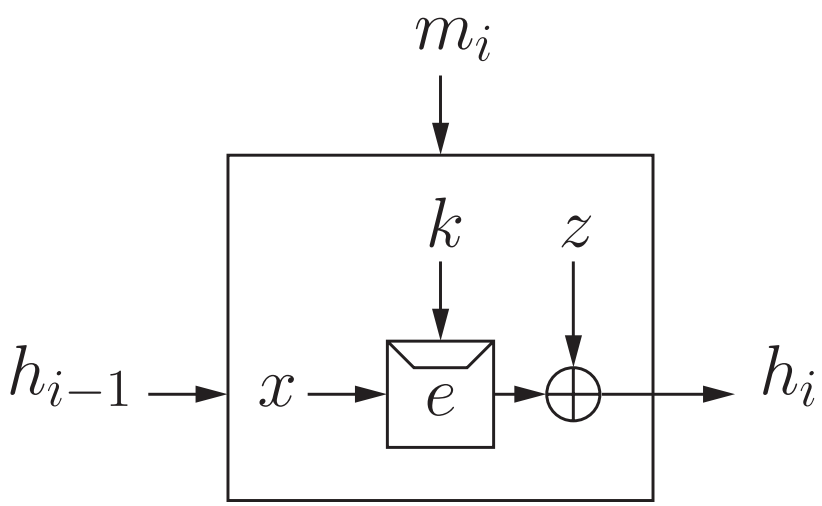
理想暗号モデルでは，h_i はランダムに決まる

いかなる攻撃の成功確率も，誕生日攻撃の成功確率と同等

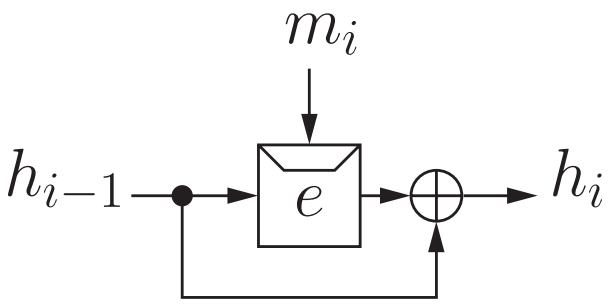
単ブロック長圧縮関数のモデル [Preneel, Govaerts, Vandewalle 93]

$x, k, z \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i, 0\}$

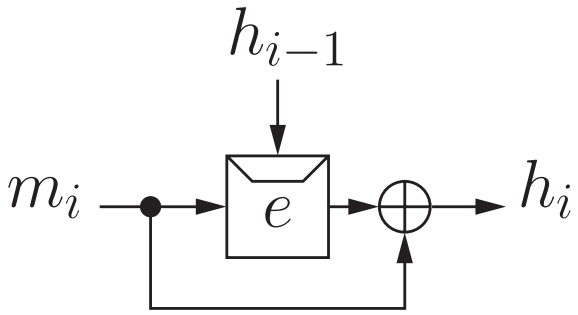
$4^3 = 64$ 通りの構成



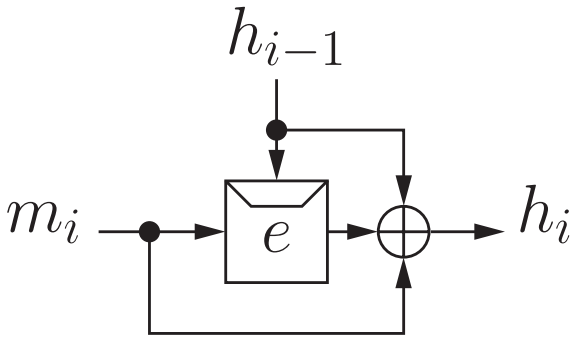
例)



Davies-Meyer



Matyas-Meyer-Oseas



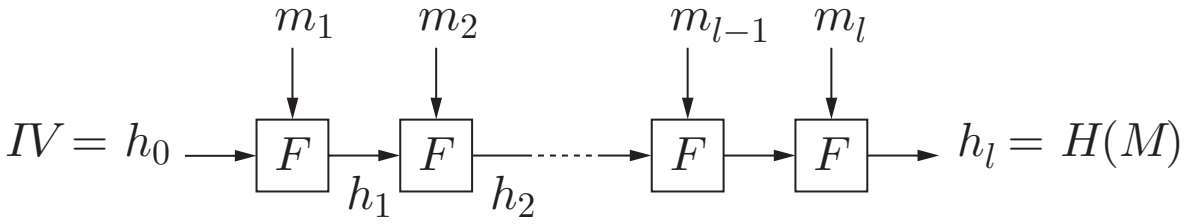
Miyaguchi-Preneel

単ブロック長ハッシュ関数の安全な構成法

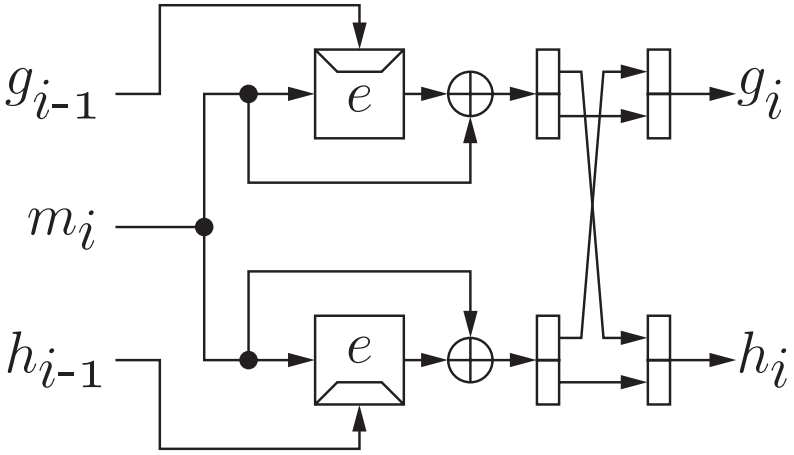
定理 (Black, Rogaway, Shrimpton 02)
Preneel らのモデルに属する圧縮関数について，理想暗号モデルで

1. CR 圧縮関数が 12 個存在する
2. 上記以外の 8 個の圧縮関数を用いて CR 反復型ハッシュ関数が構成できる

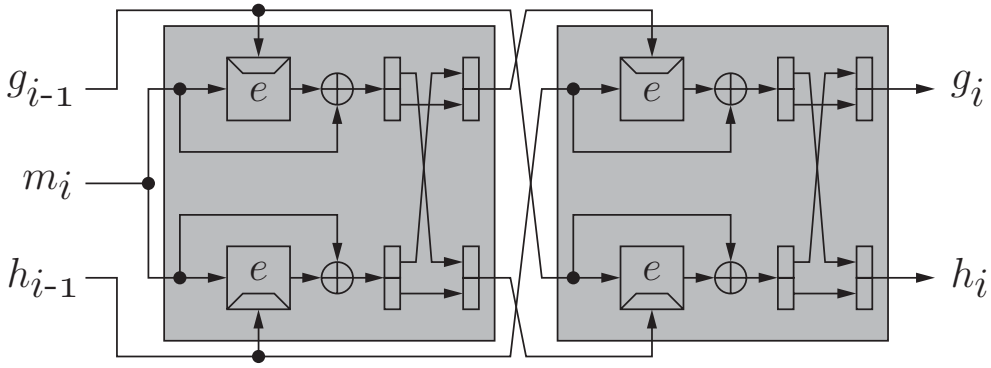
- 1 の証明は Merkle の定理と同様
- 2 の証明は反復型の構造を利用して行われる



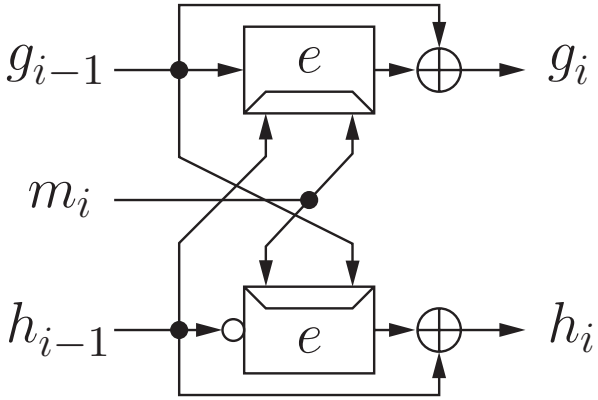
倍ブロック長ハッシュ関数の既存の主な構成法



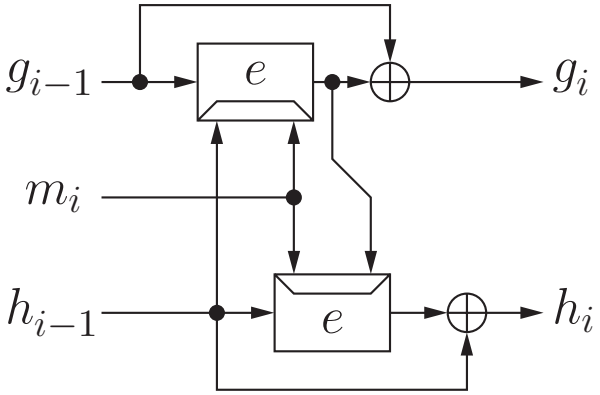
MDC-2



MDC-4

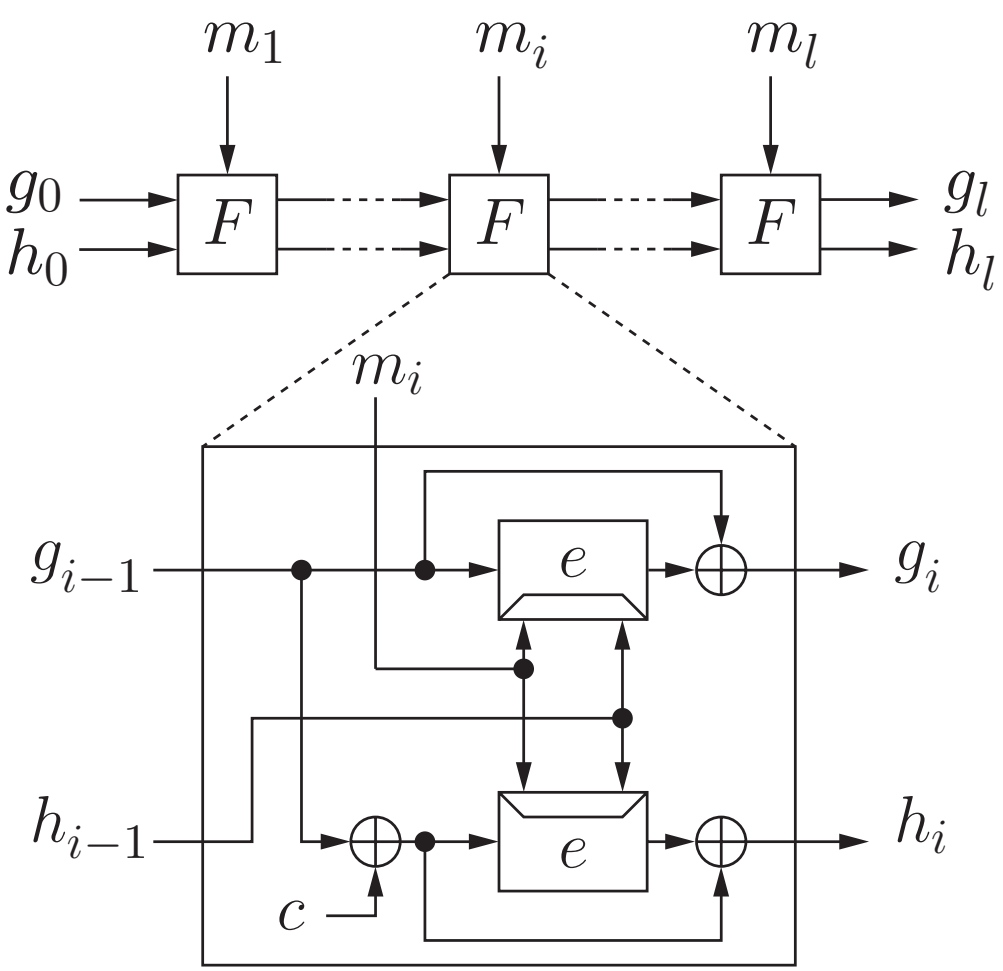


abreast Davies-Meyer



tandem Davies-Meyer

倍ブロック長ハッシュ関数の安全な構成法



- c は非零の定数
- 二つの暗号化関数の鍵に対応する入力が同一
  - 鍵拡大が 1 度で済む
- AES を利用する場合
  - 出力長は 256 ビット
  - 鍵長 192/256 ビットで利用

倍ブロック長ハッシュ関数の安全な構成法

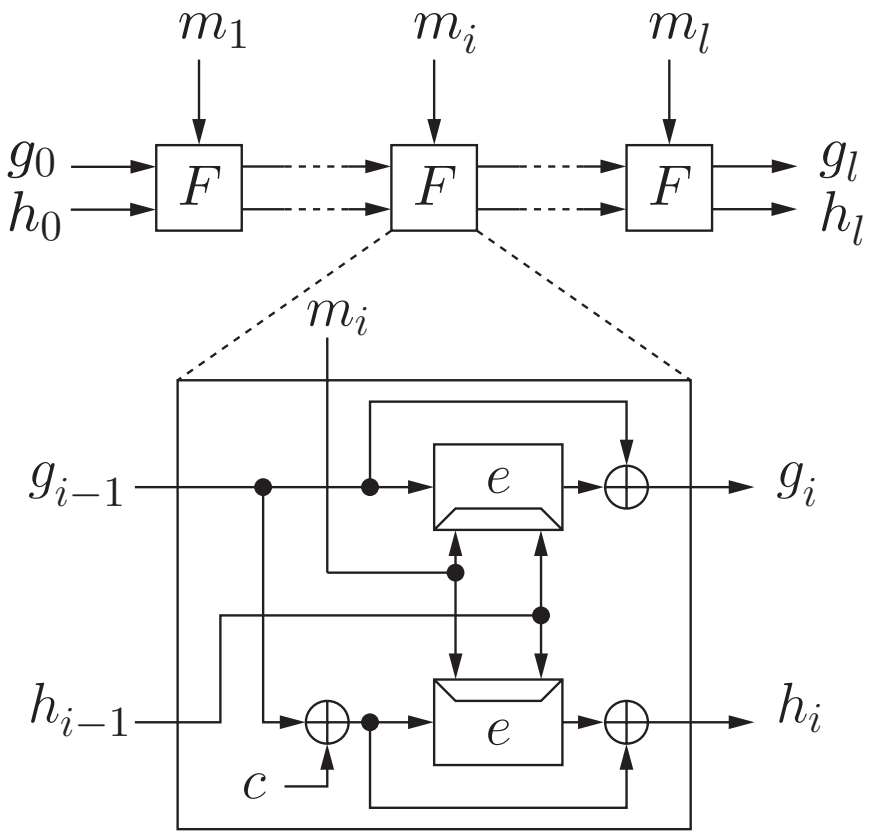
定理 (Hirose 06)

$F : \{0,1\}^{2n+b} \to \{0,1\}^{2n}$

(e, e^{-1}) に q 組の質問をする攻撃者を A とする

このとき，理想暗号モデルで，
$1 \le q \le 2^{n-2}$ に対して

$$\mathbf{Adv}_H^{\mathrm{coll}}(A) \le 12\left(\frac{q}{2^n}\right)^2$$



ここで，$\mathbf{Adv}_H^{\mathrm{coll}}(A)$ は A が H の衝突を得る確率

反復型ハッシュ関数の構造を利用した攻撃法

- Joux の多衝突攻撃 (multi-collision attack) (CRYPTO 2004)
- Kelsey と Schneier の第二原像攻撃 (EUROCRYPT 2005)

多衝突攻撃 (Multi-collision Attack)

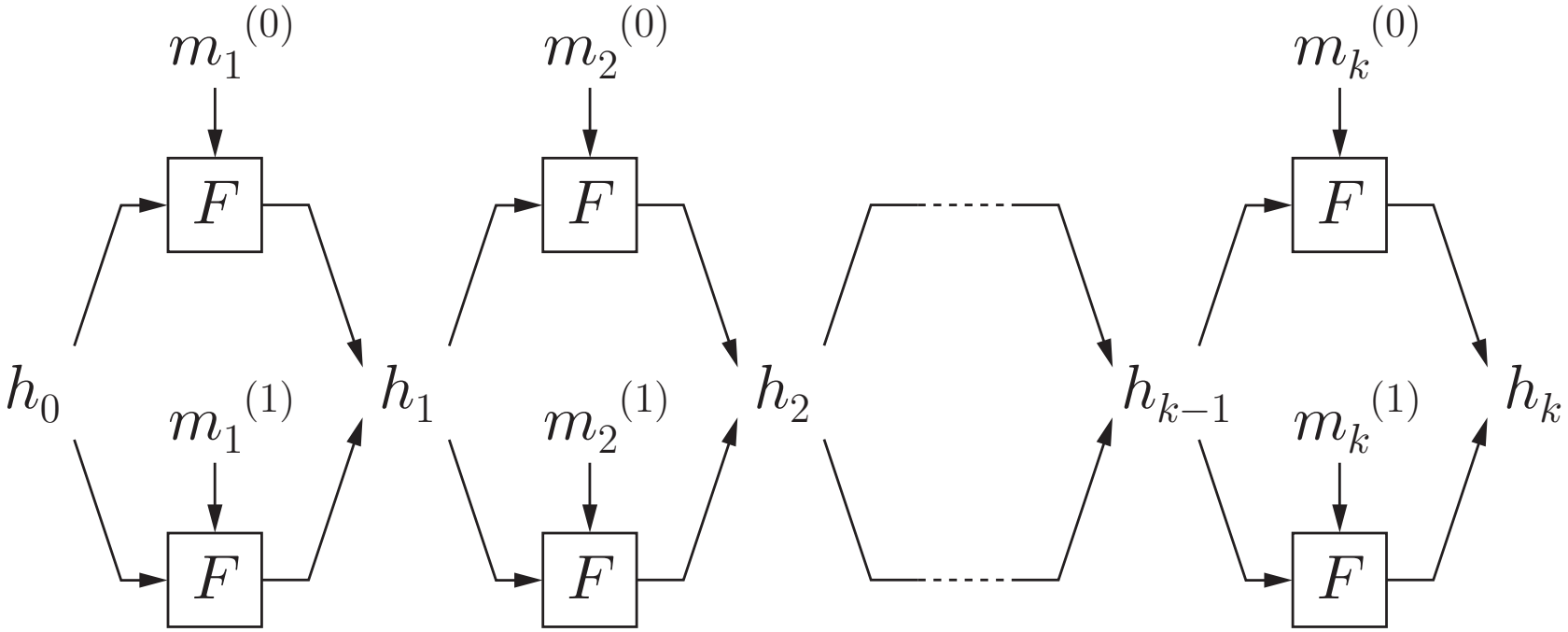
K 衝突攻撃 (K-collision attack)

出力の等しい K 個の相異なる入力を計算する攻撃

ハッシュ関数の構造を利用することなく，誕生日攻撃で K 衝突 (K-collision) を得るための計算量は $O\left(2^{\frac{K-1}{K}\ell}\right)$

ℓ はハッシュ値の長さ

反復型ハッシュ関数に対する Joux の多衝突攻撃



2^k 衝突: $m_1^{(b_1)}, m_2^{(b_2)}, \ldots, m_k^{(b_k)}$ に対応する出力はすべて等しい.

計算量は $O(k\,2^{\ell/2})$, ℓ はハッシュ値の長さ

Joux の多衝突攻撃の応用

H は反復型ハッシュ関数で出力長は ℓ_1
G は任意のハッシュ関数で出力長は ℓ_2

$H(M)||G(M)$ への衝突攻撃

1. H の $2^{\ell_2/2}$ 衝突を計算する．計算量は $O(\ell_2\, 2^{\ell_1/2})$.
   H の $2^{\ell_2/2}$ 衝突を $C_H = \{M_1, M_2, \ldots, M_{2^{\ell_2/2}}\}$ とする．
2. C_H の中で G の 2 衝突を探す．計算量は $O(2^{\ell_2/2})$.

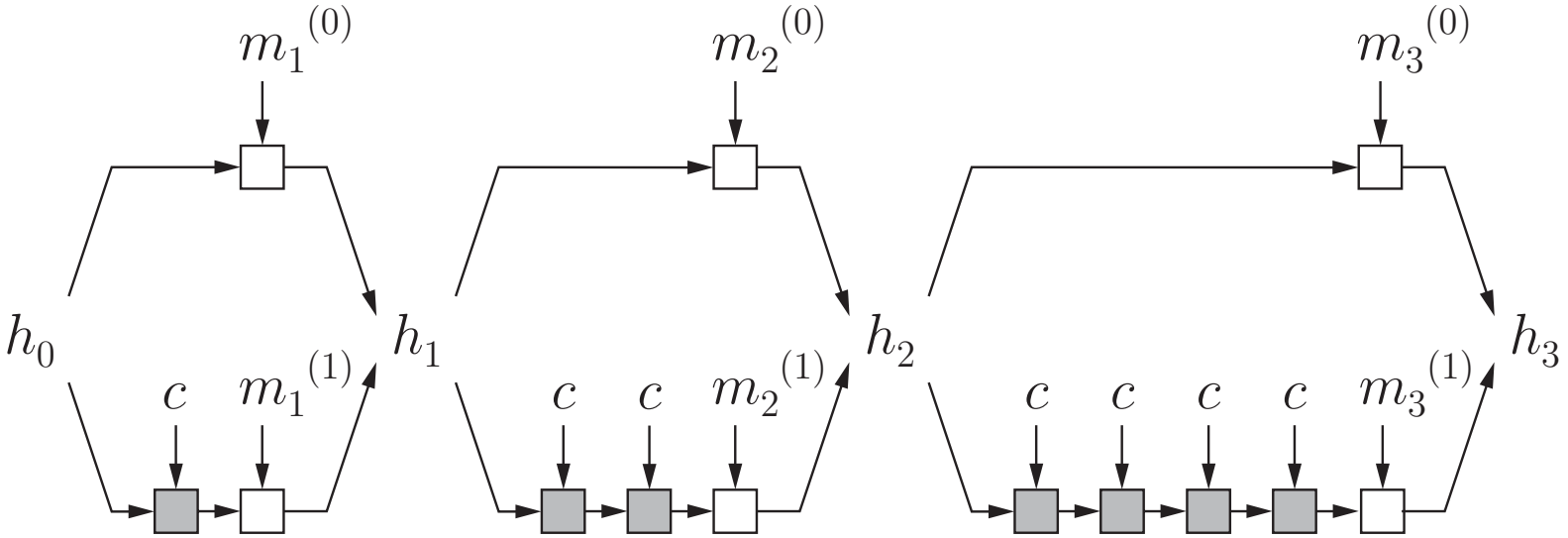
この攻撃の計算量は $O(\ell_2\, 2^{\ell_1/2} + 2^{\ell_2/2})$.

参考）$H(x)||G(x)$ への誕生日攻撃の計算量は $O(2^{(\ell_1+\ell_2)/2})$

Kelsey と Schneier の第二原像攻撃

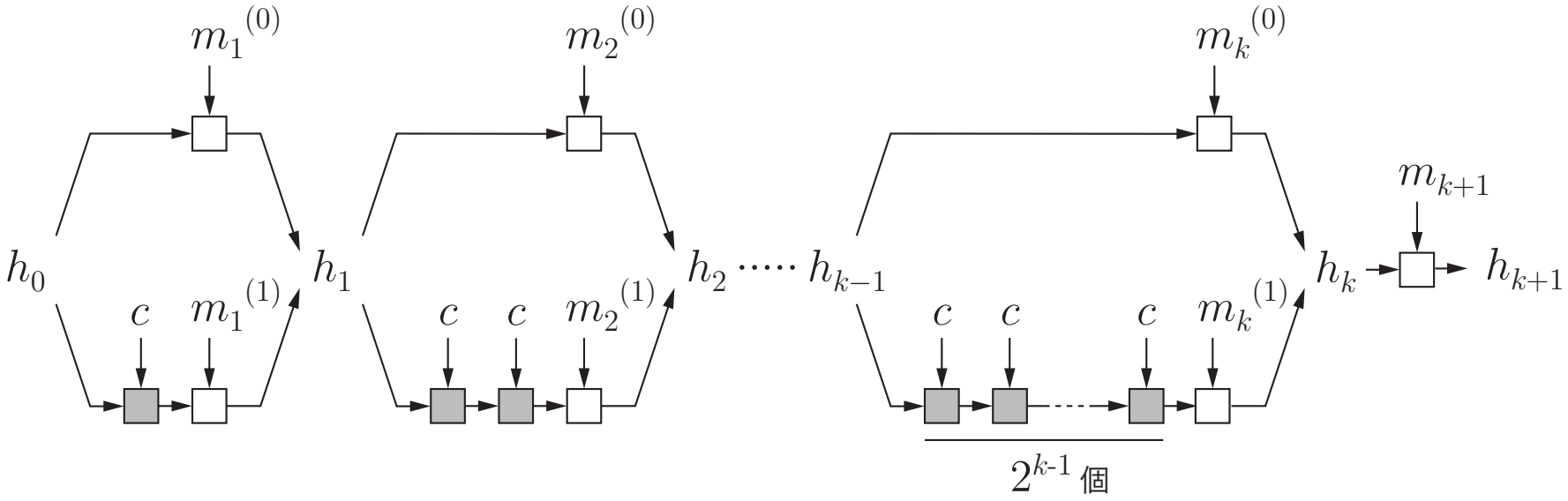
Joux の多衝突攻撃の応用

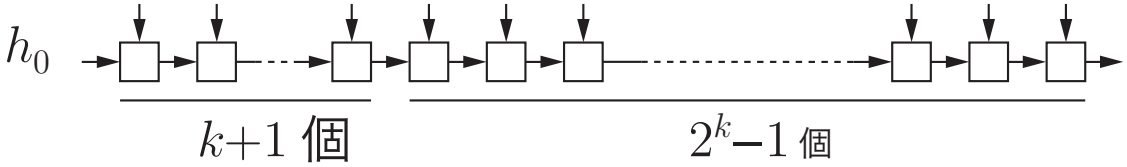
例)



- 長さが 3 から 10 $(= 3 + 2^0 + 2^1 + 2^2)$ ブロックの入力
- どの出力も h_3 で等しい

Kelsey と Schneier の第二原像計算攻撃





$k = \ell/2$ とすれば，h_{k+1} を下の $2^k - 1$ 個のいずれかに衝突させられる．

計算量は $O(\ell\, 2^{\ell/2})$．

ここまでのまとめ：CR ハッシュ関数の構成は難しい？

ハッシュ関数の構造を利用した強力な衝突攻撃

- Wang らの差分攻撃
- Joux の多衝突攻撃

より構成の容易な関数での代替は可能か？

CR と UOW

ハッシュ関数の集合

$$\left\{h_k \,|\, h_k : \{0,1\}^l \to \{0,1\}^\ell,\ l > \ell,\ k \in \{0,1\}^n\right\}$$

Universal one-wayness [Naor, Yung 89]

はじめに攻撃者が入力 x を選ぶ．ランダムに与えられた k について，$h_k(x) = h_k(x') \wedge x \neq x'$ を満たす入力 x' を計算するのが困難.

Collision resistance

ランダムに与えられた k について，$h_k(x) = h_k(x') \wedge x \neq x'$ を満たす入力 x, x' を計算するのが困難.

- UOW は Target Collision Resistance (TCR) とも呼ばれる.
- CR を計算量理論に基づいて扱うときは，関数の集合を考える.

CR と UOW とは本質的に異質

UOW ハッシュ関数の構成

- 一方向置換をブラックボックスとして利用 [Naor, Yung 89]
- 一方向関数をブラックボックスとして利用 [Rompel 90]

CR ハッシュ関数の構成

- 一方向置換をブラックボックスとして利用した構成は**不可能** [Simon 98]

UOW ハッシュ関数によるダイジェストを用いたデジタル署名

UOW ハッシュ関数の集合 $\{h_k \mid k \in \{0,1\}^n\}$

メッセージ M への署名の方法

1. 署名者が k を乱択する.
2. $k \| h_k(M)$ に署名する.

問題点

- 署名者には $h_k(M) = h_k(M')$ を満たす M, M' を得る可能性有.
  （実用上の支障はない？）
- 任意長入力の h_k がうまく構成できるか？

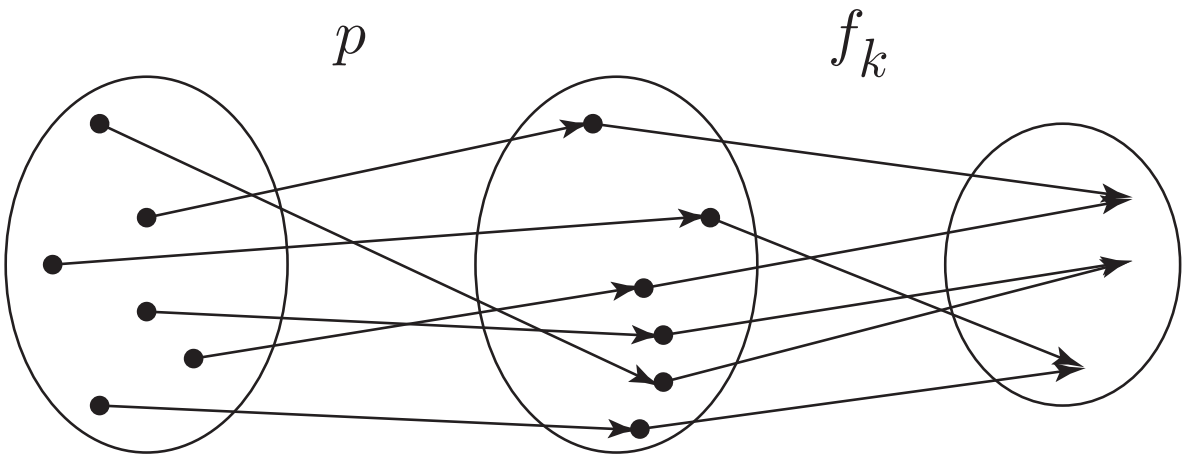
一方向置換を用いたUOWハッシュ関数の構成法

$p : \{0,1\}^\ell \to \{0,1\}^\ell$ は一方向置換

$F = \{f_k \mid f_k : \{0,1\}^\ell \to \{0,1\}^{\ell-1}, k \in \{0,1\}^n\}$

$H = \{f_k \circ p \mid f_k \circ p : \{0,1\}^\ell \to \{0,1\}^{\ell-1}, k \in \{0,1\}^n\}$

定理　F がユニバーサルハッシュ関数族 $\Rightarrow$ H はUOW



ユニバーサルハッシュ関数族 [Wegman, Carter 81]

$$F = \{f \mid f : X \to Y,\ |X| \geq |Y|\}$$

定義　F がユニバーサルハッシュ関数族 $\Leftrightarrow$

任意の $a_1, a_2 \in X$ と $b_1, b_2 \in Y$ について，$a_1 \neq a_2$ のとき，

$$\Pr[f(a_1) = b_1 \wedge f(a_2) = b_2 \mid f \xleftarrow{\$} F] = \frac{1}{|Y|^2}$$

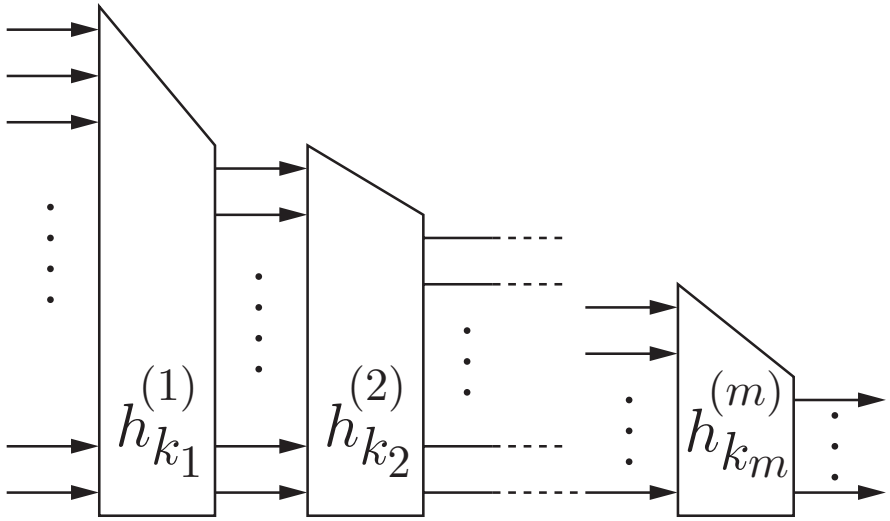
$f \xleftarrow{\$} F$ は F からの無作為な選択

一方向置換を用いたUOWハッシュ関数の構成法

$$H^{(i)} = \{h_{k_i}^{(i)} \mid h_{k_i}^{(i)} : \{0,1\}^{\ell_i} \to \{0,1\}^{\ell_{i-1}}, k_i \in \{0,1\}^{n_i}\}$$

$$H = \{h_{k_1}^{(1)} \circ h_{k_2}^{(2)} \cdots \circ h_{k_m}^{(m)} \mid k_i \in \{0,1\}^{n_i}\}$$

定理　$H^{(1)}, \ldots, H^{(m)}$がUOW $\Rightarrow$ HはUOW

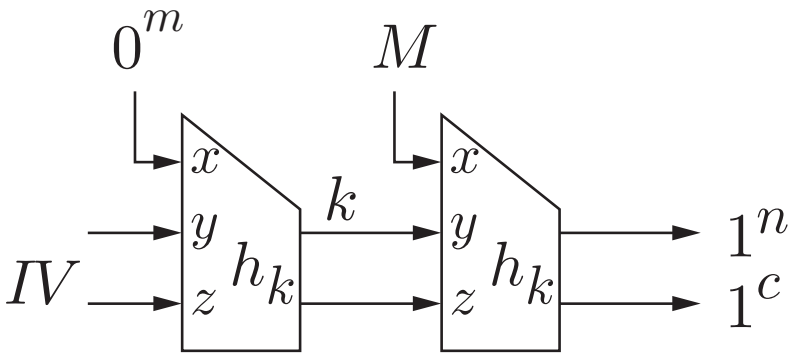


任意長入力の UOW ハッシュ関数

Merkle, Damgård の構成は一般にはうまく働かない [Bellare, Rogaway 97]

例 $\{h_k \mid h_k : \{0,1\}^{m+n+c} \to \{0,1\}^{n+c},\ k \in \{0,1\}^n\}$

$$h_k(x,y,z) = \begin{cases} (k, f_k(x,y,z)) & \text{if } y \neq k \\ (1^n, 1^c) & \text{if } y = k \end{cases}$$



M は任意なので，衝突が容易に見つかる

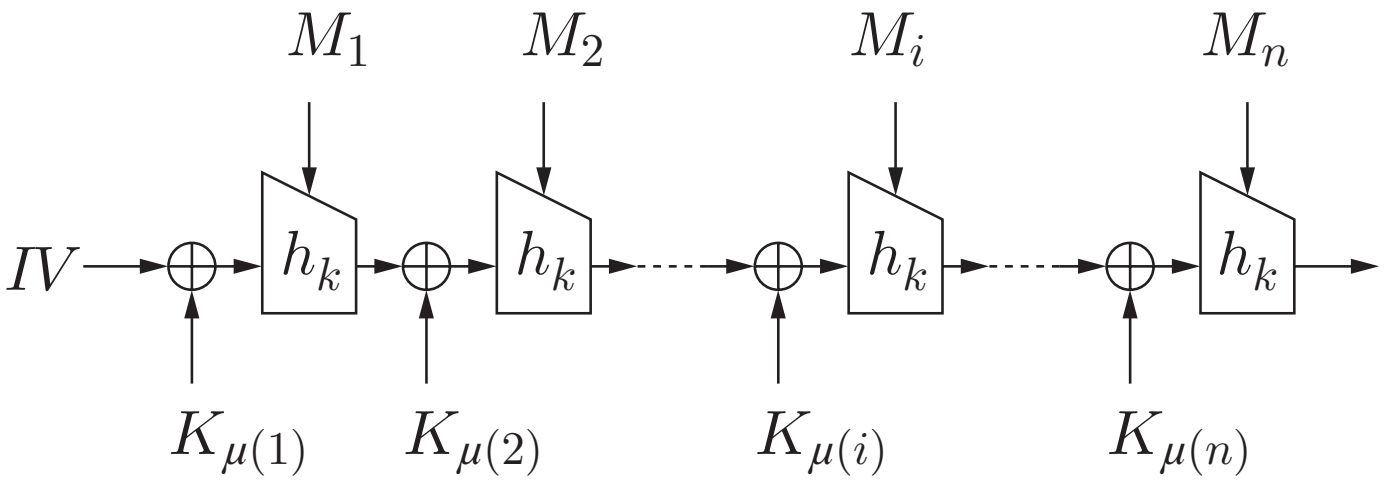
任意長入力の UOW ハッシュ関数

$$h_k(x,y,z) = \begin{cases} (k, f_k(x,y,z)) & \text{if } y \neq k \\ (1^n, 1^c) & \text{if } y = k \end{cases}$$

ここで，$f_k : \{0,1\}^{m+n+c} \to \{0,1\}^c$

補題　$\{f_k\}$ が UOW $\Rightarrow$ $\{h_k\}$ は UOW

任意長入力の UOW ハッシュ関数の構成 [Shoup 00]



$$\mu(i) = \text{largest integer } \mu \text{ such that } 2^{\mu} | i$$

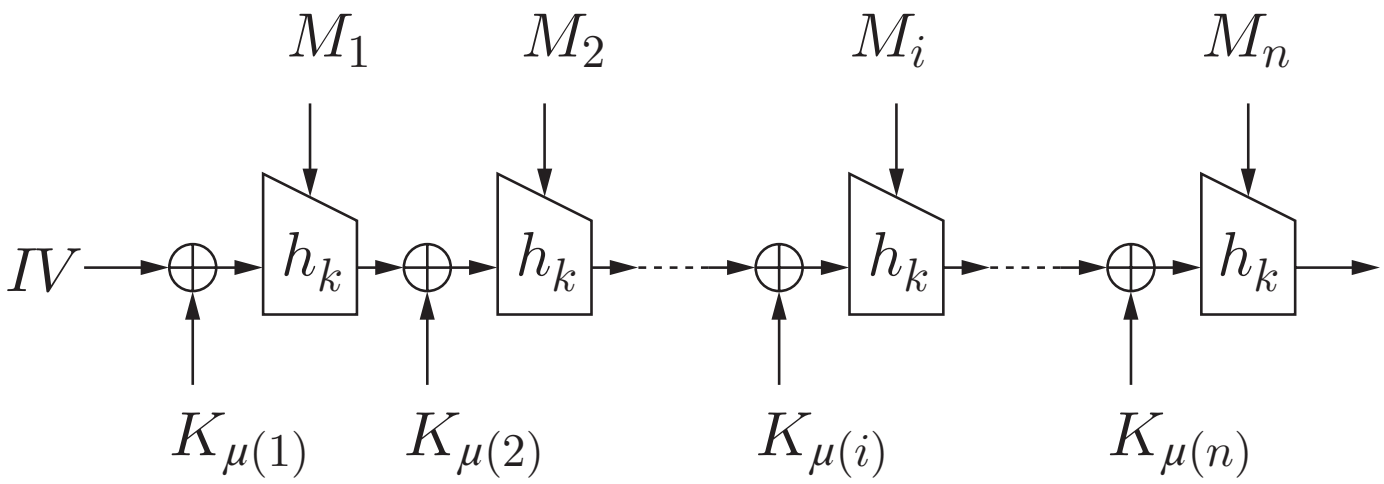
k および $K_0, K_1, \ldots, K_{\lfloor \log n \rfloor}$ がランダムに選択される.

定理　$\{h_k\}$ が UOW $\Rightarrow$ 上図の関数の集合は UOW

問題点）K_i の個数が $\log n$ に比例して増える.

任意長入力の UOW ハッシュ関数の構成

下図の型に限れば Shoup 00 の構成は最適 [Mironov 01]



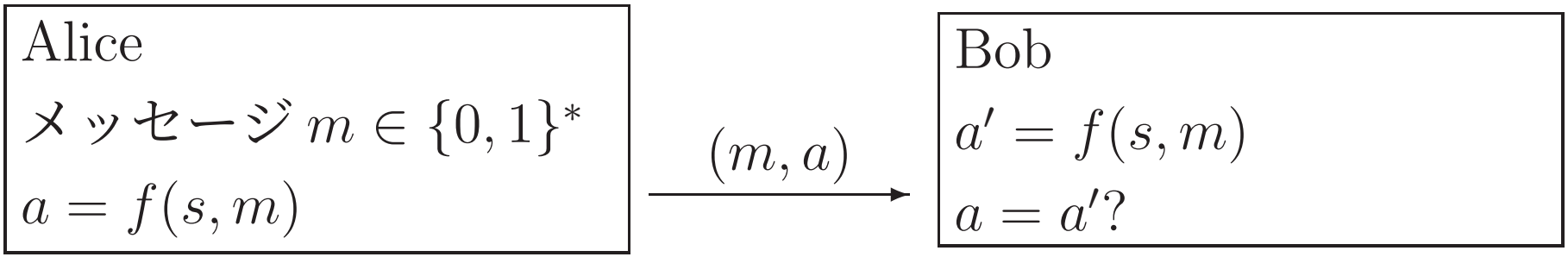
定理　任意の関数 μ について

上記の関数の集合が UOW $\Rightarrow |\mu(\{1, 2, \ldots, n\})| > \log n$

メッセージ認証コード (MAC)

MAC 関数を $f : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^\ell$ とする

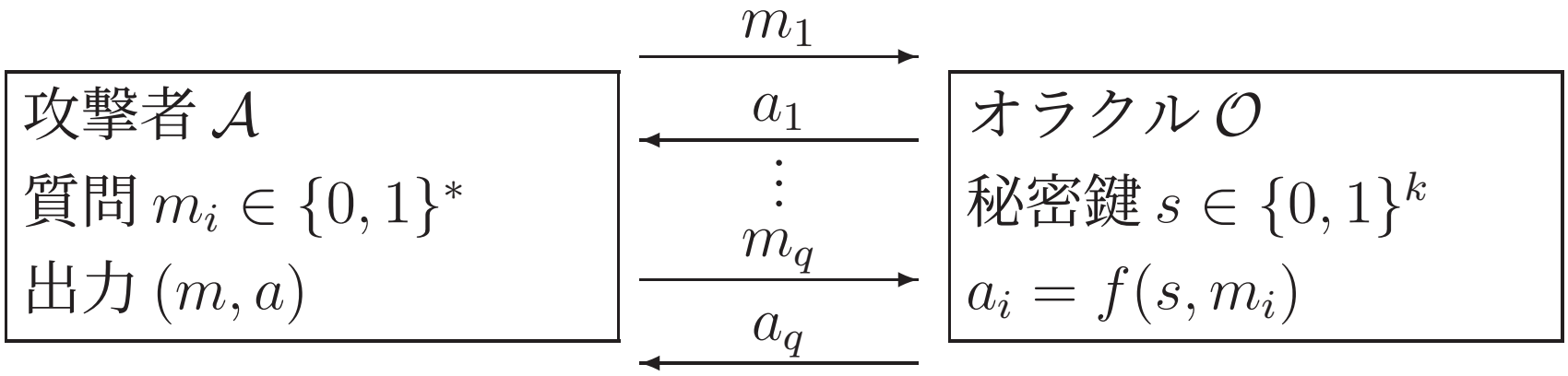
Alice と Bob が秘密鍵 $s \in \{0,1\}^k$ を共有している.



f はメッセージが改竄なく伝送されたかを確認するため利用される.

安全な MAC

MAC の安全性 = 偽造不可能性

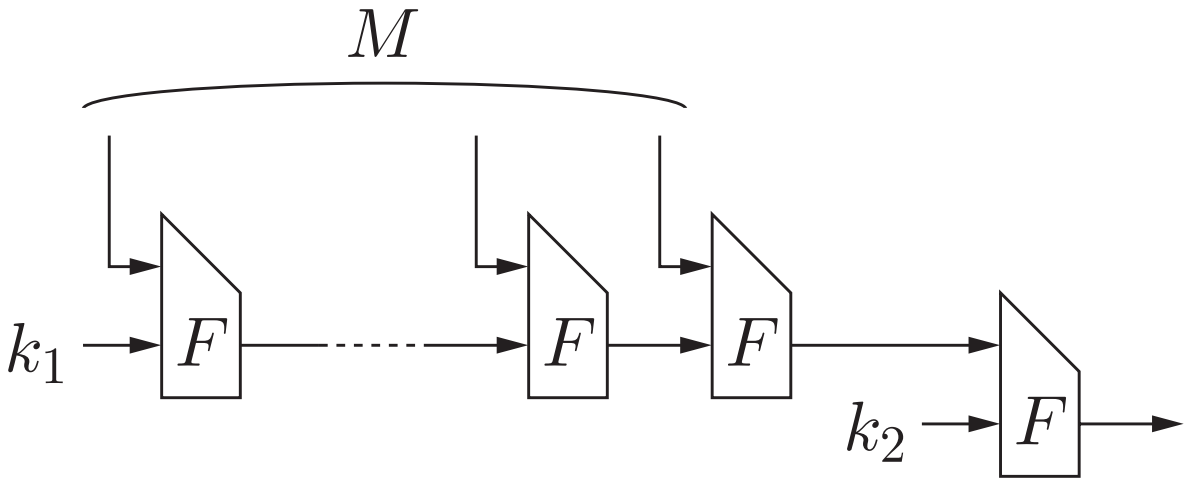


$a = f(s, m)$ かつ $m \notin \{m_1, \ldots, m_q\}$ ならば，$\mathcal{A}$ は偽造に成功

定義　任意の多項式時間限定確率アルゴリズムである攻撃者 $\mathcal{A}$ の成功確率が無視できる位小さいとき，f は安全な MAC である．

圧縮関数を用いたMAC関数の構成法 NMAC

$F : \{0,1\}^{\ell} \times \{0,1\}^{b} \to \{0,1\}^{\ell}$ をハッシュ関数の圧縮関数とする.



二つの秘密鍵 k_1, k_2 が用いられる.

定理 **(Bellare 06)** F が擬似ランダム $\Rightarrow$ NMAC は擬似ランダム

命題 擬似ランダム関数 $\Rightarrow$ 安全な MAC 関数

ハッシュ関数を用いた MAC 関数の構成法 HMAC

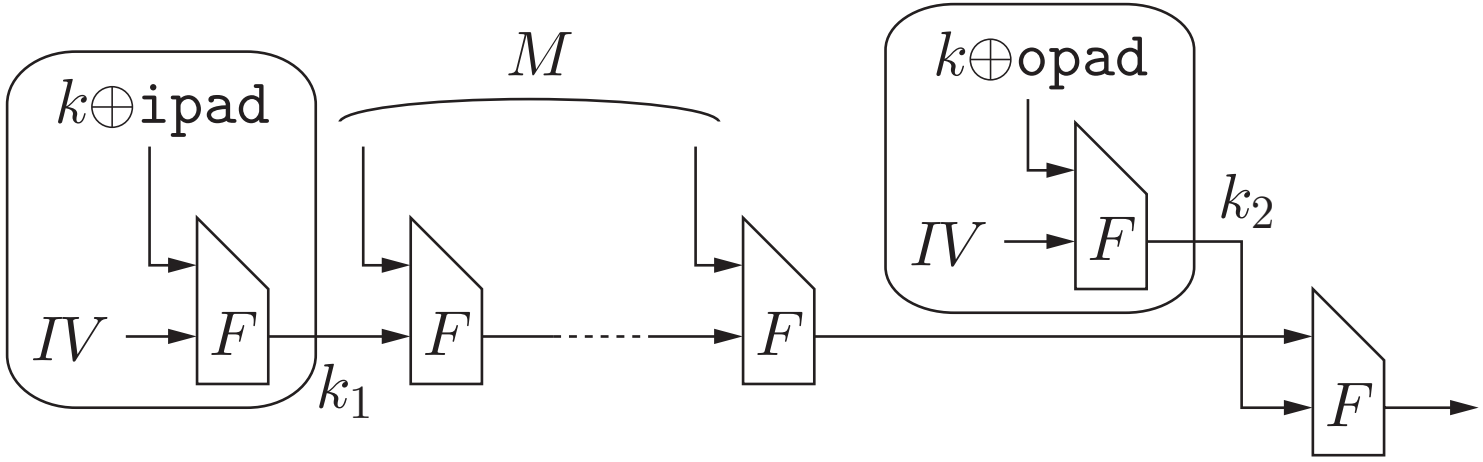
$H : \{0,1\}^* \to \{0,1\}^\ell$ をハッシュ関数とする.

$$\mathsf{HMAC}_k(M) = H((k \oplus \mathtt{opad}) \| H((k \oplus \mathtt{ipad}) \| M))$$

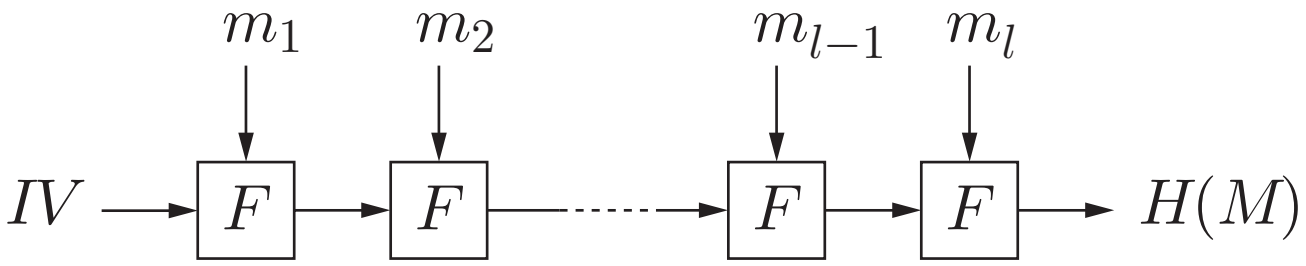
$$\mathtt{ipad} = \mathtt{0x3636 \cdots 36}$$

$$\mathtt{opad} = \mathtt{0x5c5c \cdots 5c}$$

H が反復型ハッシュ関数のとき



反復型ハッシュ関数の問題点



利点

- F が CR $\Rightarrow$ H が CR

欠点

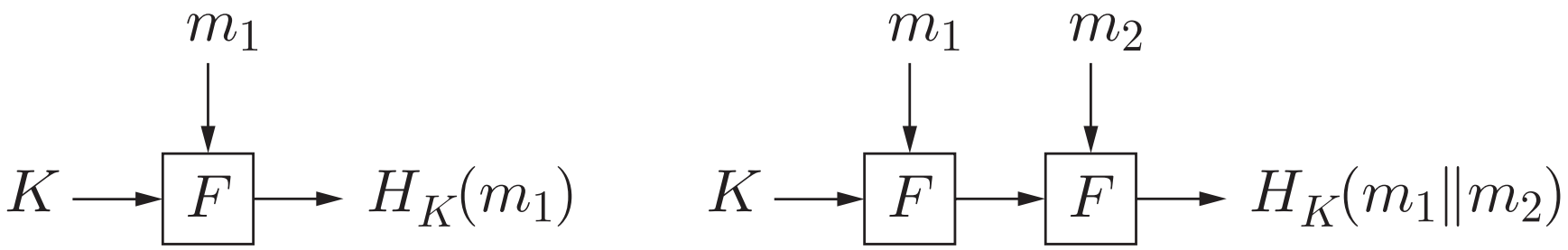
- F が擬似ランダム $\not\Rightarrow$ H が擬似ランダム
  （例えば IV を秘密鍵としたとき）
- F がランダムオラクル $\not\Rightarrow$ H がランダムオラクル

F の複数の性質を同時に保存する繰返し構造は？

反復型ハッシュ関数の問題点

メッセージ拡張攻撃

例）擬似ランダム性で考える

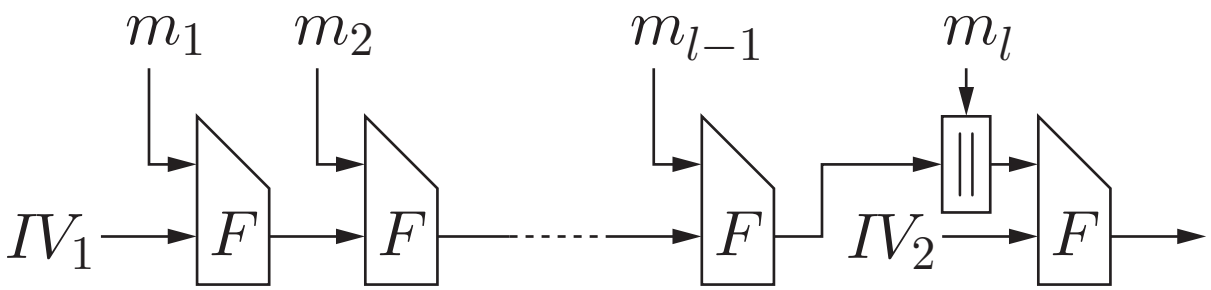


$H_K(m_1)$ を知れば $H_K(m_1 \| m_2)$ が分かるので，擬似ランダムでない．

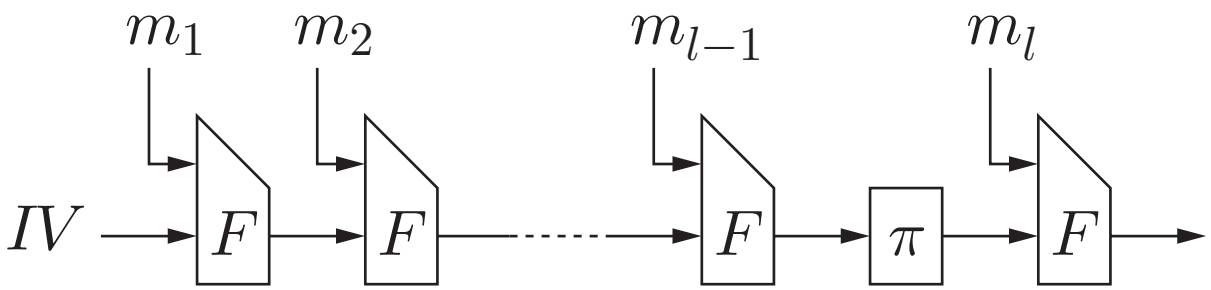
最重要ポイントは，メッセージ拡張攻撃を如何にして防ぐか？

CR, 擬似ランダム, ランダムオラクルを保存する繰返し構造

EMD (Enveloped MD (Merkle-Damgård)) [Bellare, Ristenpart 06]



MDP (MD with a Permutation) [Hirose, Park, Yun 07]



π は置換

注意）MDP では，PRF で F の鍵関連攻撃に対する安全性を仮定

CR ハッシュ関数を利用した暗号方式

H を CR ハッシュ関数とする.

送信者は x をランダムに選び，$y = H(x)$ を受信者に渡す.

- 受信者は**計算能力が無限でも**，x が $H^{-1}(y)$ のどの要素か全くわからない.
- 送信者は計算能力が有限なら，$x' \neq x \wedge y = H(x')$ なる x' を計算できない.

上記の特徴を利用した暗号プロトコルの例

- 故障停止署名 [Damgård, Pedersen, Pfitzmann 93]
- 非対話コミットメント [Halevi, Micali 96]
  計算能力無限の受信者に対して統計的に安全

CR ハッシュ関数を利用したワンタイム故障停止署名

故障停止署名の特長

> 検証に合格する署名の偽造がなされても，正しい署名者は偽造であることを証明できる．

仮定　H は CR ハッシュ関数．各 y について $H^{-1}(y)$ は十分大きい．

秘密鍵と公開鍵の生成

1. x_0, x_1 をランダムに選び，$y_0 = H(x_0)$, $y_1 = H(x_1)$ を計算する．
2. (x_0, x_1) を秘密鍵，(y_0, y_1) を公開鍵とする．

署名の生成　メッセージ $b \in \{0, 1\}$ に対する署名は x_b

CR ハッシュ関数を利用したワンタイム故障停止署名

メッセージ $c \in \{0,1\}$ に対する偽造署名 x'_c が現れた場合

- $y_c = H(x'_c)$（x'_c は検証に合格する）
- 署名者は x_c を示して偽造を証明する.

偽造であることの証明に成功する確率

$$\Pr[x'_c \neq x_c] = 1 - \frac{1}{|H^{-1}(y_c)|} \approx 1$$

CR ハッシュ関数を利用した単純な系列コミットメント方式

仮定　H は CR ハッシュ関数

系列 x へのコミットの単純な方法

コミット 送信者は $y = H(x)$ を計算して，y を受信者に渡す.

開示 送信者は x を受信者に渡す.

検証 受信者は $y = H(x)$ が成立するかどうかを確認する.

問題点）y から，$x \in H^{-1}(y)$ という情報が漏れる.

Halevi-Micali 系列コミットメント方式

仮定

- $H : \{0,1\}^* \to \{0,1\}^\ell$ は CR ハッシュ関数
- $F = \{f \mid f : \{0,1\}^{O(n+\ell)} \to \{0,1\}^n\}$ はユニバーサルハッシュ関数族

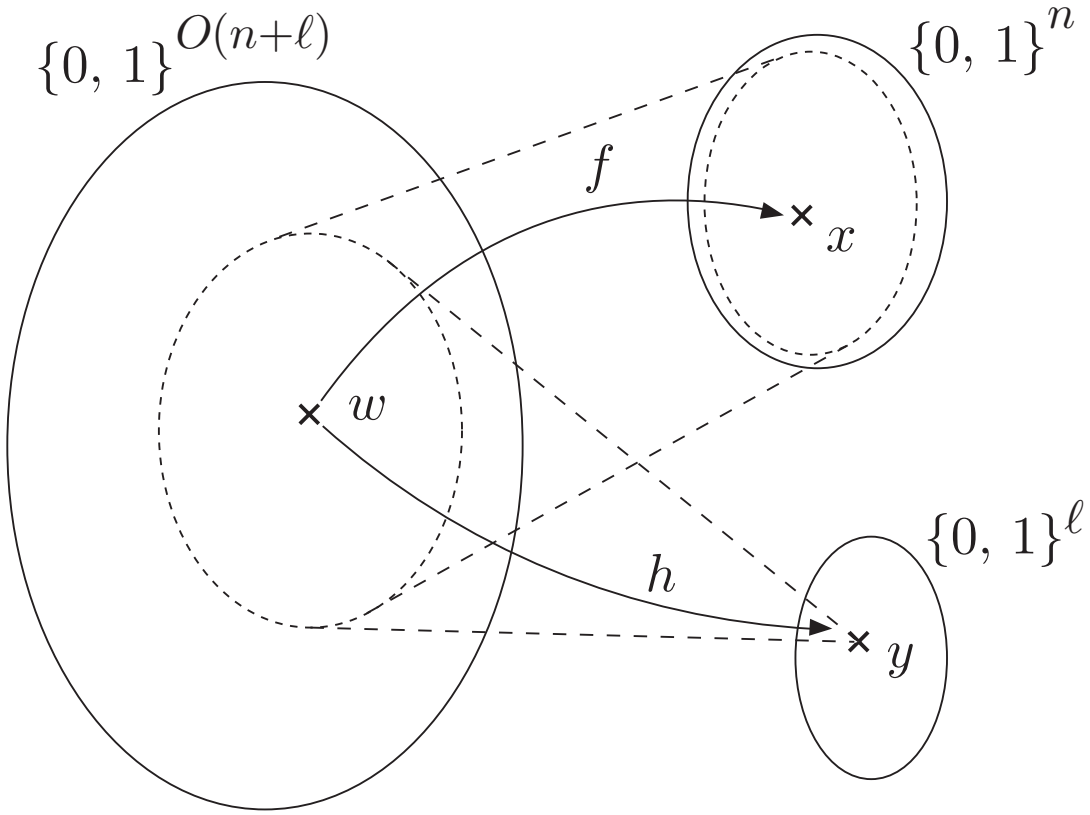
コミット 送信者は系列 $x \in \{0,1\}^n$ に対して以下を実行する.

1. $x = f(w)$ を満たす $f \in F$ と $w \in \{0,1\}^{O(n+\ell)}$ を乱択する.
2. $y = H(w)$ を計算する.
3. f, y を受信者に送る.

開示 送信者は w を受信者に渡す.

Halevi-Micali 系列コミットメント方式

受信者の計算能力が無制限であっても，統計的に安全



コミットで f, y を得ても，x に関する情報はほとんど得られない．

まとめ

ハッシュ関数の応用は多岐に渡り，要求される性質も様々

今回取り上げていない主な話題

- SHA-2 族，新しいハッシュ関数の提案
- 専用ハッシュ関数による HMAC (NMAC) に対する攻撃
- ランダム化ハッシュモード

ハッシュ関数の研究が盛んになりつつある．