ISEC チュートリアル講演会「暗号技術の証明可能安全性」(2005/5/17, 機械振興会館)

ハッシュ関数の証明可能安全性

廣瀬勝一

福井大学工学部電気・電子工学科

2005 年 5 月 17 日

暗号ハッシュ関数

$h : \{0,1\}^l \to \{0,1\}^\ell,\ l > \ell$

- Preimage resistance
  ランダムに与えられた出力 y について，$h(x) = y$ を満たす入力 x を計算するのが困難．

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

- Collision resistance
  $h(x) = h(x') \wedge x \neq x'$ **を満たす入力** $x,\ x'$ **を計算するのが困難．**

universal one-way **は後述**

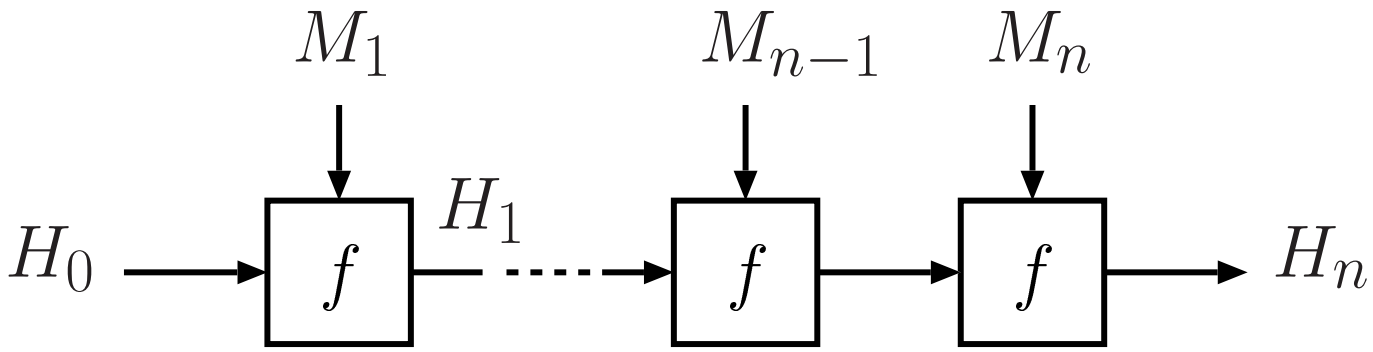
内容

- CR **ハッシュ関数の証明可能安全性**
- CR vs. (U)OW
- CR **ハッシュ関数を利用した暗号方式**

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** $H_0 \in \{0,1\}^{\ell}$

入力 $M = (M_1, M_2, \ldots, M_n),\ M_i \in \{0,1\}^{\ell'}\ (1 \le i \le n)$

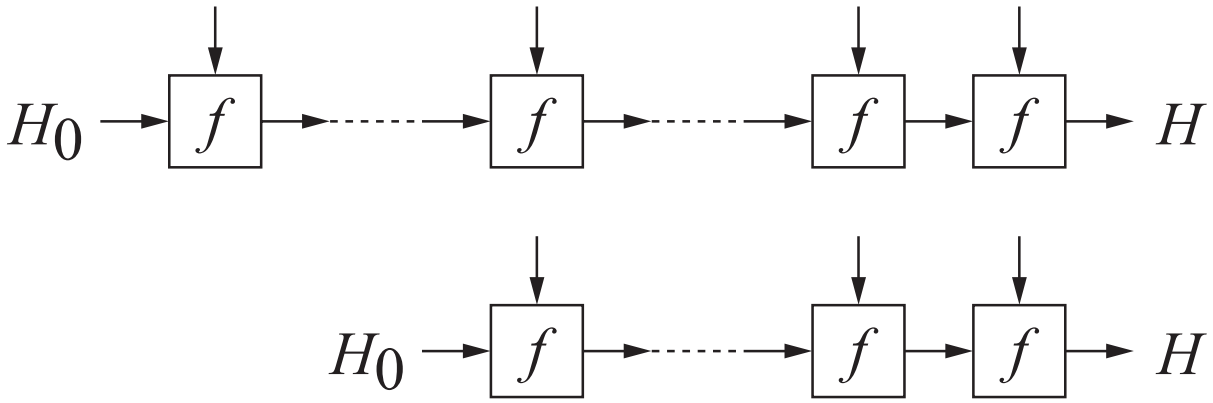


$h(M) = H_n$

反復型ハッシュ関数の CR

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

ブロック暗号に基づくハッシュ関数の証明可能安全性

単ブロック長（出力長がブロック暗号のブロック長と等しい）

- Winternitz 84
  $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$，PR について最適
- Merkle 89
  $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$，CR について最適
- Black, Rogaway, Shrimpton 02
  Preneel-Govaerts-Vandewalle 方式，CR について最適

CR について最適

どのような攻撃も，誕生日攻撃と同等の効果しかもたない

ブロック暗号に基づくハッシュ関数の証明可能安全性

倍ブロック長

- Merkle 89
  圧縮関数 $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$ **を用いた構成，**CR **について最適，**
  低レート $(< 0.276\cdots)$
- Knudsen, Preneel 96, 97, 02
  ランダムな圧縮関数を利用した構成，誤り訂正符号に基づく構成，
  CR **について最適ではない**
- Hirose 04
  $(m, 2m)$ **ブロック暗号，**CR **について最適，レート** $1/2$
- Nandi, Lee, et. al. 04
  CR **について最適ではない，レート** $2/3$

ブロック暗号のブラックボックスモデル

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．

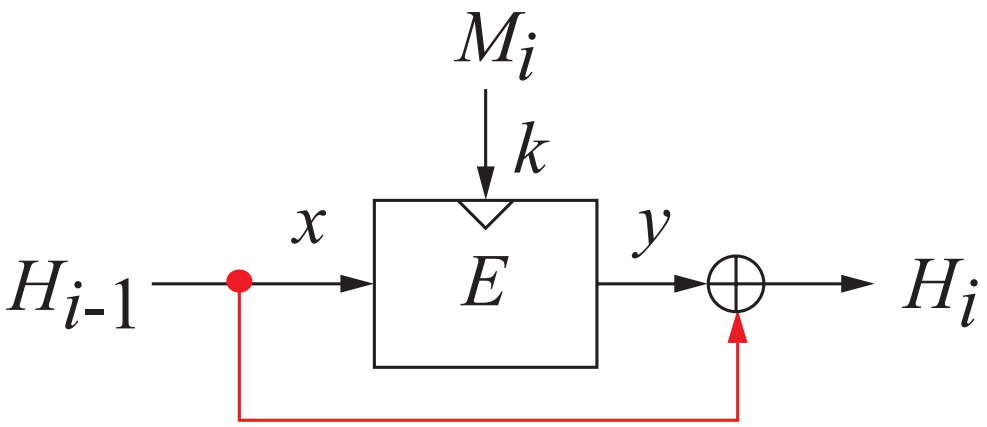
オラクル	質問	返答
暗号化 E	(鍵, 平文)	暗号文
復号 D	(鍵, 暗号文)	平文

- 各鍵について，E，D は 1 対 1 関数
- E，D 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**

Merkle 89

定理　圧縮関数 $H_i = E_{M_i}(H_{i-1}) \oplus H_{i-1}$ **は** CR **について最適**



圧縮関数を計算するためには，E **か** D **かを計算しなければならない．**

$$H_i = E_k(x) \oplus x \text{ または } H_i = y \oplus D_k(y)$$

ブラックボックスモデルでは，圧縮関数の出力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．

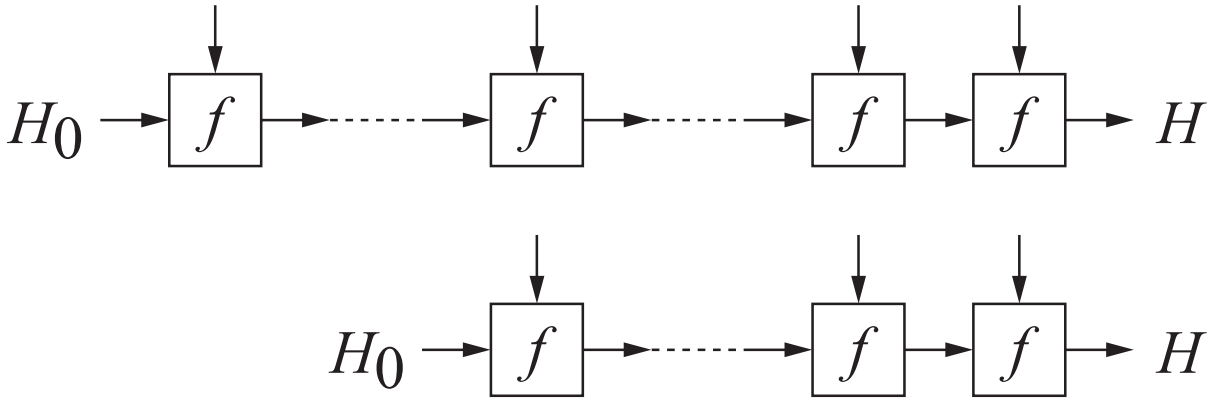
Merkle 89

圧縮関数 f が CR $\Rightarrow$ ハッシュ関数 h が CR

定理　h **を圧縮関数** f **の反復型ハッシュ関数とすると，**

$$\mathbf{Adv}_h^{\mathrm{coll}}(q) \leq \mathbf{Adv}_f^{\mathrm{coll}}(q)$$

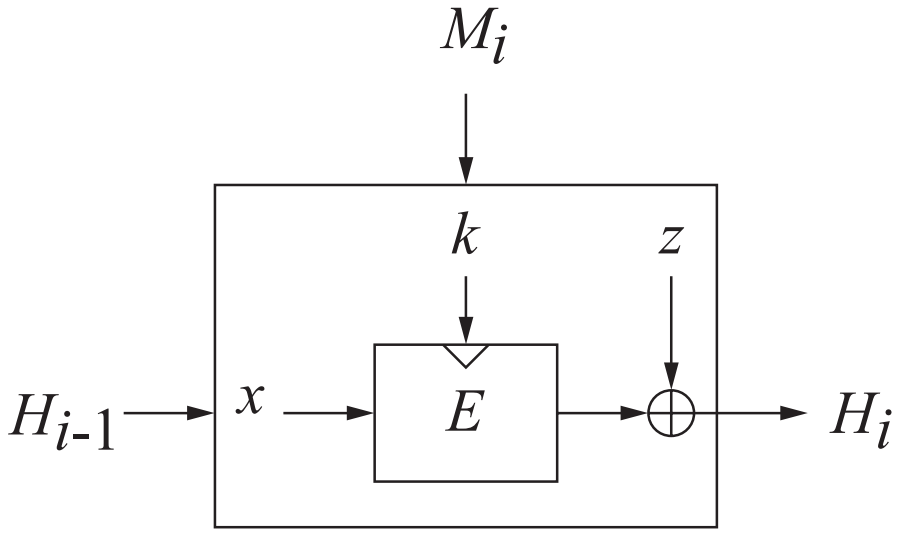
$\mathbf{Adv}_h^{\mathrm{coll}}(q)$ **は，質問回数** q **の攻撃で** h **の衝突が見つかる確率の最大値**



Black, Rogaway, Shrimpton 02

Preneel-Govaerts-Vandewalle **方式**

$x, k, z \in \{H_{i-1}, M_i, H_{i-1} \oplus M_i, 0\}$

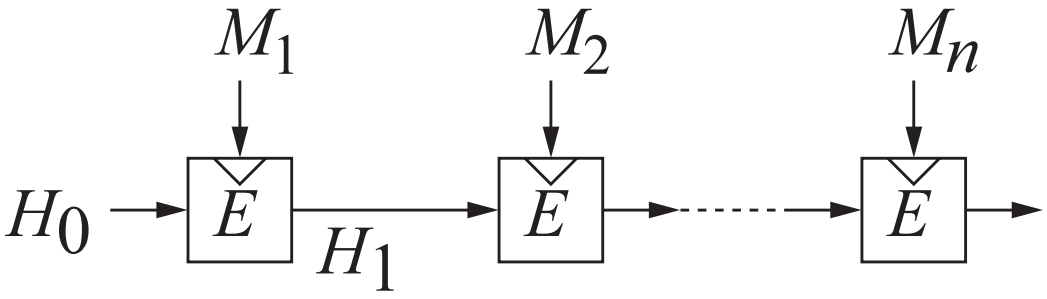


- CR **に関して最適な圧縮関数が** 12 **個存在する．**
  Merkle 89 **と同様の方法**
- **圧縮関数が** CR **に関して最適でなくても，**CR **に関して最適なハッシュ関数が** 8 **個存在する．**

Black, Rogaway, Shrimpton 02

圧縮関数が CR に関して最適でなくても，CR に関して最適なハッシュ関数が 8 個存在する．

例

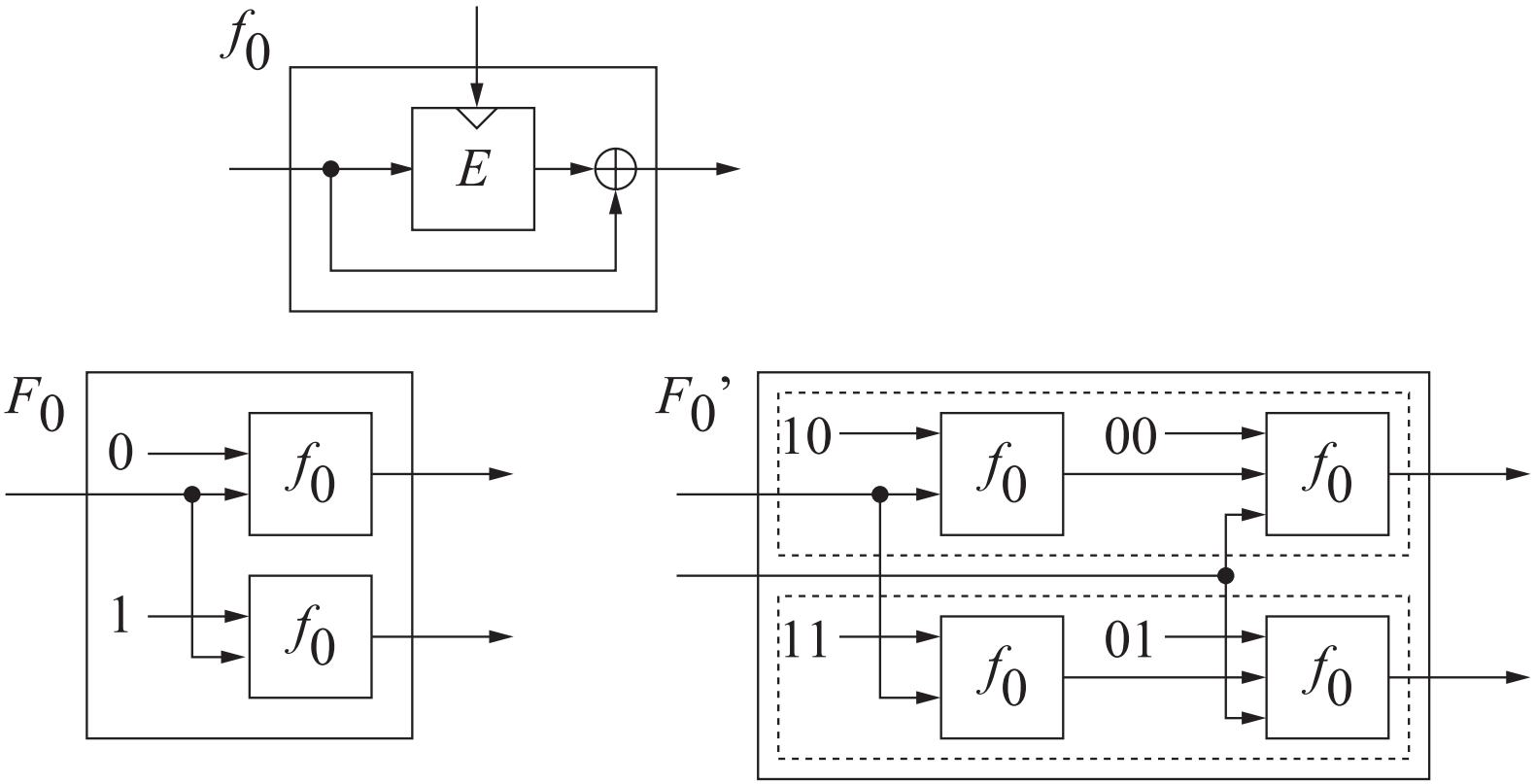


インフォーマルな解析では…

- **遡行攻撃しても，$H_0 = E_{M_1}(H_1)$ なる M_1 を見つけるのは困難**
- **中間一致攻撃の成功確率は，誕生日攻撃と同等**

Merkle 89

圧縮関数 f_0 を用いた倍ブロック長ハッシュ関数の構成



f_0 を用いて，**入力が同一の互いに独立な圧縮関数を二つ作る**．

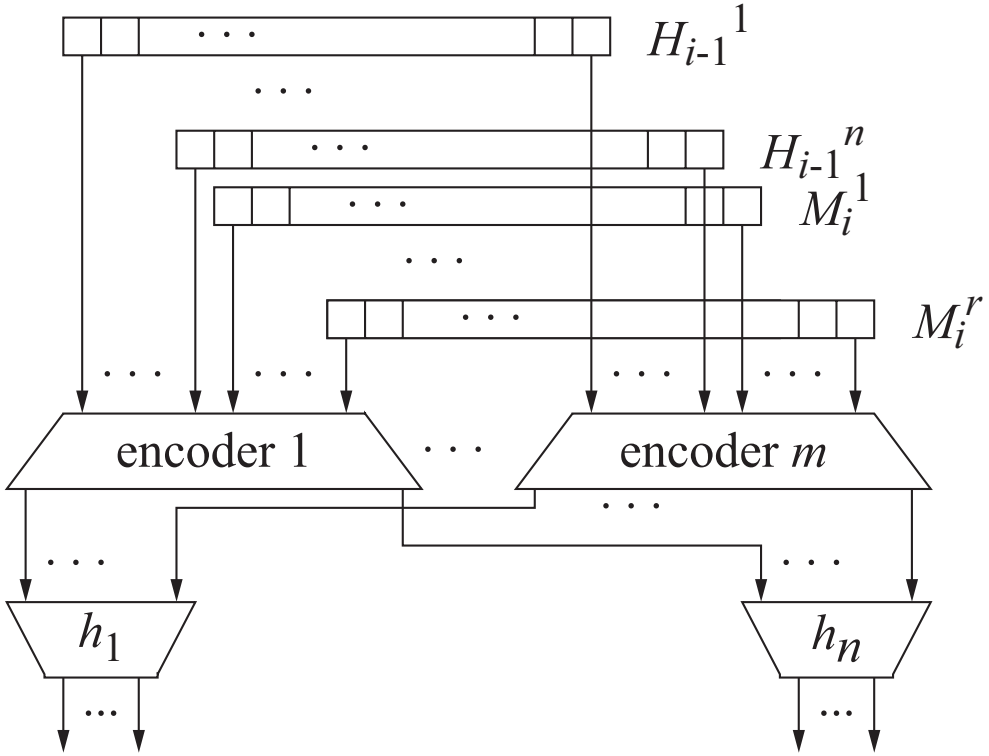
F_0' は F_0 よりレートが大きい．

Knudsen, Preneel 96, 97, 02

$h_1, \ldots, h_n$ **は真にランダムな圧縮関数**

encoder 1, . . . , encoder m **が最小距離** d **の符号化器**

入力 $H_{i-1}^1, \ldots, H_{i-1}^n, M_i^1, \ldots, M_i^r$ **を変化させると，** $h_1, \ldots, h_n$ **の内，** d **個以上の入力が変化する．**



CR と一方向性とは本質的に異質 [Simon 98]

以下のようなオラクル $\mathcal{O}$ の存在することが示された．

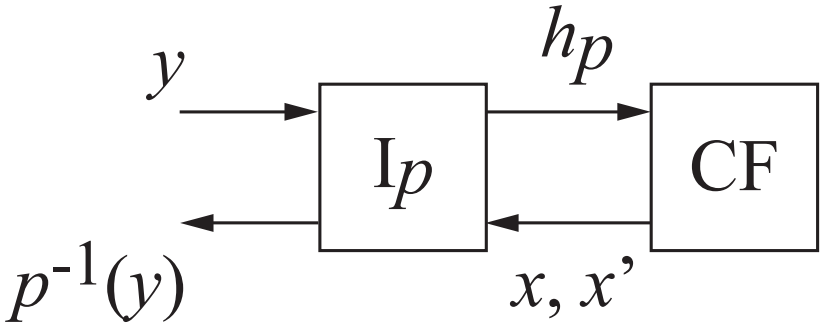
$\mathcal{O}$ の下で一方向置換は存在するが，CR ハッシュ関数は存在しない．

したがって，

「任意の一方向置換 p を用いて，CR ハッシュ関数を構成できる」は偽．

もしこれが証明できると仮定すると，

与えられたハッシュ関数の衝突を返すオラクルを用いて，p の原像を効率よく計算するアルゴリズムが構成できる．



CR と UOW

暗号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]
  始めに攻撃者が入力 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．

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．

UOW ハッシュ関数の構成法

- **一方向置換を用いた構成法** [Naor, Yung 89]
- **一方向関数を用いた構成法** [Rompel 90]

CR と UOW **とは本質的に異質**

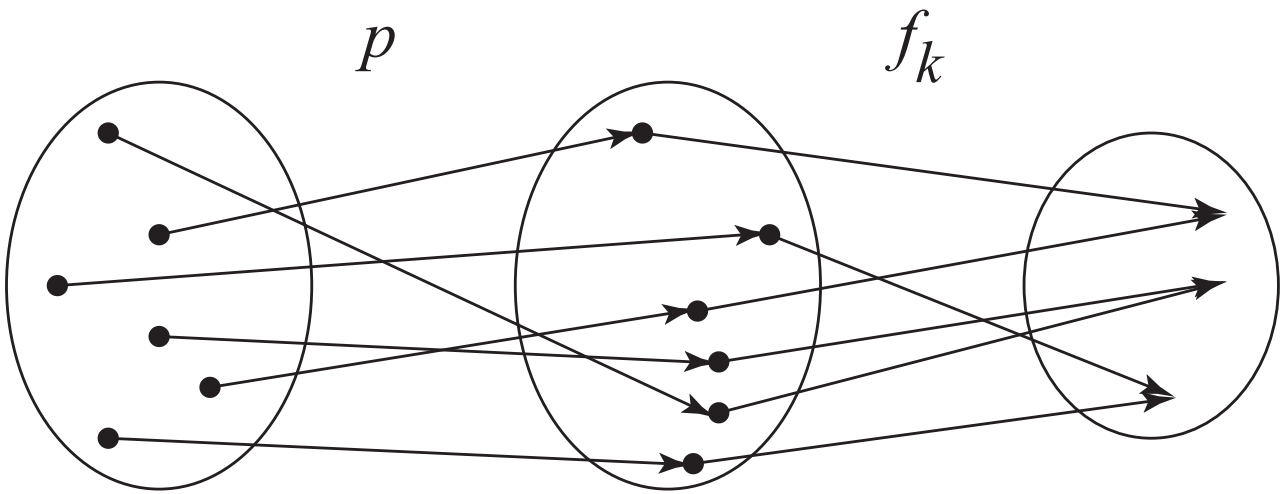
一方向置換を用いた UOW ハッシュ関数の構成法 [Naor, Yung 89]

$p : \{0,1\}^\ell \to \{0,1\}^\ell$ **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の集合** $\Rightarrow$ H **は** UOW

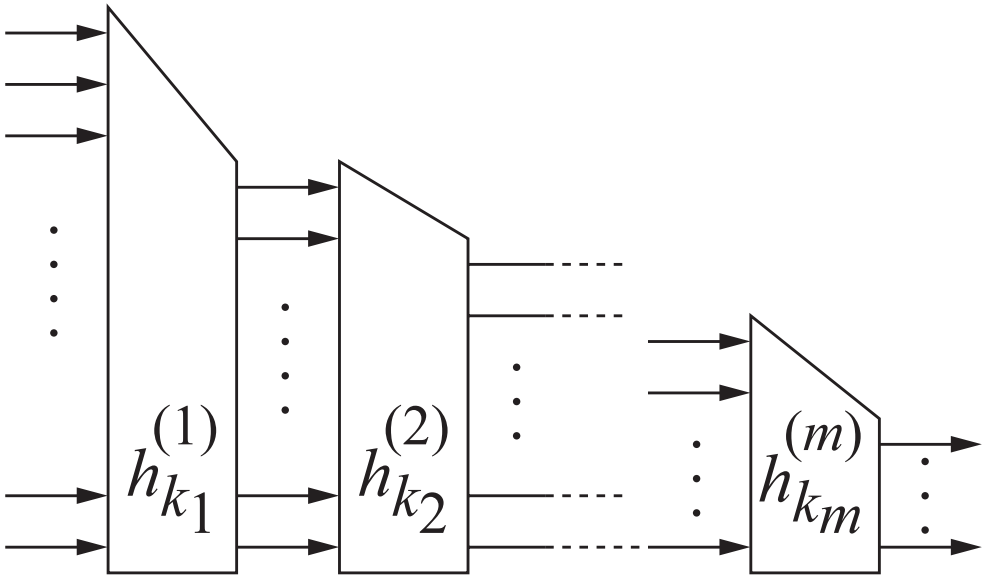


一方向置換を用いた UOW ハッシュ関数の構成法 [Naor, Yung 89]

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

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

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

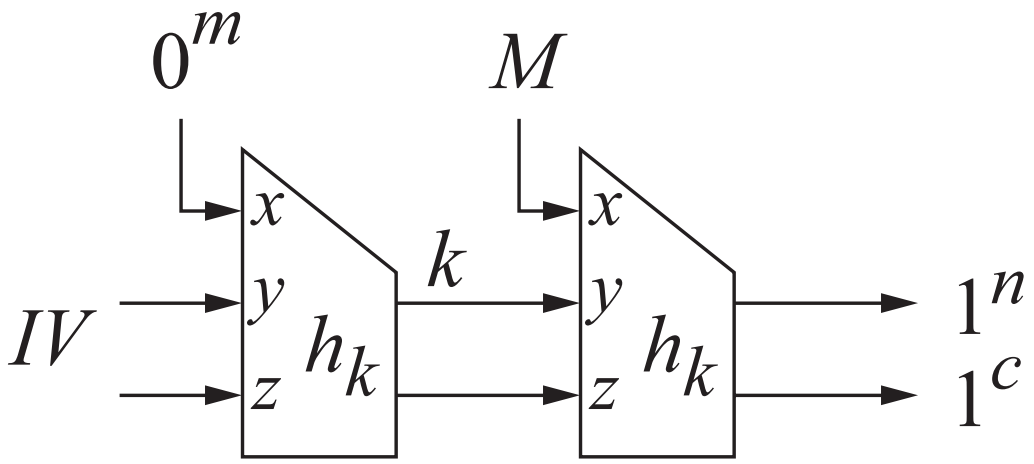


任意長入力の UOW ハッシュ関数

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

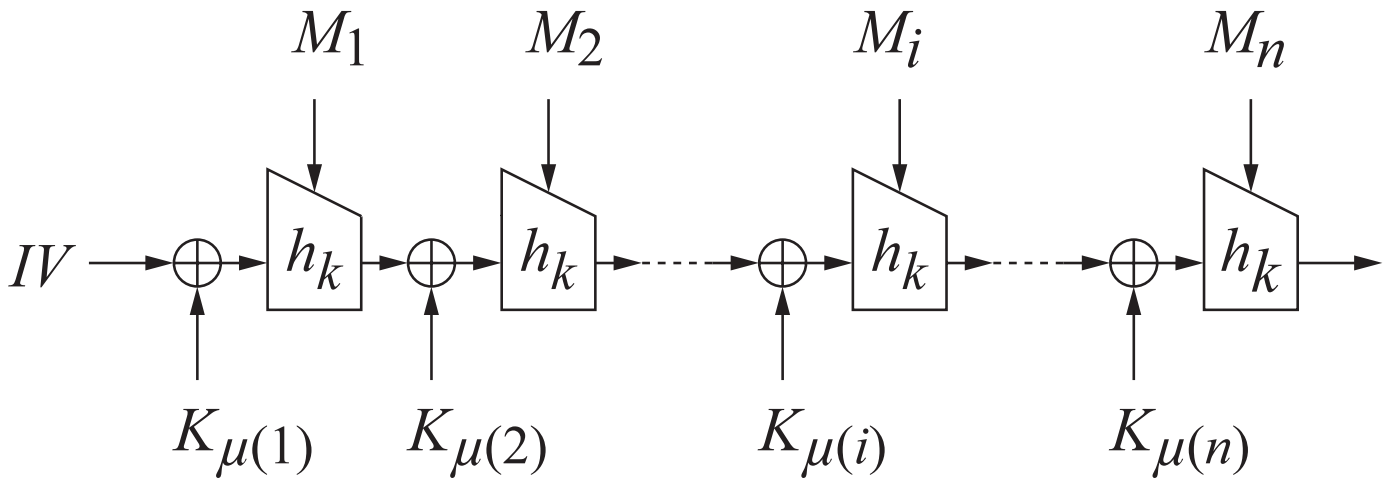
任意長入力の UOW ハッシュ関数

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

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]



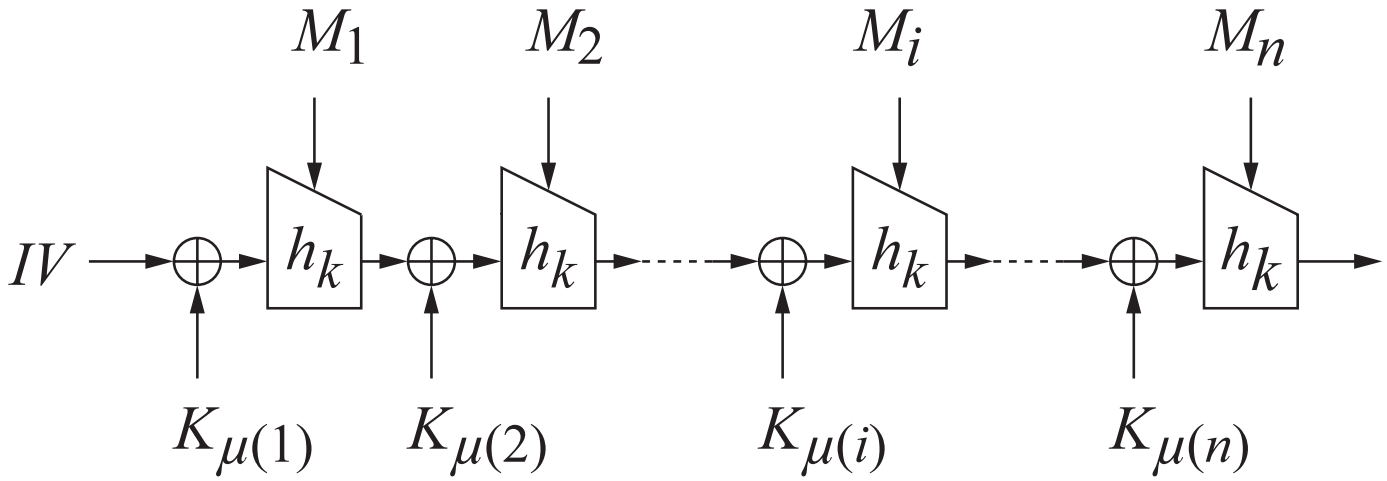
$\mu(i) =$ largest integer μ such that $2^\mu | i$

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

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 $\Rightarrow |\mu(\{1, 2, \ldots, n\})| > \log n$

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．

S は x をランダムに選び，$y = h(x)$ を R に渡す．

- R 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．
- S 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]
- **計算能力無限の受信者に対して統計的に安全な非対話コミットメント** [Halevi, Micali 96]

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　S は $y = h(x)$ を計算して，y を R に渡す．

開示　S は x を R に渡す．

検証　R は $y = h(x)$ が成立するかどうかを確認する．

y からは，$x \in h^{-1}(y)$ という情報が漏れる．

CR ハッシュ関数を利用した系列コミットメント方式 [Halevi, Micali 96]

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の集合

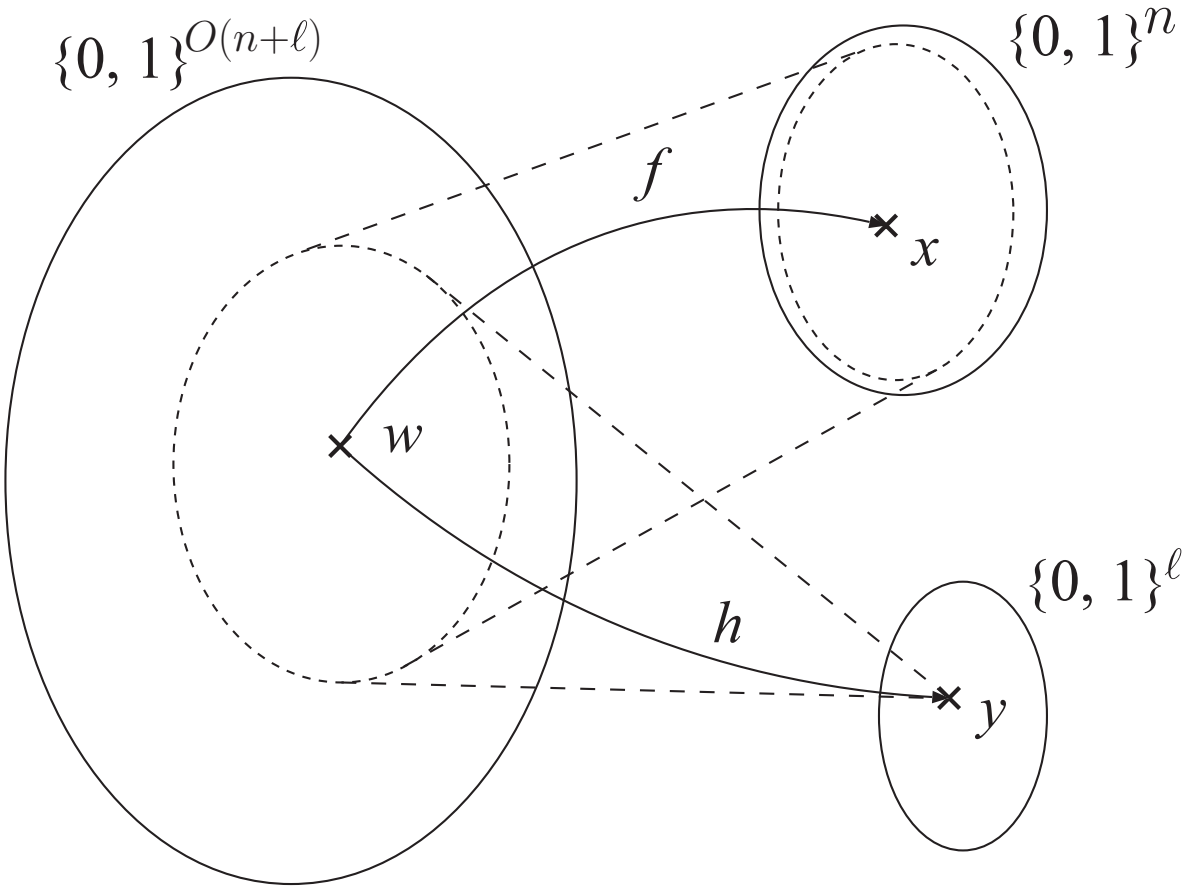
コミット　S は系列 $x \in \{0,1\}^n$ に対して以下を実行する．

1. $x = f(w)$ を満たすように $f \in F$ と $w \in \{0,1\}^{O(n+\ell)}$ をランダムに選ぶ．
2. $y = h(w)$ を計算する．
3. f, y を R に送る．

開示　S は w を R に渡す．

CR ハッシュ関数を利用した系列コミットメント方式 [Halevi, Micali 96]

R 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

- CR **ハッシュ関数の証明可能安全性**
- CR vs. (U)OW
- CR **ハッシュ関数を利用した暗号方式**