lecture 17



# HF Based on Block ciphers

- HF based on Block ciphers
- Customized hash functions
- HF based on Modular Arithmetic

**Benefit** → There are lots of efficient method to implement them either by hardware / software implementation.

**Def:** An (n,r) Block cipher is a block cipher defining an invertible function from n-bit plaintexts to n-bit ciphertexts using an r-bit key → $E_k(x)$

n-bit → 64 ← DES → 56 ← r-bit key

**Def:** Let "h" be an iterated hash function constructed from a block cipher, with compression function f which performs s block encryptions to process each successive n-bit message block. Then the rate is $1/s$.

1. single-length MDCs of rate 1 → key size is almost the same as blocklength size
2. Double-length MDCs

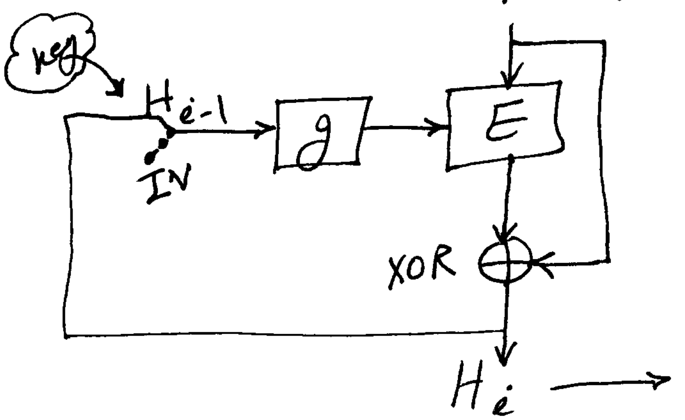
→ unkeyed hash functions

## ① Single-length MDCs of rate 1

requirements
- ⓐ a generic n-bit block cipher $E_K$
- ⓑ a function "g" which maps n-bit inputs to keys K
- ⓒ a fixed (usualy n-bit) initial value IV

### 1-1 Matyas-Meyer-Oseas

$x_i: x_1\, x_2 \dots x_t$ → t blocks of n-bit (message)

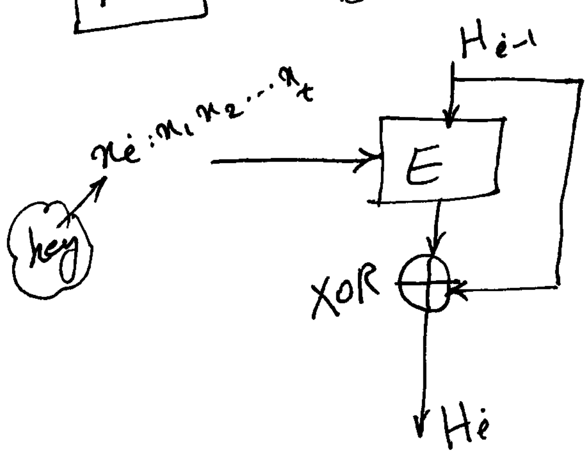



compressed

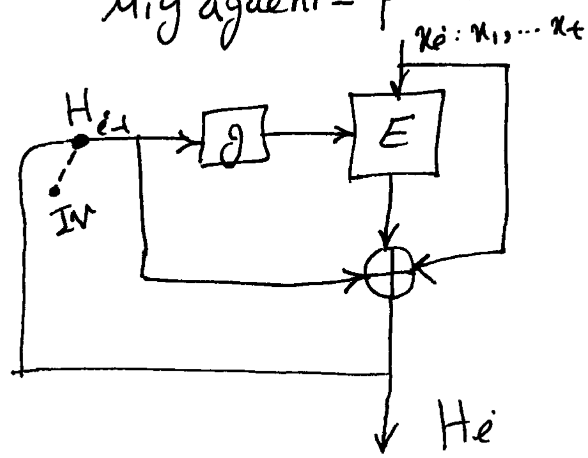
$H_i$ → last ciphertext is a block of n-bit (hash vale)

### 1-2 Davies-Meyer

$H_0 = IV$




### 1-3 Miyaguchi-preneel



## 1-1 alg. M-M-O

Input: bitstring $x \xrightarrow{x_i} x_1\ x_2 \cdots x_t$

output: n-bit hash code of $x$

1. $x$ is divided into n-bit blocks & padded. you predefine IV.

2. $H_t$ : $H_0 = IV$

plaintext

$$H_i = E_{g(H_{i-1})}(x_i) \oplus x_i \qquad 1 \leq i \leq t$$

Enc scheme　　key

## 1-2 alg. D-M.

"

"

"

2. $H_t$ : $H_0 = IV$

$$H_i = E_{x_i}(H_{i-1}) \oplus H_{i-1} \qquad 1 \leq i \leq t$$

## 1-3 alg. M-P

"

"

2. $H_t$ : $H_0 = IV$

$$H_i = E_{g(H_{i-1})}(x_i) \oplus x_i \oplus H_{i-1}$$

$$1 \leq i \leq t$$

## ② Double-length $MDC_s$:

requirements

ⓐ DES bitlength = 64 key size 56

ⓑ $g$ & $\tilde{g}$ which maps 64-bit to 56-bit

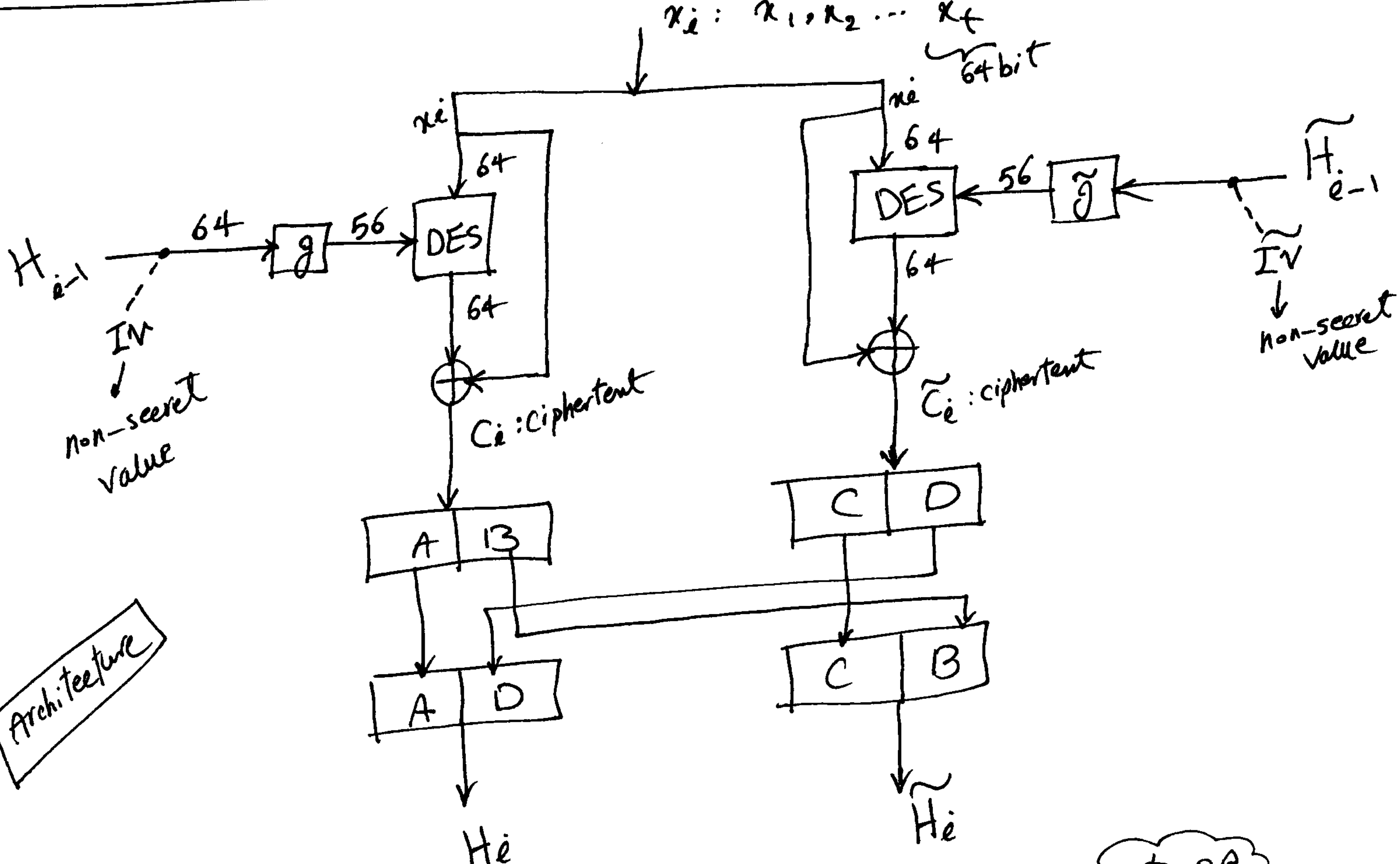
$$g(U) = u_1\, 1\, 0\, u_4 \cdots u_7\, \cancel{u_8}\, u_9 \cdots u_{63}$$

$$\tilde{g}(U) = u_1\, 0\, 1\, u_4 \cdots u_7\, \cancel{u_8}\, u_9 \cdots u_{63}$$

(exclude $u_8, u_{16}, u_{24}, \ldots u_{64}$)

---

Architecture




not OR

$$H_t = H_t \,\|\, \tilde{H}_t \longrightarrow \text{Concatination}$$

(128-bit hash value)

---

$\boxed{\text{algorithm}}$

$H_0 = IV \qquad k_i = g(H_{i-1}) \qquad C_i = E_{k_i}(x_i) \oplus x_i$

$\tilde{H}_0 = \tilde{IV} \qquad \tilde{k}_i = g(\tilde{H}_{i-1}) \qquad \tilde{C}_i = E_{\tilde{k}_i}(x_i) \oplus x_i$

$H_i = C_i^L \,\|\, \tilde{C}_i^R$ (A) (D) $\qquad \tilde{H}_i = \tilde{C}_i^L \,\|\, C_i^R$ (C) (B)