lecture 18

MAC- Based on Block Ciphers

CBC-MAC Algorithm

Input: data x, Block Cipher E, secret MAC key for E

output: n-bit MAC on x (n is the block size of E)

1. Padding & blocking: pad x if nessary, divide the padded text into n-bit blocks denotated $x_1, \dots x_t$

2. CBC processing. E_k is our Encryption scheme with k

   Compute H_t as follows:

$$H_1 \leftarrow E_k(x_1) \qquad IV = 0$$

$$H_i \leftarrow E_k(H_{i-1} \oplus x_i) \quad \text{for } 2 \leq i \leq t$$

optional → 3. optional process to increase the strength of MAC.

Second secret key → $k' \neq k$

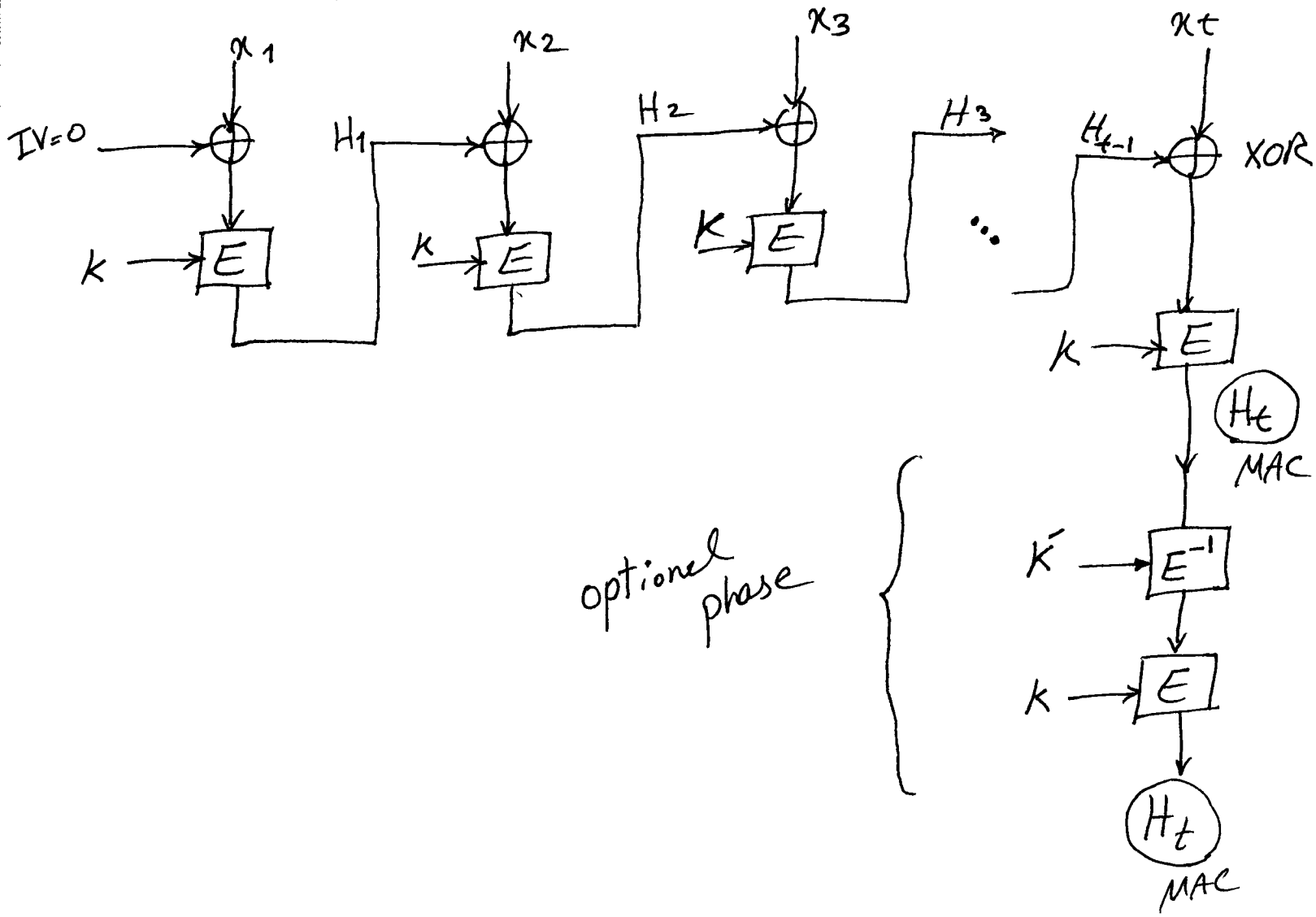
$$H'_t \leftarrow E^{-1}_{k'}(H_t)$$

$$H_t \leftarrow E_k(H'_t)$$

(original key)

4. n-bit block H_t (hash value)

Architecture



Note: MAC is different than MDC (Mod Detection Code)

MAC: Integrity & Authentication
* secret key

MDC: Integrity
* unkeyed

MASH: Modular Arithmetic Secure HASH algo.

1995 →

Input: data x of bitlength $0 \leq b < 2^{n/2}$

output: n-bit hash of x (n is almost the bitlength of Modulus M)

1. Setup: ⊛ select $M = pq$ of bitlength "m"

p, q are primes & factorization is hard

⊛ the bitlength "n" of the hash-result should be the largest multiple of 16 less than "m"

$n = 16n' < m$ → 16 32 48 64 ....

↓

m=64-bit

$n = 48$

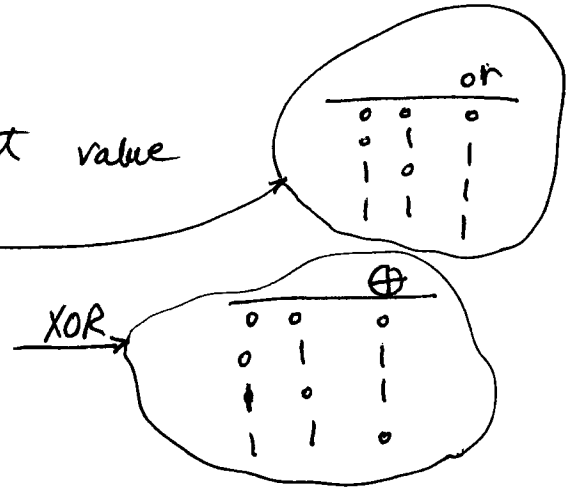
$0 \leq b < 2^{24}$

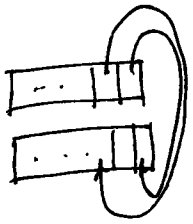
⊛ $H_0 = 0$ as IV

$A = 0x\,f0\ldots0$ Constant value

$\vee$ → inclusive or

$\oplus$ → exclusive or





bitwise operations

2. padding & Blocking

pad x with 0-bits, if necessary to obtain

$t \cdot \frac{n}{2}$-bit input ← new size for "x"

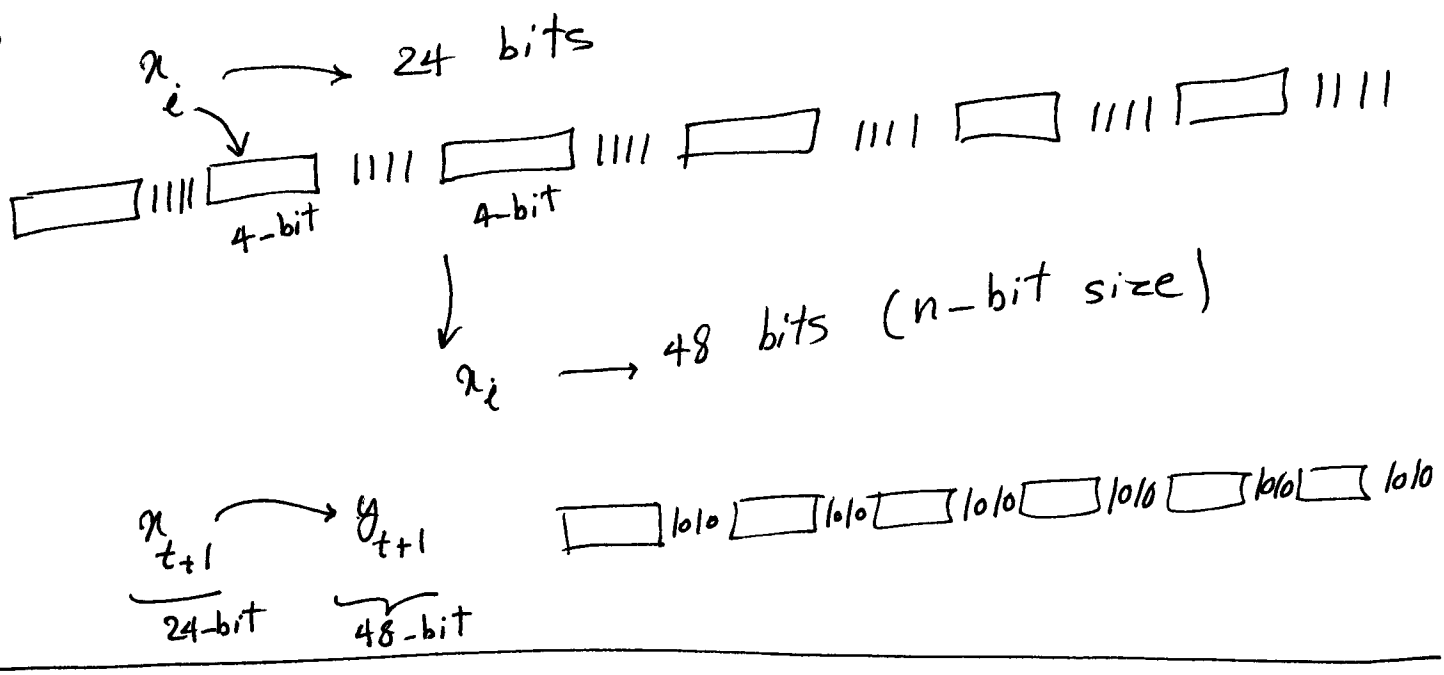
for smallest value of "t"

then, divide x into t blocks of size $\frac{n}{2}$ → $x_1 \ldots x_t$ (x_1: $\frac{24}{2}$-bit)

Finally, x_{t+1} denotes $\frac{n}{2}$-bit representation of b

3. Expand x_i to an n-bit block y_i by partitioning it into 4-bit nibbles and inserting four 1-bits preceding each, except for y_{t+1} in which the inserted nibble is 1010 (not 1111).

$\frac{n}{2} = 24$



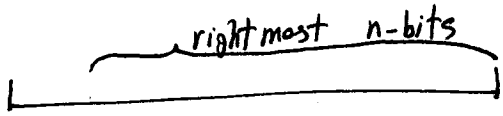
---

4. Compression function. $1 \leq i \leq t+1$

map 2 n-bit inputs (H_{i-1}, y_i) to one n-bit

$$H_i \leftarrow \left(\left((H_{i-1} \oplus y_i) \vee A \right)^2 \bmod M \right) \dashv n \oplus H_{i-1}$$

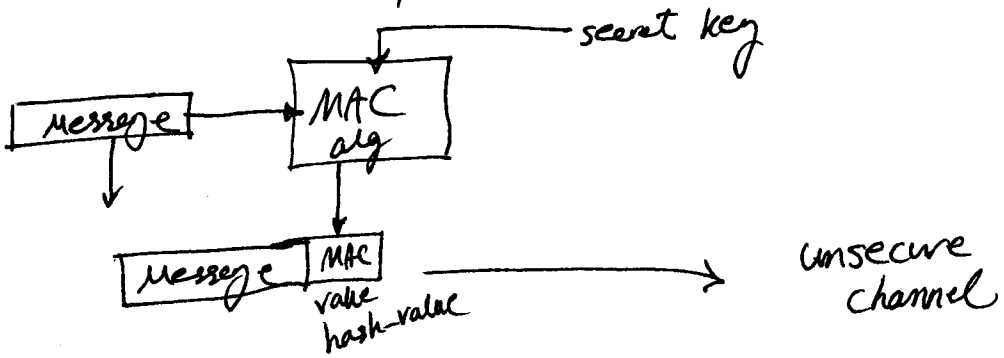
- H_{i-1}: $IV = H_0 = 0$
- y_i: n-bit that you created in the previous stage
- A: constant value
- $\dashv n$: denotes keeping the rightmost n-bits of the m-bit result to its left.



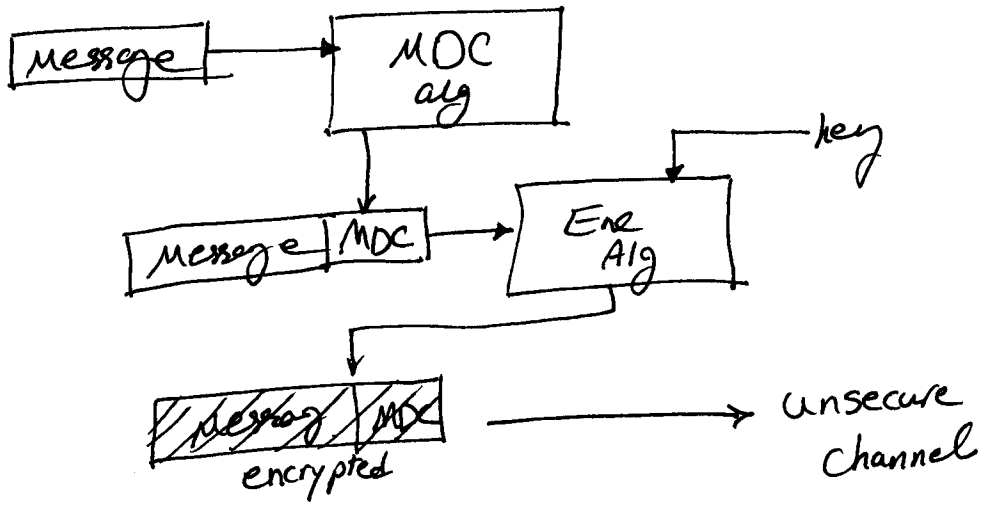
Three methods for DI using HF

DI: Data Integrity
HF: hash Functions

(a) MAC only



(b) MDC & encipherment



(c) MDC & Authentic channel