



THE UNIVERSITY OF BRITISH COLUMBIA

Symmetric Crypto Systems

EECE 412

Module Outline

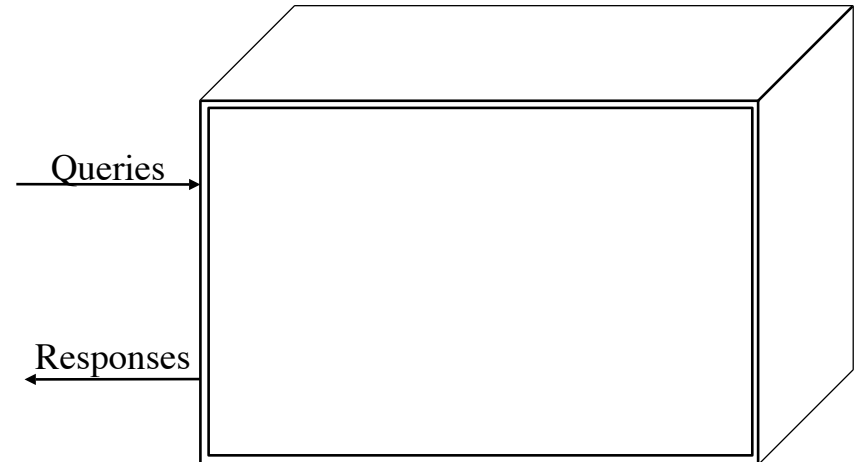
- Stream ciphers “under the hood”
- Block ciphers “under the hood”
- Modes of operation for block ciphers

Stream Ciphers



Random Generator (Stream Cipher) as Random Oracle

- In:
 - short string (key)
 - length of the output



- Out: long random stream of bits (keystream)
- Applications:
 - Communications encryption
 - Storage encryption

Properties

- Should not reuse
 - Use seed



Stream Ciphers

- Not as popular today as block ciphers
- A5/1
 - Designed for hardware implementations
 - Based on shift registers
 - Used in GSM mobile phone system
- RC4
 - Designed for software implementations
 - Based on a changing lookup table
 - Used many places

A5 / 1

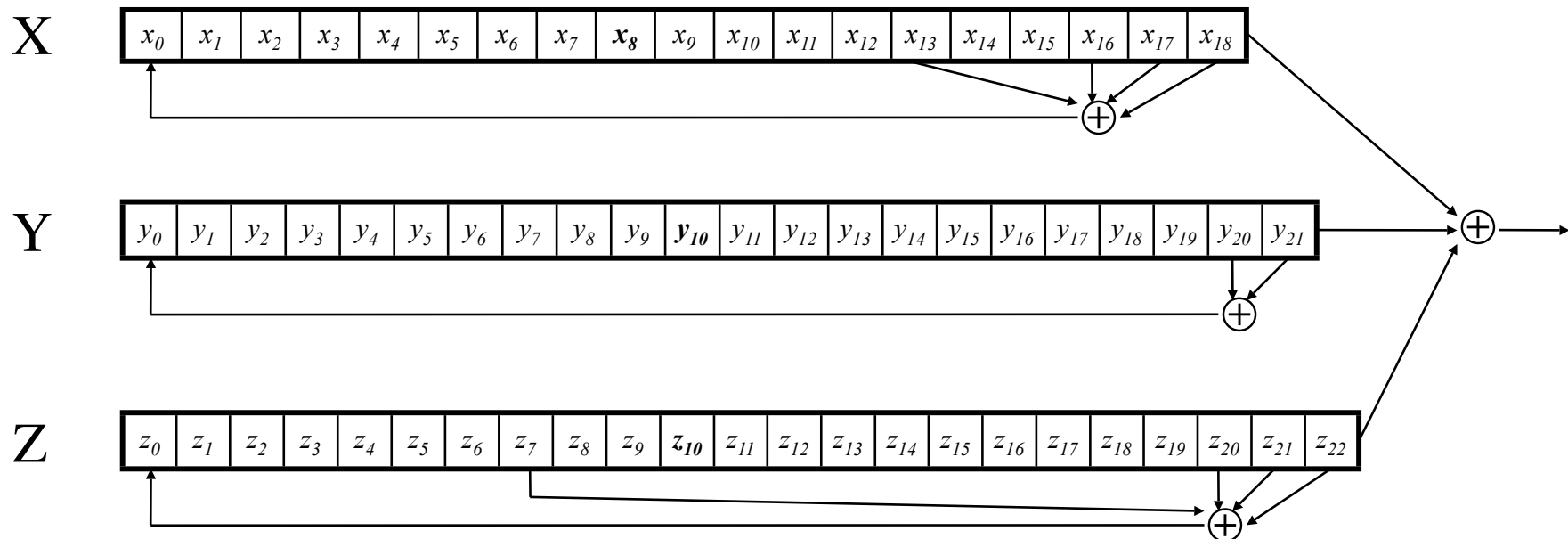
- A5/1 consists of 3 shift registers
 - X: 19 bits ($x_0, x_1, x_2, \dots, x_{18}$)
 - Y: 22 bits ($y_0, y_1, y_2, \dots, y_{21}$)
 - Z: 23 bits ($z_0, z_1, z_2, \dots, z_{22}$)

A5 / 1

- At each step: $m = \text{maj}(x_8, y_{10}, z_{10})$
 - Examples: $\text{maj}(0,1,0) = 0$ and $\text{maj}(1,1,0) = 1$
- If $x_8 = m$ then X steps
 - $t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$
 - $x_i = x_{i-1}$ for $i = 18, 17, \dots, 1$ and $x_0 = t$
- If $y_{10} = m$ then Y steps
 - $t = y_{20} \oplus y_{21}$
 - $y_i = y_{i-1}$ for $i = 21, 20, \dots, 1$ and $y_0 = t$
- If $z_{10} = m$ then Z steps
 - $t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$
 - $z_i = z_{i-1}$ for $i = 22, 21, \dots, 1$ and $z_0 = t$
- Keystream bit is $x_{18} \oplus y_{21} \oplus z_{22}$

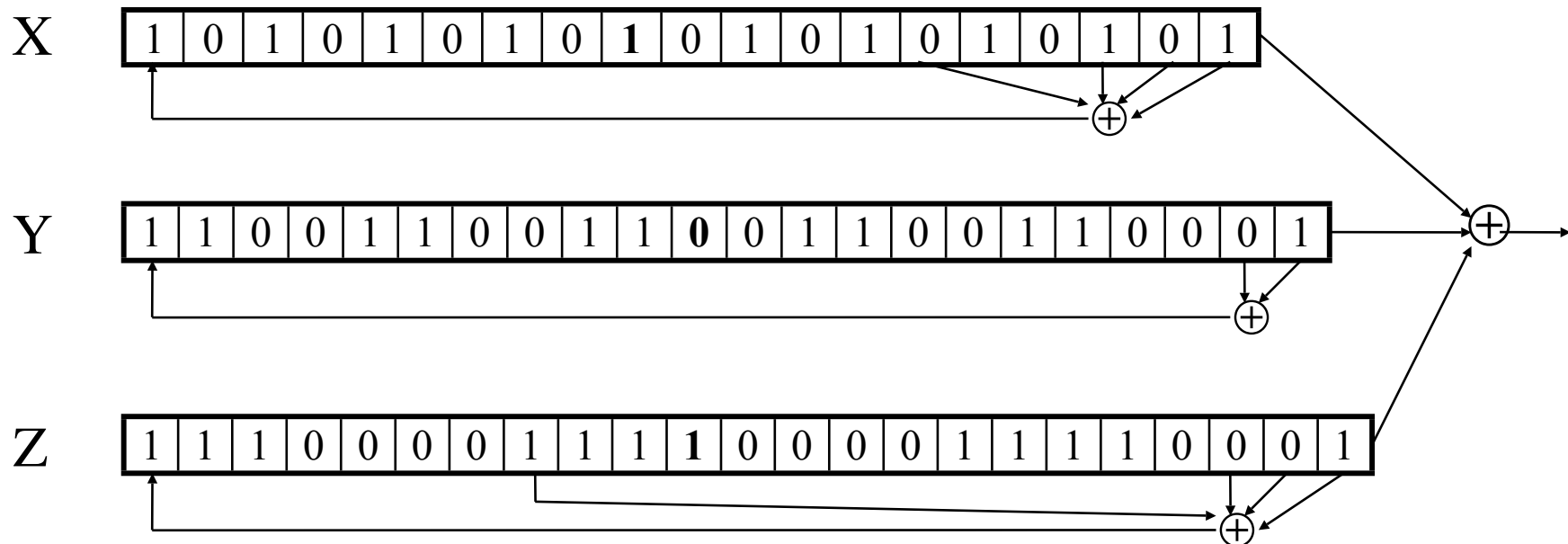


A5 / 1



- Each value is a single bit
- Key is used as **initial fill** of registers
- Each register steps or not, based on (x_8, y_{10}, z_{10})
- Keystream bit is XOR of right bits of registers

A5/1: example



- In this example, $m = \text{maj}(x_8, y_{10}, z_{10}) = \text{maj}(1, 0, 1) = 1$
- Register X steps, Y does not step, and Z steps
- Keystream bit is XOR of right bits of registers
- Here, keystream bit will be $0 \oplus 1 \oplus 0 = 1$

Shift Register Crypto

- Shift register-based crypto is efficient in hardware
- Harder to implement in software
- In the past, very popular
- Today, more is done in software due to faster processors
- Shift register crypto still used some

Use of Stream Ciphers

- Stream ciphers were big in the past
 - Efficient in hardware
 - Speed needed to keep up with voice, etc.
 - Today, processors are fast, so software-based crypto is fast enough



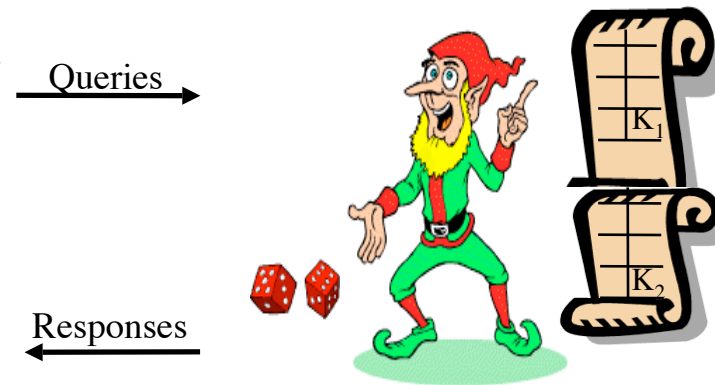
THE UNIVERSITY OF BRITISH COLUMBIA

Block Ciphers “Under the Hood”

Random Permutation (Block Cipher) as Random Oracle

- In

- fixed size short string (plaintext) M ,
 - DES -- 64 bits
- Key K



- Out

- same fixed size short string (ciphertext) C

Notation

- $C = \{ M \}_K$
- $M = \{ C \}_K$

Related Notes

- Main properties of block ciphers
 - invertible
 - confusing
 - diffusing
- Main block ciphers
 - Data Encryption Standard (DES)
 - Advanced Encryption Standard (AES) a.k.a., Rijndael

(Iterated) Block Cipher

- Plaintext and ciphertext consists of fixed sized blocks
- Ciphertext obtained from plaintext by iterating a **round function**
- Input to round function consists of key and the output of previous round
- Usually implemented in software

Feistel Cipher

- type of block cipher design, not a specific cipher
- Split plaintext block into left and right halves:

$$\text{Plaintext} = (L_0, R_0)$$

- For each round $i=1, 2, \dots, n$, compute

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

where F is **round function** and K_i is **subkey**

- Ciphertext = (L_n, R_n)



Feistel Cipher

- Decryption: Ciphertext = (L_n, R_n)
- For each round $i=n, n-1, \dots, 1$, compute
$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus F(R_{i-1}, K_i)$$
where F is round function and K_i is subkey
- Plaintext = (L_0, R_0)
- Formula “works” for any function F
- But only secure for certain functions F
 - silly round function example: $F(x, y) == 0$ for any x and y .



Advanced Encryption Standard

- Replacement for DES
- AES competition (late 90's)
 - NSA openly involved
 - Transparent process
 - Many strong algorithms proposed
 - Rijndael Algorithm ultimately selected
 - Pronounced like "Rain Doll" or "Rhine Doll"
 - invented by Joan Daemen and Vincent Rijmen
- Iterated block cipher (like DES)

AES Overview

- **Block size:** 128, 192 or 256 bits
- **Key length:** 128, 192 or 256 bits (independent of block size)
- 10 to 14 rounds (depends on key length)
- Each round uses 4 functions (in 3 “layers”)
 - ByteSub (nonlinear layer)
 - ShiftRow (linear mixing layer)
 - MixColumn (nonlinear layer)
 - AddRoundKey (key addition layer)



AES demonstration

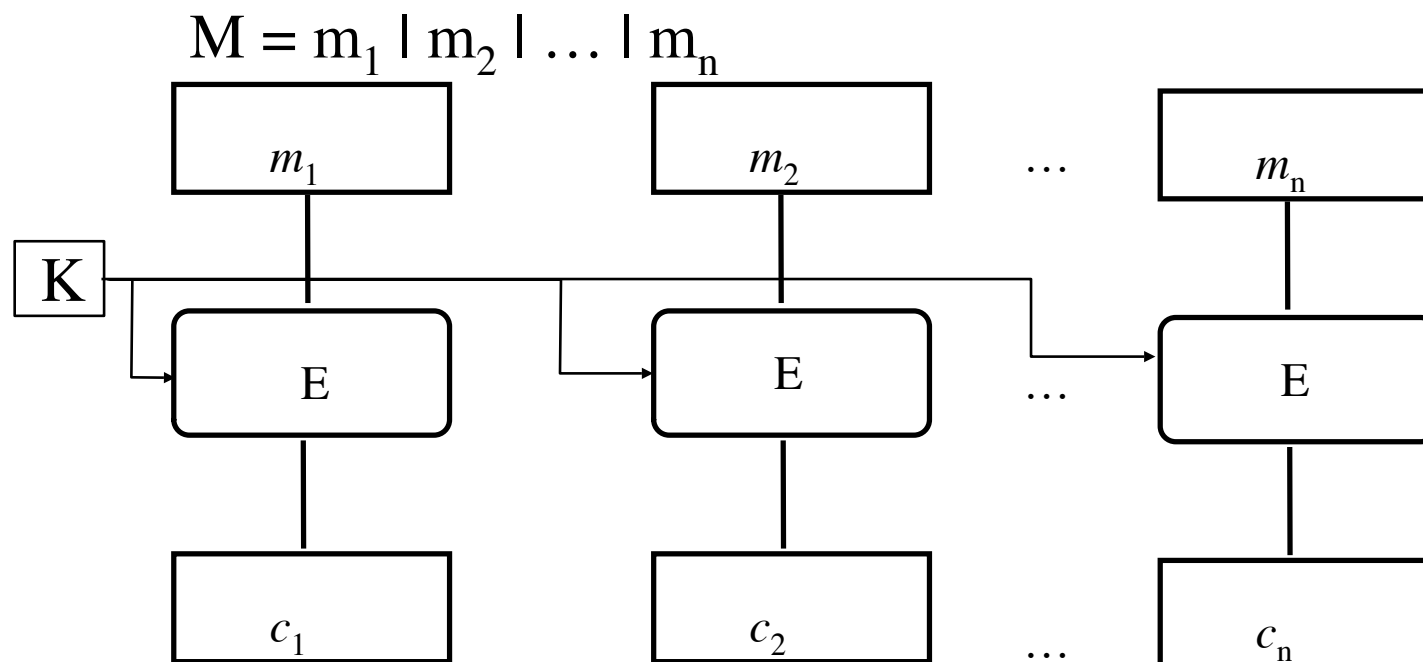




THE UNIVERSITY OF BRITISH COLUMBIA

Modes of Operation

Electronic Code Book (ECB)



$$C = c_1 | c_2 | \dots | c_n$$

$$c_i = E_K(m_i)$$

Drawbacks

- Same message has same ciphertext
- Redundant/repetitive patterns will show through
- Subject to "cut-and-splice" attacks

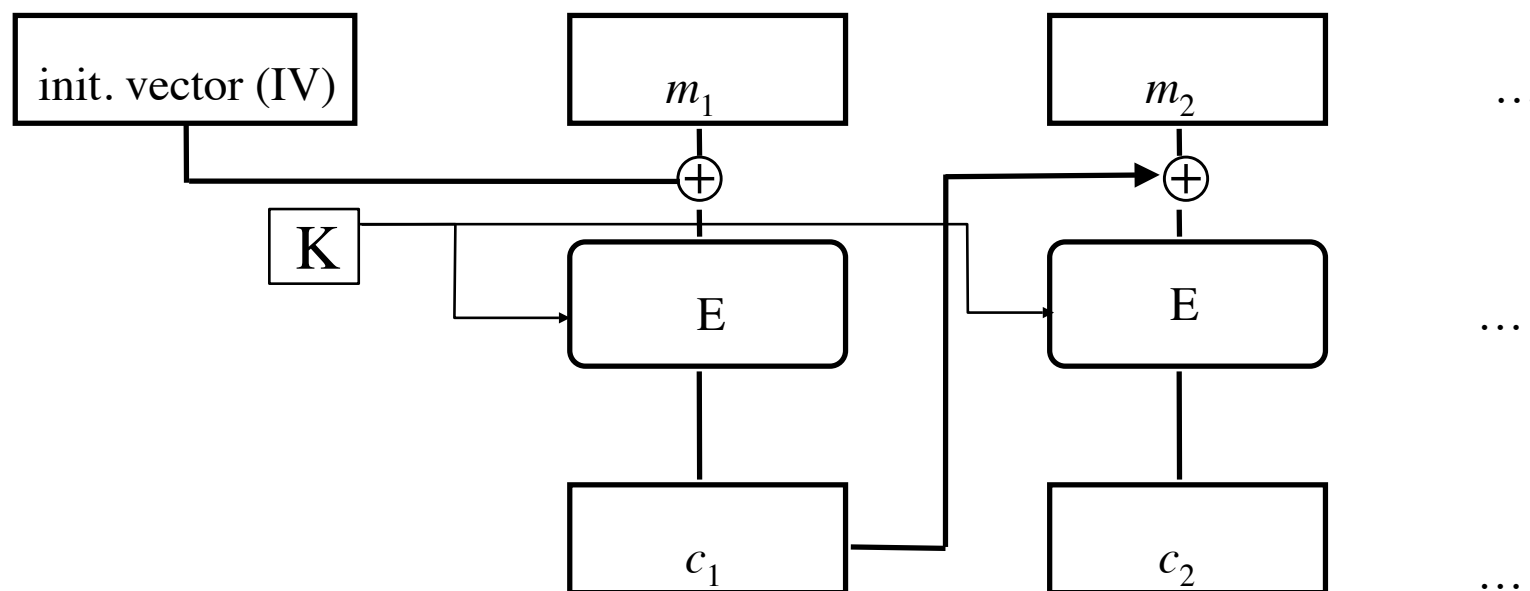
Alice in ECB Mode



Cipher Block Chaining (CBC)

$$c_i = E_K(m_i \oplus c_{i-1})$$

$$M = m_1 \mid m_2 \mid \dots \mid m_n$$

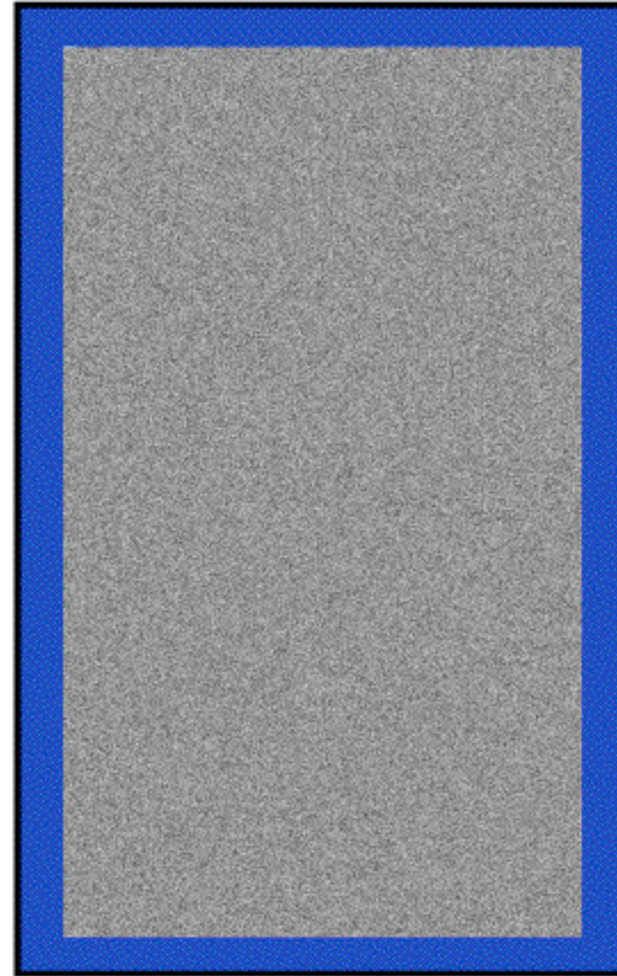


$$C = IV \mid c_1 \mid c_2 \mid \dots \mid c_n$$

Decrypting with CBC

$$c_i = D_K(m_i) \oplus c_{i-1}$$

Alice in CBC Mode



Output Feedback (OFB) Mode

- $C_i = m_i \oplus E_K(K_{i-1}), K_0 = IV$
- $K_1 = E_K(IV), K_2 = E_K(K_1), \dots K_i = E_K(K_{i-1}) \dots$
- Purpose
 - use block cipher as a stream cipher

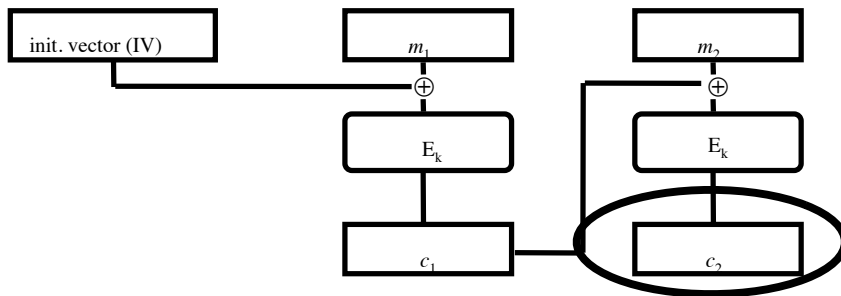


Counter Encryption

- Drawbacks of feedback modes
 - Hard to parallelize
 - CBC -- cannot pre-compute
 - OFB -- memory requirements
- Counter Encryption is easier to parallelize
 - $c_i = m_i \oplus E_K(IV+i)$
 - $m_i = c_i \oplus E_K(IV+i)$

Message Authentication Code (MAC)

- Purpose
 - protect message integrity and authenticity
- How to do MAC with a block cipher?

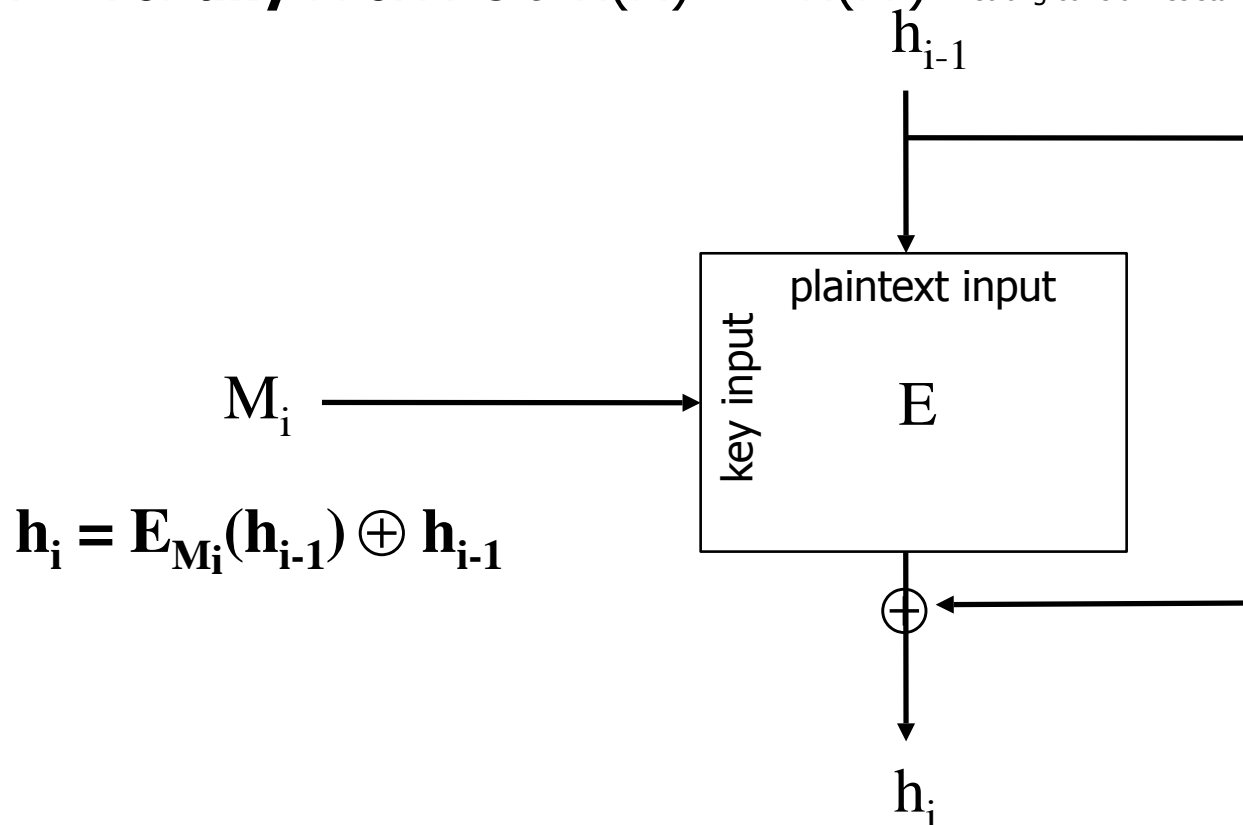


In CBC mode, the last block of cipher text serves as the MAC for the entire message

Hash Function from a Block Cipher

$$h = H(M)$$

1. Easy to compute h from M - efficient
2. Hard to compute M from h - one way
3. For **given** M , hard to find another M' s.t. $H(M) == H(M')$ - weak collision resistance
4. For **any** M & M' s.t. $H(M) == H(M')$ - strong collision resistance



Common Hash Functions and Applications

- Common hash functions
 - (Message Digest) MD5 value 128b
 - (Secure Hash Algorithm) SHA-1 180b value, SHA-256, SHA-512
- Applications
 - MACs
 - $MAC_K(M) = H(K, M)$
 - $HMAC_K(M) = H(K \oplus A, H(K \oplus B, M))$, A & B = magic (pg. 94, Stamp)
 - Time stamping service
 - key updating
 - $K_i = H(K_{i-1})$
 - Backward security
 - Autokeying
 - $K_{i+1} = H(K_i, M_{i1}, M_{i2}, \dots)$
 - Forward security



Key Points

- Ciphers are either substitution, transposition (a.k.a., permutation), or product
- Any block cipher should confuse and defuse
- Block ciphers are implemented in SP-networks
- Stream ciphers and hash functions are commonly implemented with block ciphers
- Hash functions used for
 - fingerprinting data, MAC, key updating, autokeying,