MONASH University
Information Technology

FIT2093 INTRODUCTION TO CYBER SECURITY

Week 2 Lecture

Cryptography I:

Symmetric Key Encryption

Part 2

Principles for CONFIDENTIALITY


Outline

Symmetric Cryptography

Part 2: Modern Encryption Algorithms

- Block ciphers
  - Design requirements
  - Case study I: Data Encryption Standard (DES)
  - Case study II: Advanced Encryption Standard (AES)
- Block Cipher Modes: How to use block ciphers for encryption
  - ECB, CBC, CTR

MONASH University

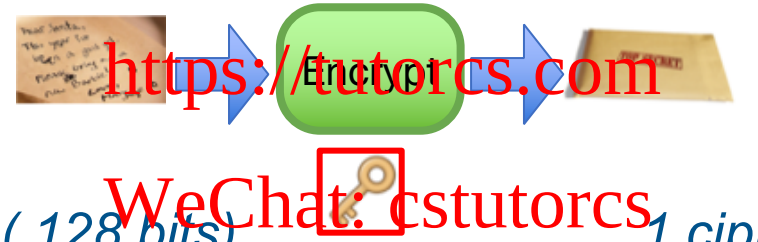
Block Ciphers

Modern Ciphers vs Classical Ciphers


Symmetric Cryptography

- vs **classical** ciphers: process plaintext in **characters**
- **modern** ciphers: process plaintext in **blocks**

Classical: 1 plaintext character e.g. A *1 ciphertext char e.g. X*



Modern: 1 plaintext block (128 bits) *1 ciphertext block (128 bits)*
e.g. 11010…111 *e.g. 01011…011*

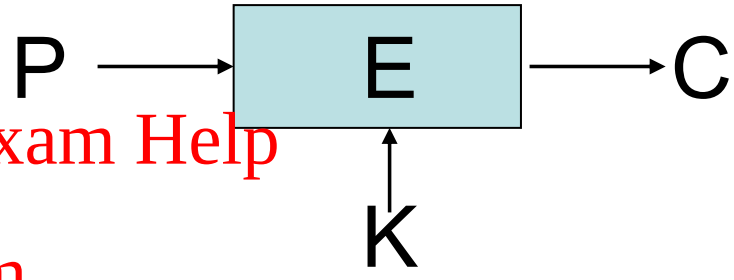
Modern approach to encryption design:

- Design a **block cipher** (for encrypting **one** block length, e.g. 128 bit)
- Use a block cipher **mode of operation** to encrypt (arbitrary length messages)
  - Another approach: stream ciphers – bit by bit encryption (*we won't study)*

Block Cipher

Symmetric Cryptography

- Inputs:
  - plaintext block P of length n bits
  - secret key K of length k bits
- Output:
  - ciphertext block C of length n bits



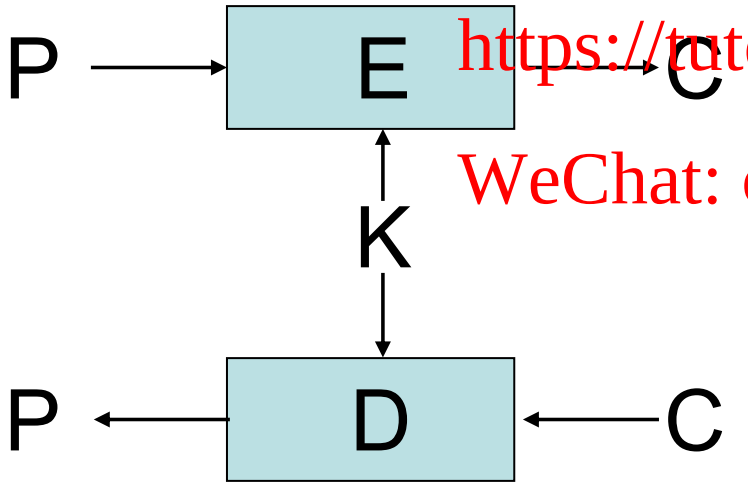
- algorithm E applies efficient substitution & permutation (transposition) operations to compute C = E(K,P)


Block Cipher: Requirements

Symmetric Cryptography

- Decryption Correctness:
  - Decryption algorithm D correctly decrypts C produced by E, i.e.

If:



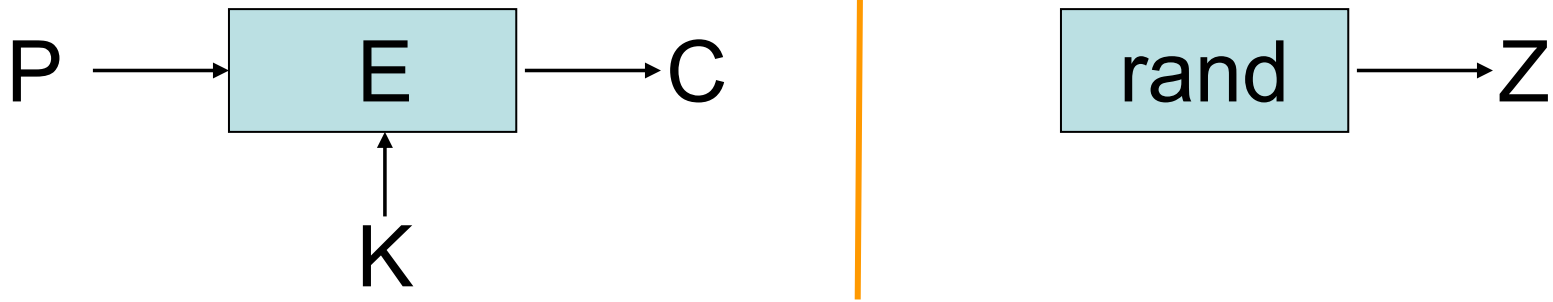
Then:

  - i.e. for any (P, K), if C = E(K,P) then D(K,C) = P

Block Cipher: Requirements

Symmetric Cryptography

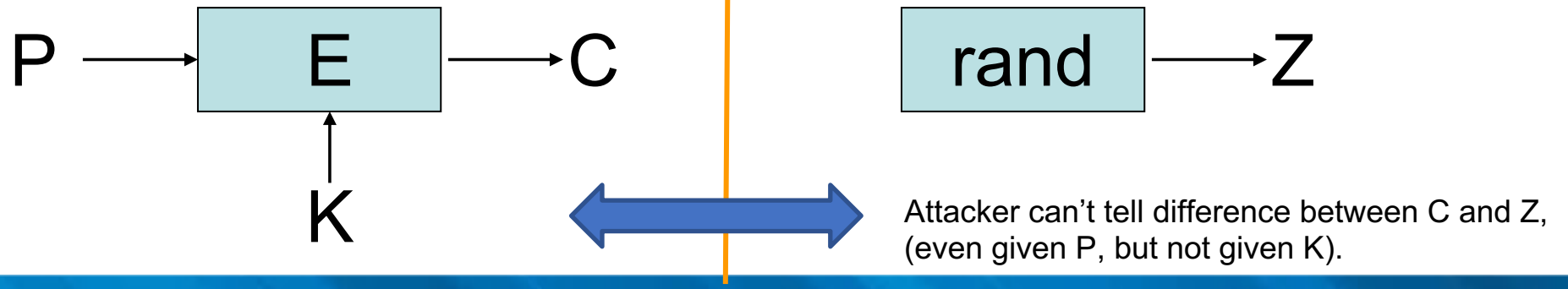

- Pseudorandom Function Security (PRF):
  - E should have the **Confusion & Diffusion** properties
  - Outputs of E look random: if K chosen randomly,
    - **computationally infeasible** for an attacker to **distinguish output ciphertexts** $C_i = E(K,P_i)$ **from random bit strings** of same length,
    - even if attacker corresponding distinct plaintexts P_i's (known plaintext attack)



Block Cipher: Requirements

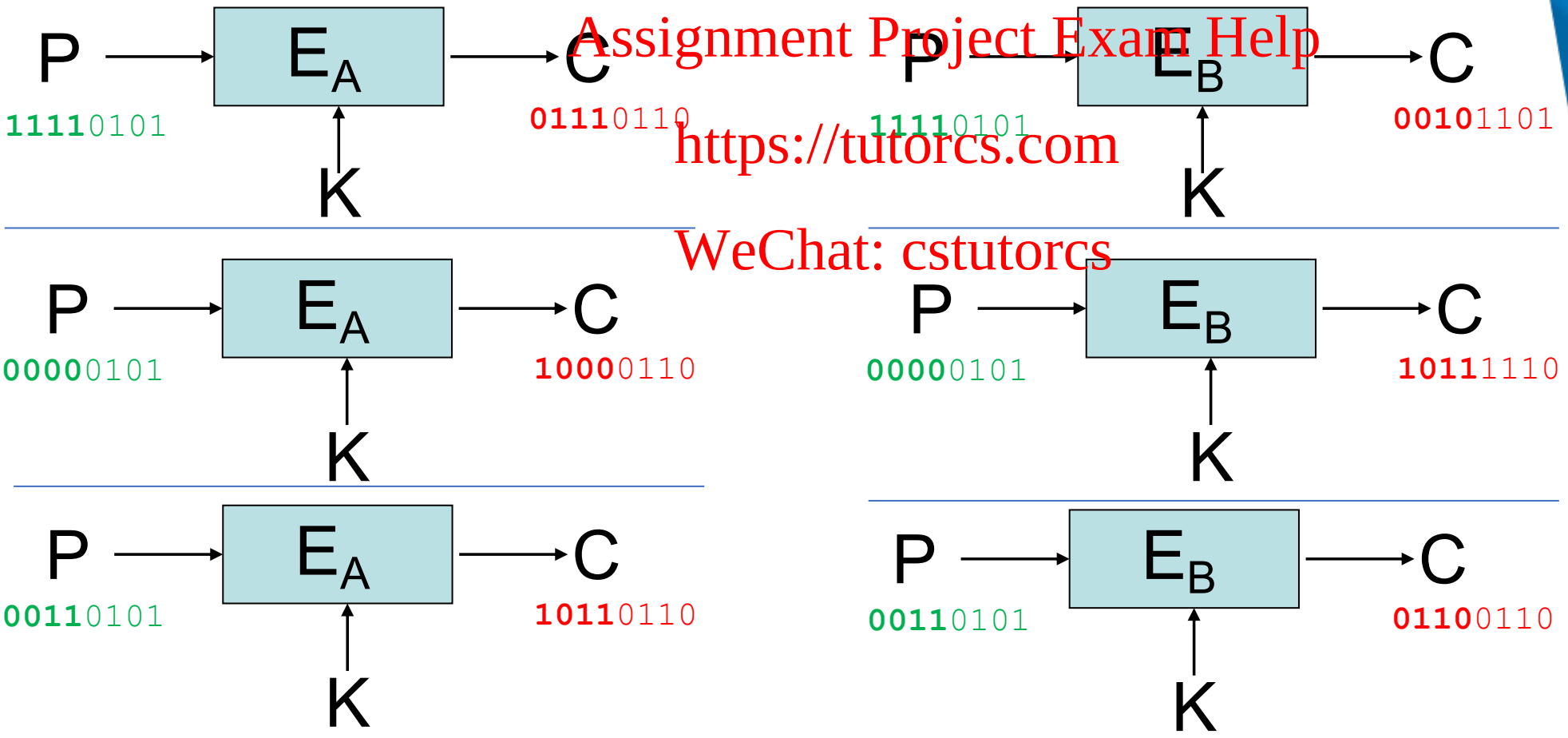
Symmetric Cryptography


- Pseudorandom Function Security (PRF):
  - i.e. as long as K is random,
    - outputs C from encryption E should look like random bit strings Z of same length
      - *attacker cannot tell the difference, even if s/he knows P*
    - → *means the encryption really able to "mix"* P *up*



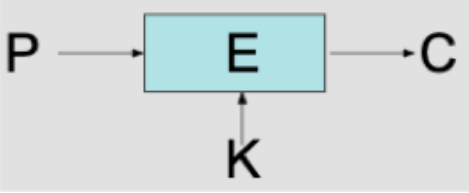
PRF Security: Simple Examples

- Q: Do you think Cipher E_A has PRF security?
- What about cipher E_B?



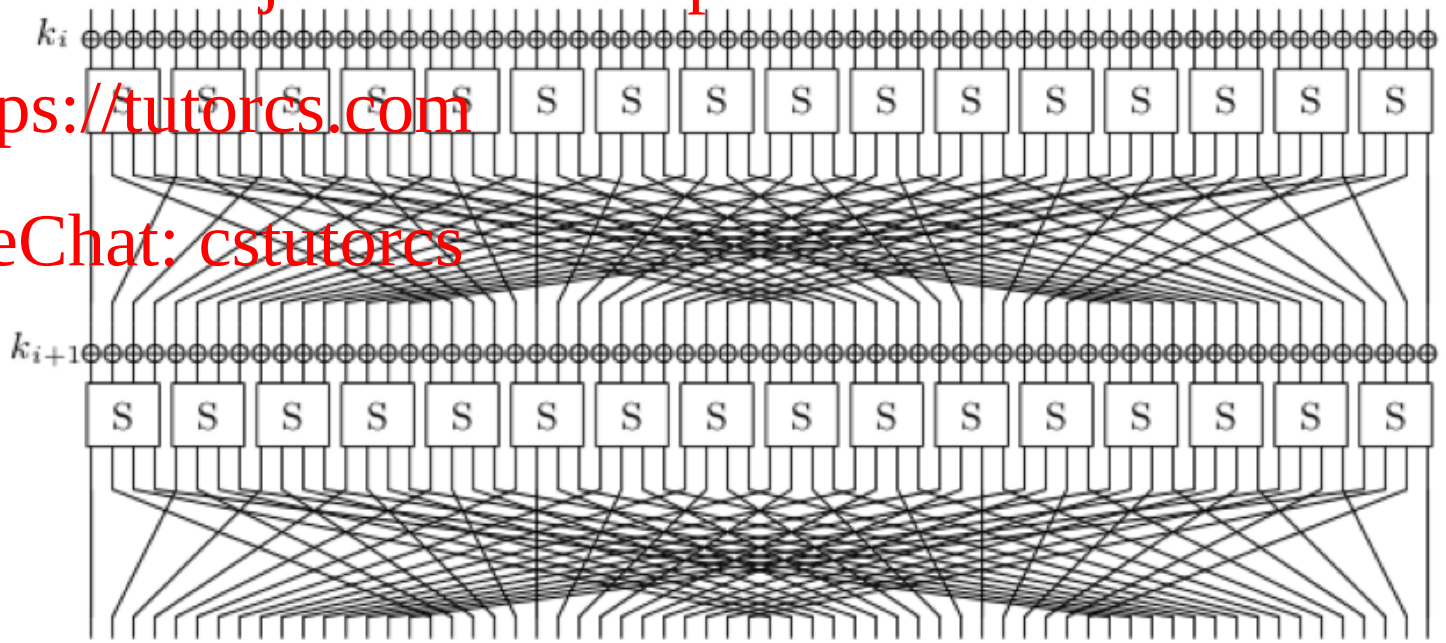
Block Cipher: Design

Symmetric Cryptography



- **Repeated** combination of …
  - Confusion via Substitution
    - substitutes
  - Diffusion via Permutation
    - spreads

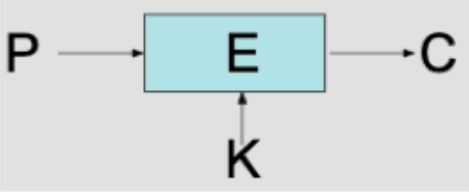
Q: why repeat?



Note: image by PRESENT designers

Block Cipher: Older ones

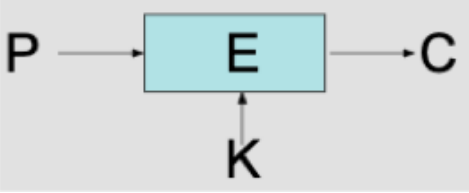
Symmetric Cryptography



- **DES [US, 1977] (*stronger version called* 3DES)**
  - |P|=|C|= 64 bits, |K| = 56 bits
- LOKI [ADFA, Australia, 1989]
  - |P|=|C|=|K| = 64 bits
- FEAL [NTT, Japan, 1990]
  - |P|=|C|= 64 bits, |K| = 128 bits
- IDEA [Lai & Massey, Switzerland, 1991], used in PGP
  - |P|=|C|= 64 bits, |K| = 128 bits
- MISTY1 [Mitsubishi, Japan, 1995]
  - |P|=|C|= 64 bits, |K| = 128 bits

Block Cipher: Newer ones

Symmetric Cryptography



- **KASUMI for 3G standard [Mitsubishi+3GPP, 2000]**
  - |P|=|C|= 64 bits, |K| = 128 bits
- **AES [Joan Daemen & Vincent Rijmen, Belgium, 2001]**
  - |P|=|C|= 128 bits, |K| = 128, 192, 256 bits

- **Lightweight** Block Ciphers
  - ISO/IEC 29192-2:2019:
  - PRESENT [2007]: |P|=|C|= 64 bits, |K| = 180, 128 bits
  - CLEFIA [2007]: |P|=|C|= 128 bits, |K| = 128,192,256
  - LEA [2013]: |P|=|C|= 128 bits, |K| = 128,192,256

Block Cipher: Data Encryption Standard (DES)

Symmetric Cryptography

- **DES** [US, 1977]
  - |P|=|C|= 64 bits, |K| = 56 bits
  - has Feistel structure [Horst.Feistel@IBM]
  - used to be *the* standard for encryption
  - obsolete, totally insecure, withdrawn as standard [2005]

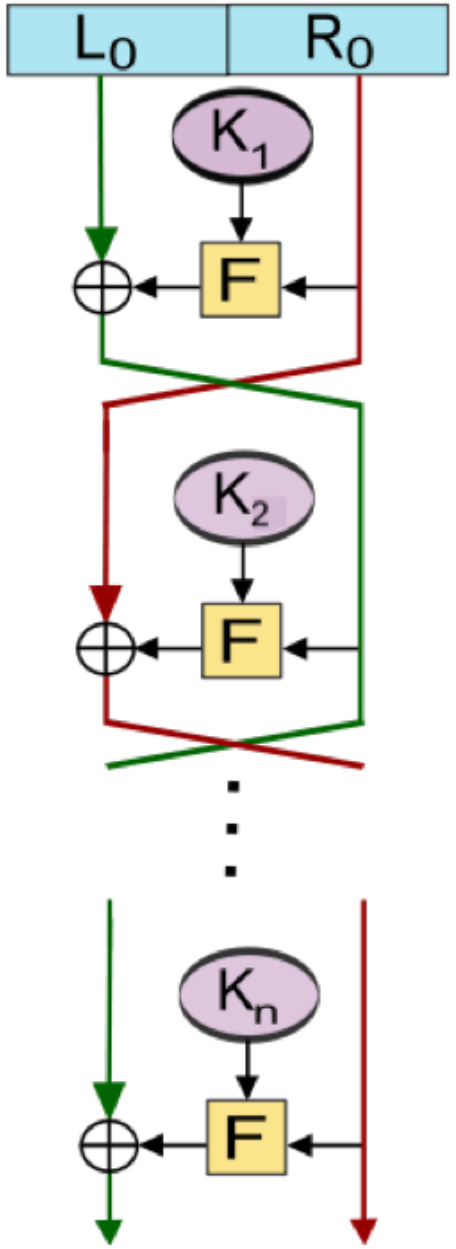
- **Triple-DES (3-DES)** [US, 1998]
  - stronger version of DES
  - do DES 3 times
  - still in use as a standard

DES Structure: Feistel

Symmetric Cryptography

- 1st round:
  - $L_1 = R_0$
  - $R_1 = L_0 \oplus f(R_0 , K_1)$
- for each round i (i = 1,2,...,n):
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1} , K_i)$

  where
  - f = round function
  - K_i = round sub-key
- Decryption: reverse the process



Bit-wise exclusive-or (XOR) : Refresher

Symmetric Cryptography

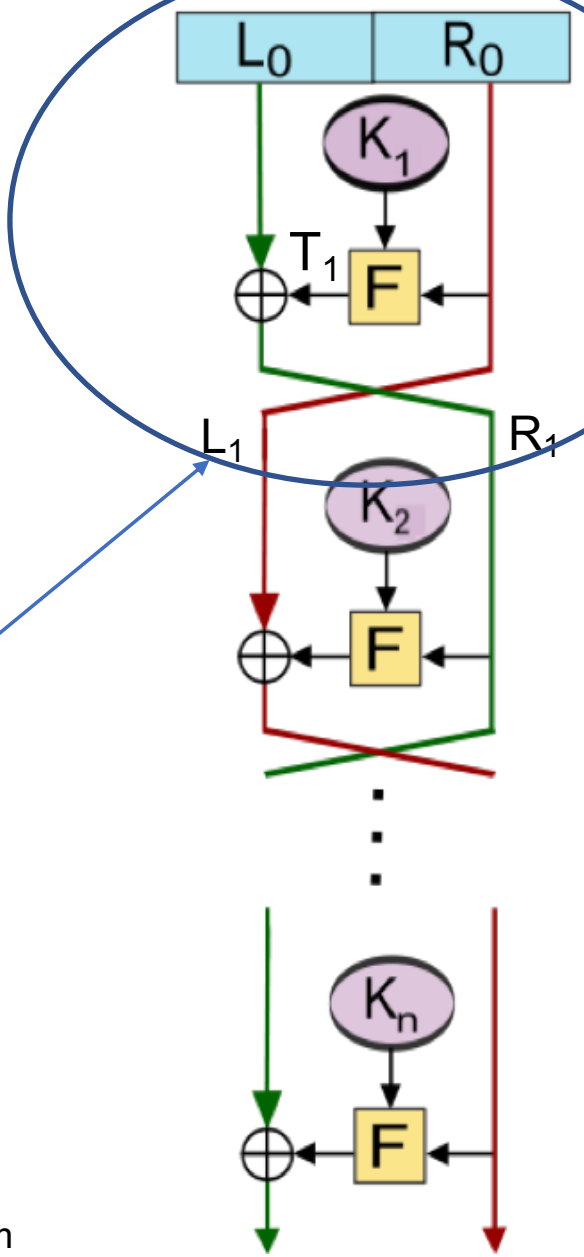
Decrytibility property of XOR: $(P \oplus K) \oplus K = P$

e.g. $P = 1110$, $K = 0101$,

Encrypt: $C = P \oplus K = 1110 \oplus 0101 = 1011$

Decrypt: $P = C \oplus K = 1011 \oplus 0101 = 1110$

0	⊕	0	=	0
1	⊕	1	=	0
0	⊕	1	=	1
1	⊕	0	=	1



Q: How to use decryptibility of XOR to decrypt the first round of Feistel cipher?

i.e. given (L_1, R_1), and K_1, how to compute (L_0, R_0)?

- 1) How can we find R_0 from L_1? (Hint: Look at the top of left diagram!)
- 2) How can we find T_1 from R_0 and K_1? (Hint: Look at the top of left diagram!)
- 3) How can we find L_0 from R_1 and T_1? (Hint: From diagram, $R_1 = L_0 \oplus T_1$)

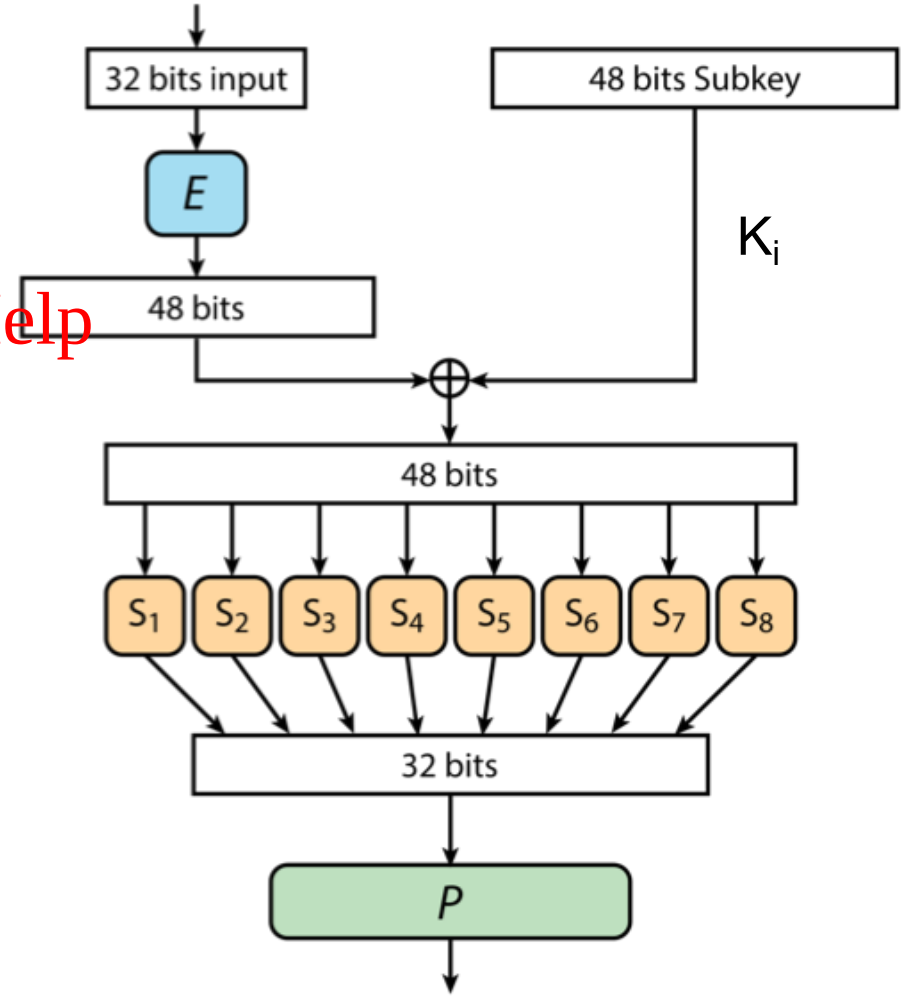
Activity (2 mins)

1) Add your question responses to the Ed forum

DES Structure: Round Function F

Symmetric Cryptography

- 32 bits go into f
- E: 32 ⟶ 48 bits, & permuted
- ⊕: XOR with subkey K_i
- 48 ⟶ 6 | 6 | 6 | 6 | 6 | 6 bits
- 8 S-boxes: 6 ⟶ 4 bits
- 4 | 4 | 4 | 4 | 4 | 4 ⟶ 32 bits
- P: Permutes the 32 bits of Sbox outputs



Block Cipher: Relation of Parameters to Security

Symmetric Cryptography

- Block size |P|=|C|: ↑ block size ⇒ ↑ security
- Key size |K|: ↑ key size ⇒ ↑ security
- Number of rounds r: ↑ rounds ⇒ ↑ confusion/diffusion ⇒ ↑ security

- **Q:** Why not making blocksize, key size and number of rounds **very** large, if this (usually) tends to increase security?

Activity (2 mins)

1) Add your question responses to the Ed forum

Block Cipher: Security vs Brute Force

Symmetric Cryptography

- brute force / exhaustive key search
- effort to guess secret key K & to verify guess
- brute force guess k-bit key needs 2^k guesses & verifications
- Q: how to verify key guess?

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\ \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years	6.4×10^{6} years

Block Cipher: Advanced Encryption Standard (AES)

Symmetric Cryptography

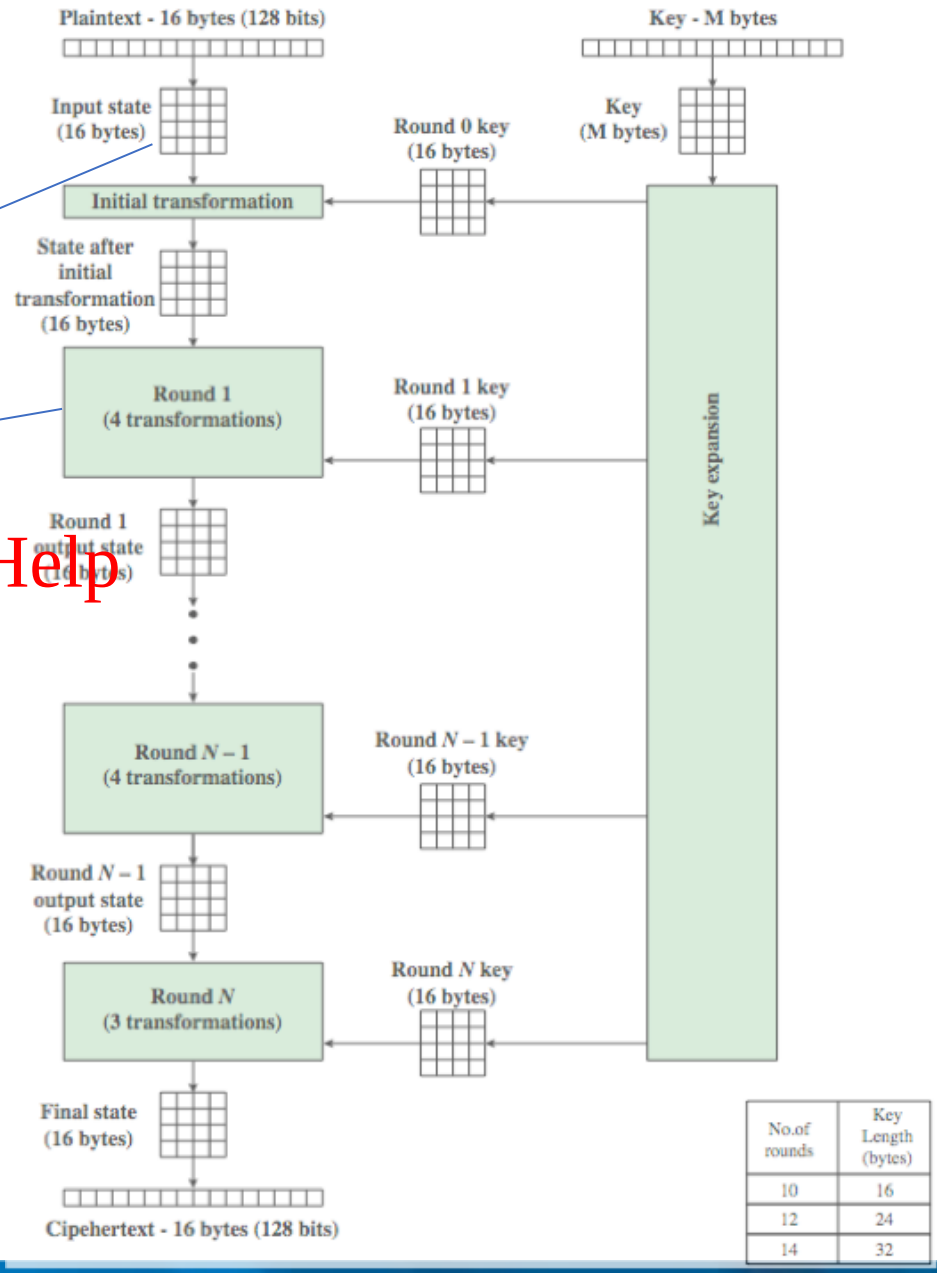
- [1997] US National Institute of Standards & Technology (NIST) issued call for proposals of algorithms for Advanced Encryption Standard (AES) to replace DES
- developed by Joan Daemen & Vincent Rijmen [Belgium]
  - also known as Rijndael algorithm
- [2000] AES algorithm selected, standardised in 2001

- Block size: |M| = |C| = 128
- Key size: |K| = 128, 192, 256 bits
- #rounds: 10, 12, 14

AES: Encryption Structure

Symmetric Cryptography

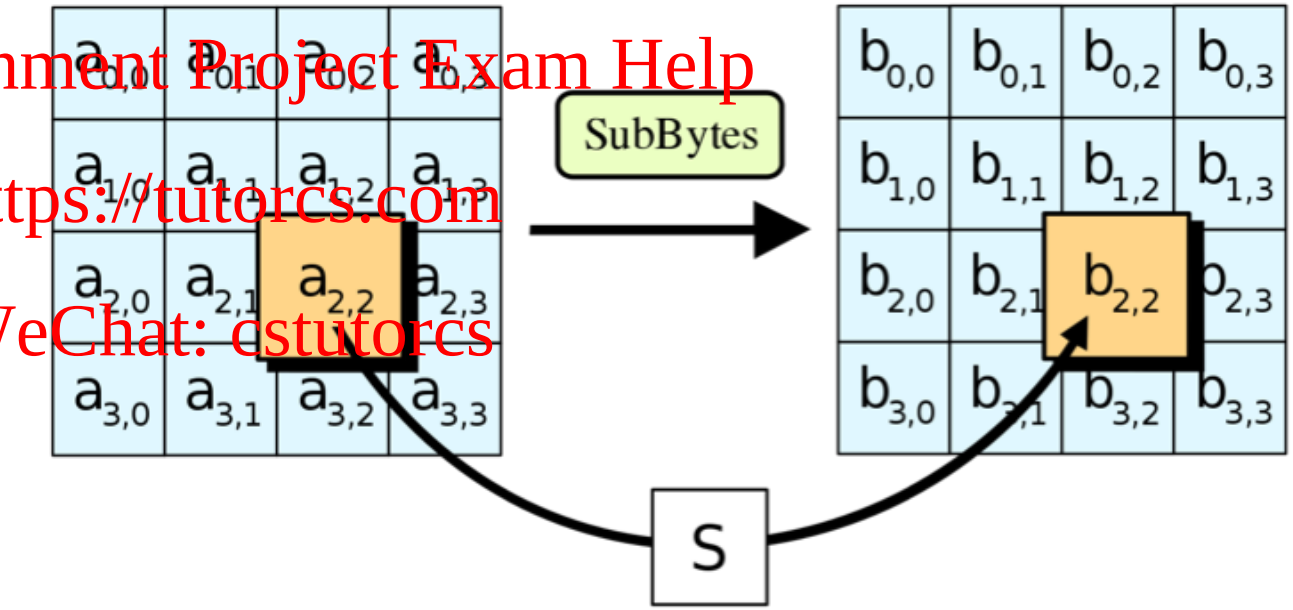
- Blocks
  - represented as array of bytes
- Round function
  - SubBytes:
  - ShiftRow
  - MixColumns
  - AddRoundKey



AES: Round Function

Symmetric Cryptography

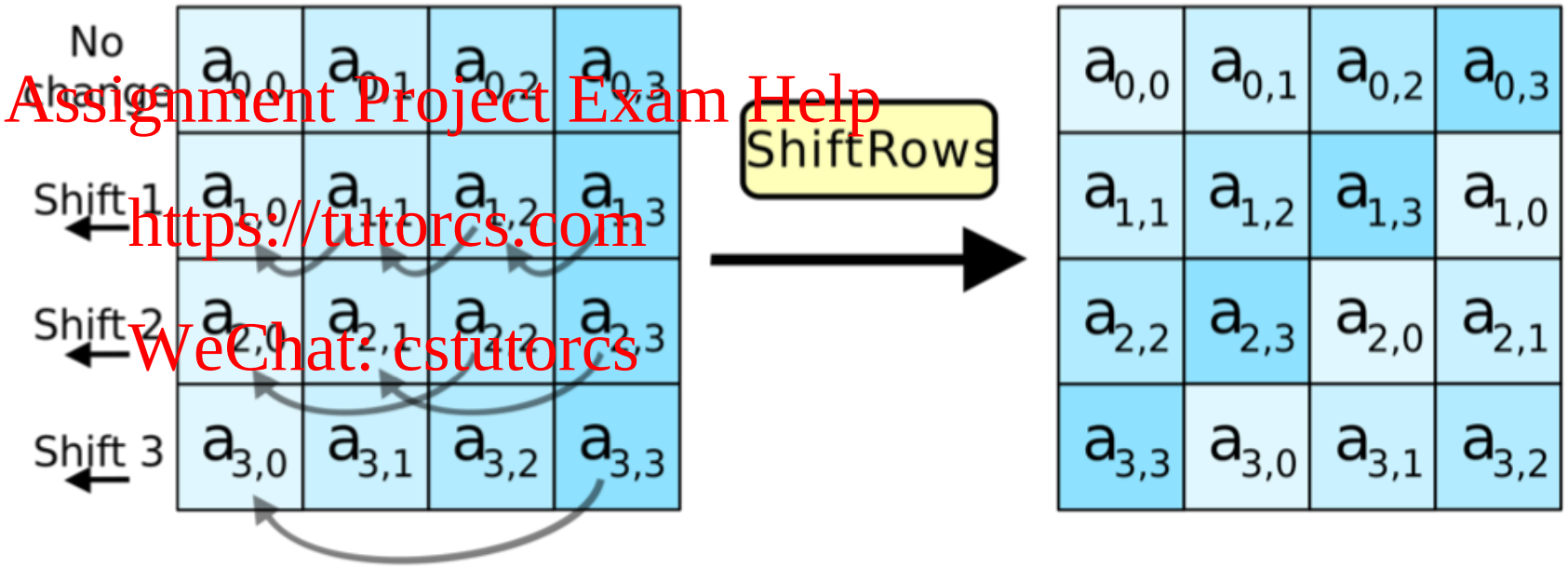
- Round function
  - **SubBytes**
  - ShiftRow
  - MixColumns
  - AddRoundKey



AES: Round Function

Symmetric Cryptography

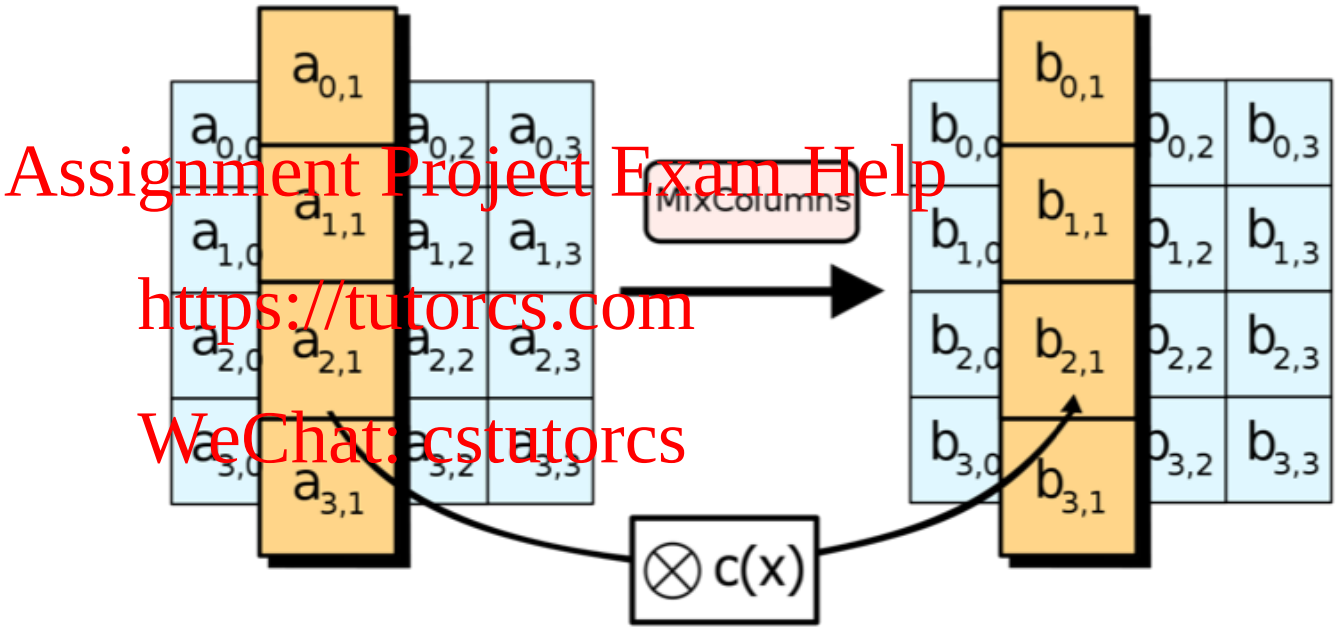
- Round function
  - SubBytes
  - **ShiftRow**
  - MixColumns
  - AddRoundKey



AES: Round Function

Symmetric Cryptography

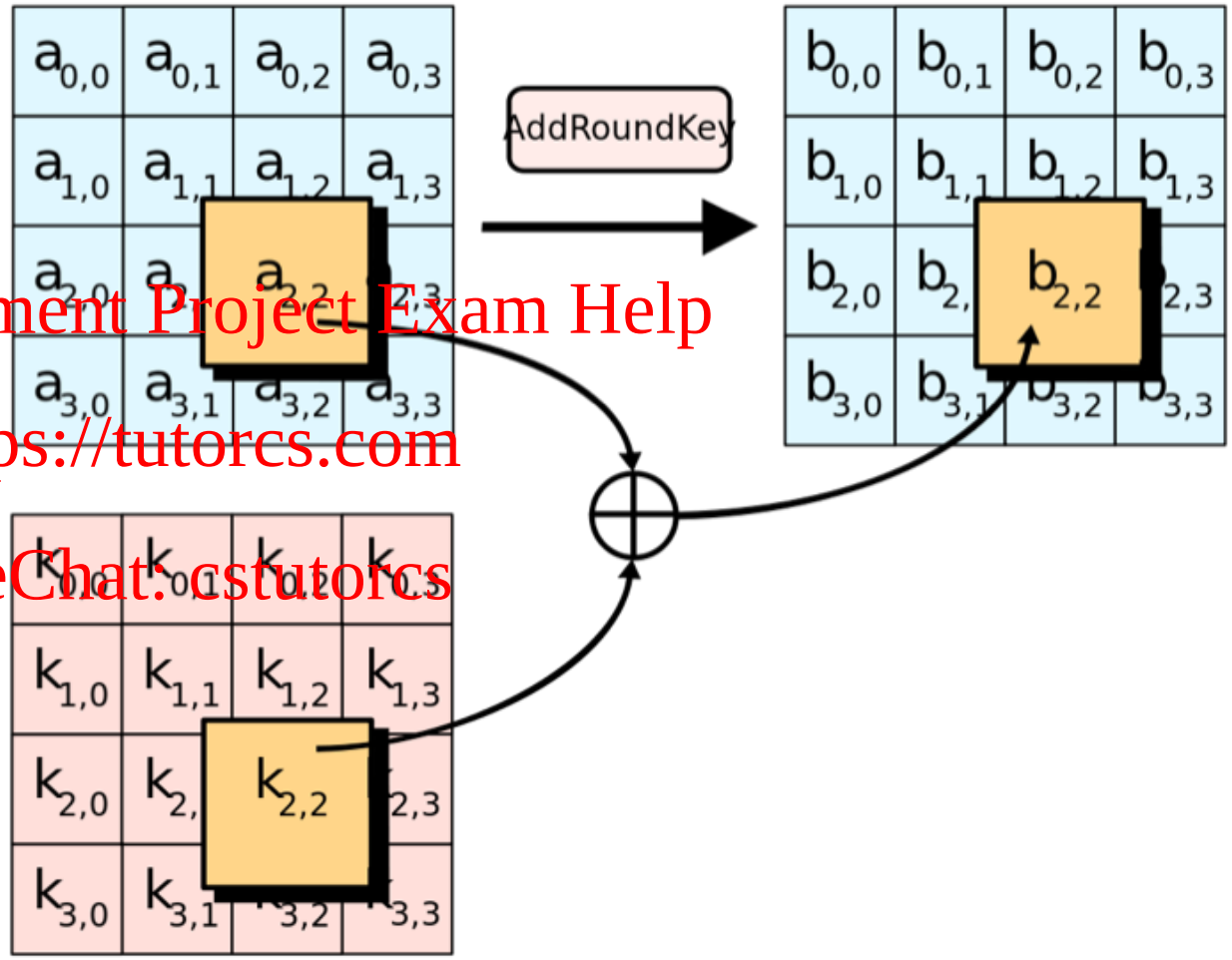
- Round function
  - SubBytes
  - ShiftRow
  - **MixColumns**
  - AddRoundKey



AES: Round Function

Symmetric Cryptography

- Round function
  - SubBytes
  - ShiftRow
  - MixColumns
  - **AddRoundKey**



AES: Implementation Aspects

Symmetric Cryptography

- efficiently implemented on 8-bit CPU
  - SubBytes: on bytes using a table of 256 entries
  - ShiftRows: simple byte shift
  - MixColumns: matrix multiplication in finite field GF(2^8) i.e. byte value coefficients
  - AddRoundKey: simple byte XOR
- Warning: Don't implement AES or DES yourself
  - straightforward implementations are insecure against side channel attacks (e.g. timing attack)
  - use a well-known crypto library written by experts

Cipher Security: Cryptanalysis

Symmetric Cryptography

- any cipher design needs to undergo security analysis
  - brute force: always applicable, upper bound
    - on average, need to try half of key space
    - e.g. for k-bit key, key space is 2^k
  - Goal of cryptanalysis: Find Shortcut attacks faster than brute force
    - exploit
      - cipher structure
      - plaintext characteristics/features/patterns
    - To try to make breaking effort < brute force
  - For a **well designed modern cipher**, **no shortcut attacks known**!
    - → Fastest attack = brute force

Cipher Security: Security Models

Symmetric Cryptography

- Q: To analyse threats against ciphers, we can specify an **attack model**
- Attack Model = Security **game** between:
  - **users** on one side, with security goal
    - > behave, follow steps
  - **attackers** on the other side, vs security goal
    - > have adversarial **goal**
    - > have adversarial **capabilities**
      - – **knowledge** of ...
      - – **access** to ...
      - – can **exploit** users

Cipher Security Models: Attacker Capabilities

Symmetric Cryptography

- Can categorize security model based on attacker's **capabilities** (what s/he can access to):
  - Ciphertext Only Attack (COA)
  - Known Plaintext Attack (KPA)
  - Chosen Plaintext Attack (CPA)
  - Chosen Ciphertext Attack (CCA)
    - can **exploit** users

Cipher Security Models: Attacker Capabilities

Symmetric Cryptography

- **Passive attacks**
  - **Ciphertext Only Attacks (COA)**
    - Attacker only has access to some samples of ciphertexts C
    - Example scenario: eavesdropping ciphertext of a password on the internet
  - **Known Plaintext Attacks (KPA)**
    - Attacker knows some samples of ciphertexts C with their corresponding plaintexts P, and try to decrypt another ciphertext C'
    - E.g. knowing C,P may reveal info. on key K, could then decrypt C'
    - examples:
      - Q: What could be a possible practical scenario for a known plaintext attack (KPA)?

Activity (2 mins)

1) Add your question responses to the Ed forum

Cipher Security Models: Attacker Capabilities

Symmetric Cryptography

- **Active attacks**

- **Chosen Plaintext Attacks (CPA)**
  - Attacker can feed encryption alg. with plaintexts P and obtain matching ciphertexts C
  - example: get sender to encrypt forwarded email received from attacker
    - Such chosen (P,C) pairs may reveal even more information on secret key K

- **Chosen Ciphertext Attacks (CCA)**
  - Attacker can feed decryption alg. with ciphertexts C and obtain info. on their decryption P
  - example:
    - attacker sends ciphertext C to web server
    - web server decrypts ciphertext C with key K, checks decryption validity
    - decryption validity or failure error may reveal info. on server's key K

Cipher Security Models: Attacker Goals

Symmetric Cryptography

- Plaintext Recovery (PR)
  - Attack goal: find the unknown **plaintext** P*
- Key Recovery (KR)
  - Attack goal: find the unknown secret **key** K
- INDistinguishability (IND)
  - *Attacker knows the plaintext list*
  - Attack goal: **distinguish** which plaintext $P^* \in \{P_0, P_1\}$ was encrypted to ciphertext C
    - Ideal security: **not even one bit of info**. on plaintext revealed to attacker! e.g is C = E(K,"Yes") or E(K,"No")?

PR

KR

IND

MONASH University

Block Cipher Modes

Block Cipher: Modes of Operation

Symmetric Cryptography

- block ciphers E **process** data in **blocks**
  - e.g. 64-bits (DES, 3DES) or 128-bits (AES)

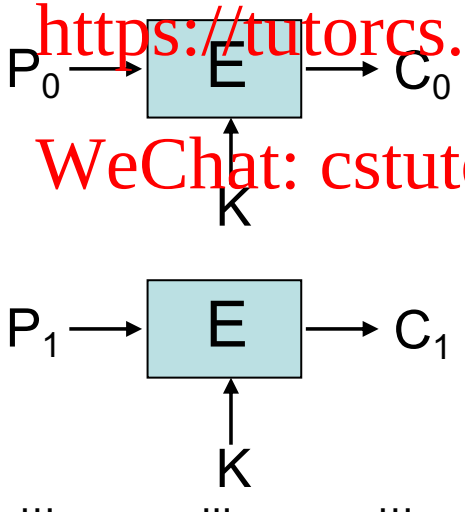
- for **longer messages**, must break up
  - & possibly **pad** the end to multiple of block size
  - unambiguous padding: 100000…0000000
- 5 **modes** (ways) of operation, to use a block cipher
  - defined in NIST SP 800-38A
  - ECB (insecure for most applications), CBC, CTR CFB, OFB (we don't study)

ECB: Electronic Code Book mode

Symmetric Cryptography

- message broken into **blocks**, & encrypted
- each block's value is **substituted**, like using a codebook
- each block is **encoded independently** of other blocks

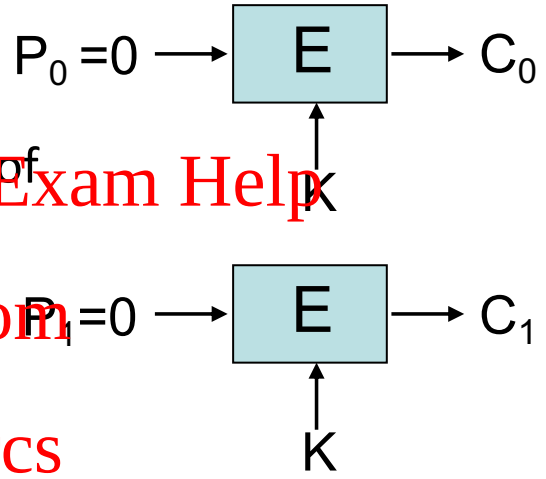
$C_i = E(P_i, K)$



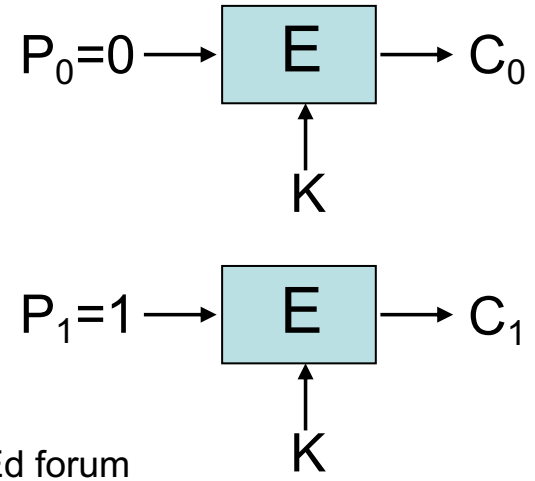
ECB Mode: Insecurity

ECB mode is insecure (i.e. not satisfy IND-CPA) even if block cipher is secure

- Statistical frequency analysis attack, again!
- e.g. easy to distinguish between ECB encryptions of
  - $M_0 = (P_0 = 0, P_1 = 0)$



  - $M_1 = (P_0 = 0, P_1 = 1)$



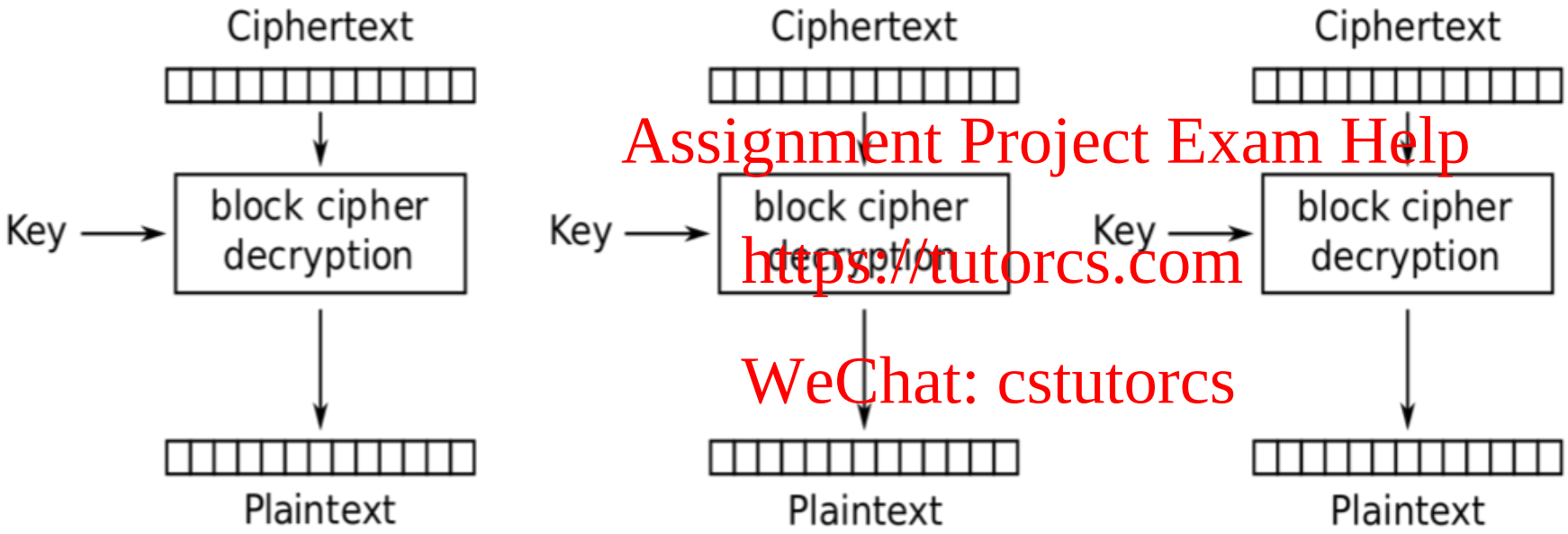
Q: How can an attacker easily distinguish whether given C_0, C_1 is ECB ciphertext of $M_0 = (0,0)$ or $M_1 = (0,1)$?

Activity (2 mins)

1) Add your question responses to the Ed forum

ECB: Electronic Code Book mode

Symmetric Cryptography



Electronic Codebook (ECB) mode decryption

ECB: Limitations

Symmetric Cryptography

- **block independence**
  - message **repetitions** may show in ciphertext
    - particularly with block data such as graphics
    - or with messages that change very little
  - can do code-book analysis
- should be **avoided** (*use a secure mode instead*)

ECB: Limitations

Symmetric Cryptography

plaintext

ECB

Secure mode

- **block independence:**
  - Message block **repetitions** may show **in ciphertext**
    - encryption does not destroy the message pattern

CBC: Cipher Block Chaining

Symmetric Cryptography

- message blocks **linked** together in encryption operation
- each **previous** cipher block is **chained** with current plaintext block, hence name

- use Initial Vector (IV) to start process

$$C_0 = IV$$

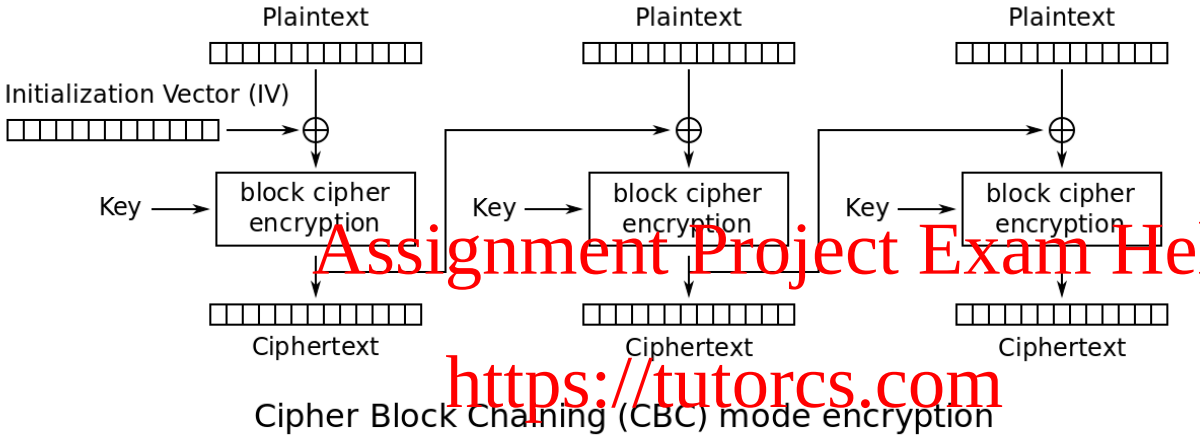
$$\text{for } i=1,\ldots,N$$

$$C_i = E(P_i \oplus C_{i-1}, K)$$

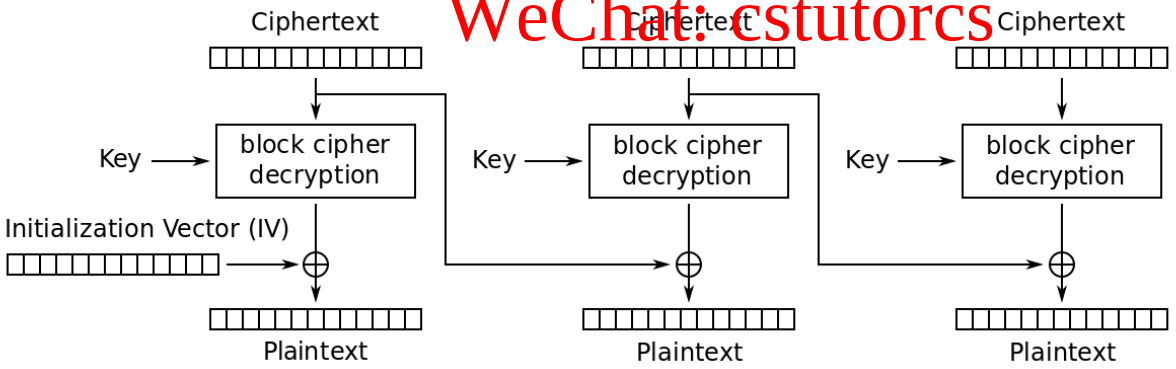
- uses: to secure **bulk data** e.g. hard disk encryption

CBC: Cipher Block Chaining

Symmetric Cryptography



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Image source: Wikipedia

CBC: Advantages & Disadvantages

Symmetric Cryptography

- ciphertext block depends on **all blocks before** it
  - Encryption **cannot** be **parallelize**d
- change to a block affects all following ciphertext blocks in encryption
- error propagation: error in received block affects two decrypted plaintext blocks
- **need** Initialization Vector **(IV)**
  - which must be **known** to sender & receiver
  - must be chosen **independently at random** for each encryption

- IND-CPA Security: if **IV random**ly chosen independently for each message, & block **cipher** is a secure **PRF**, then CBC mode is secure

Stream Modes of Operation

Symmetric Cryptography

- vs block modes: encrypt **entire block**
- may need to operate on **smaller units**
  - **real time** data
- convert **block cipher** into **stream cipher**
  - can use block cipher as some form of pseudo-random number generator (PRNG)
- Stream Modes:
  - cipher feedback (CFB) mode – we don't study
  - output feedback (OFB) mode – we don't study
  - counter (CTR) mode

CTR: Counter

Symmetric Cryptography

- **encrypts counter** value incremented after each block

- must have a different key & different counter value for every plaintext block (never reused)

$ctr_1 = IV$

for i=1,…,N

$O_i = E(ctr_i, K)$

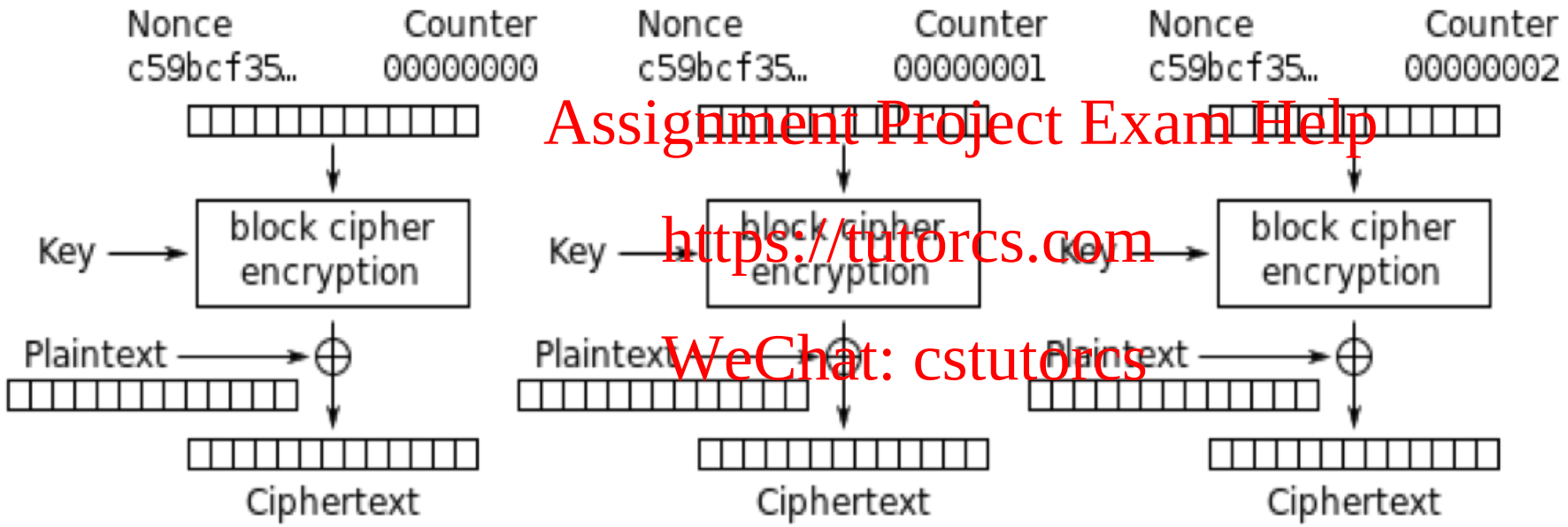
$C_i = P_i \text{ XOR } O_i$

$ctr_{i+1} = ctr_i + 1$ (increment ctr)

- uses: high-speed network encryptions, IPSec, network security

CTR: Counter

Symmetric Cryptography



Counter (CTR) mode encryption

CTR: Advantages & Limitations

Symmetric Cryptography

- efficiency
  - can do **parallel** encryptions in h/w or s/w
  - can **preprocess in advance**
  - good for **bursty** & **high speed** links
- No error propagation: error in received block does not affect decryption of subsequent blocks
- random access (independent blocks) to encrypted data blocks

- IND-CPA Security: if **IV randomly** chosen independently for each message or a nonce that's never reused, & block **cipher** is a secure **PRF**, then CTR mode is secure

Further Reading

- Chapter 20 of the textbook: *Computer Security: Principles and Practice" by William Stallings & Lawrie Brown,* Prentice Hall, 2015
- Chapters 1-3 of textbook: *Introduction to Modern Cryptography*, by Jonathan Katz and Yehuda Lindell, Chapman & Hall/CRC, 2008. [deeper understanding]
- [ZoomECB] Zoom Videoconference using ECB mode encryption: https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/