Kingdom of Saudi Arabia
Ministry of Education
Umm Al-Qura University
Adham University College
Computer Science Department

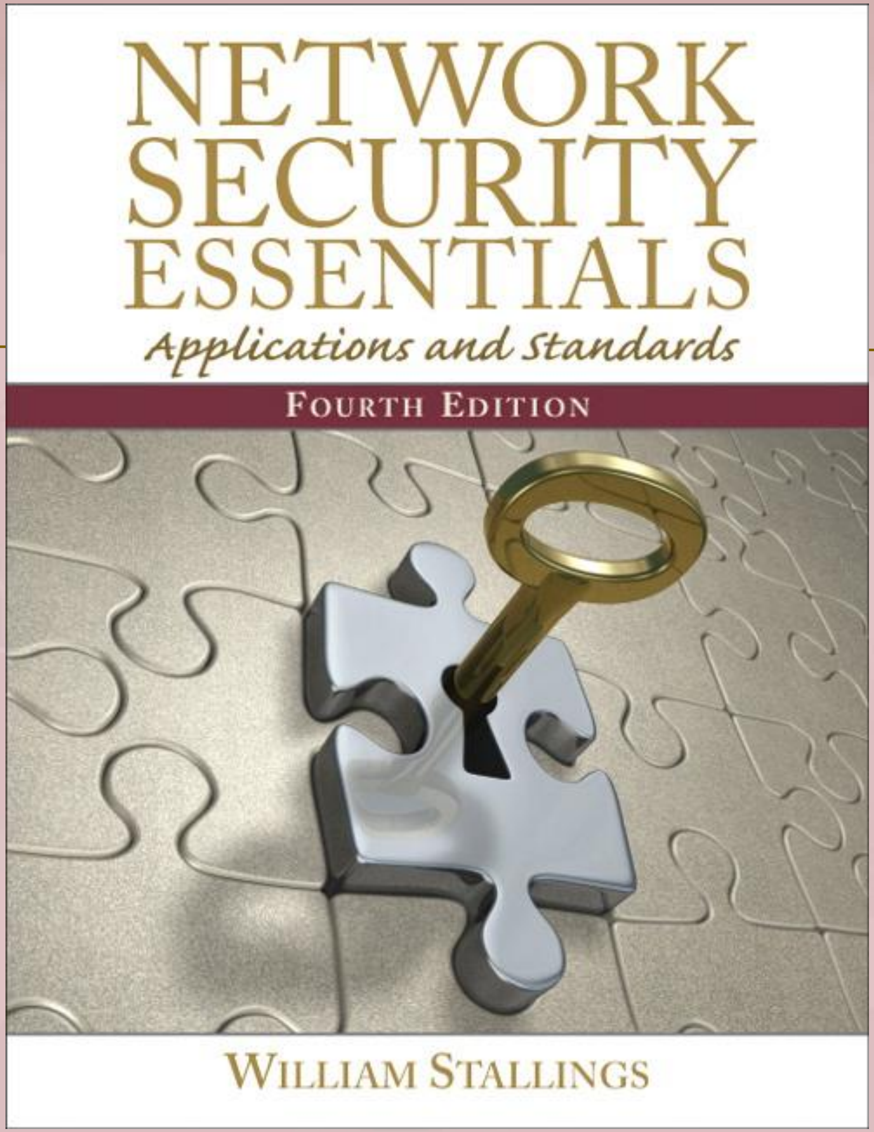
المملكة العربية السعودية
وزارة التعليم
جامعة أم القرى
الكلية الجامعية بأضم
قسم الحاسب الآلي

COMPUTER SECURITY SYSTEMS 6803532-3

T.Mariah Khayat

Main Reference:

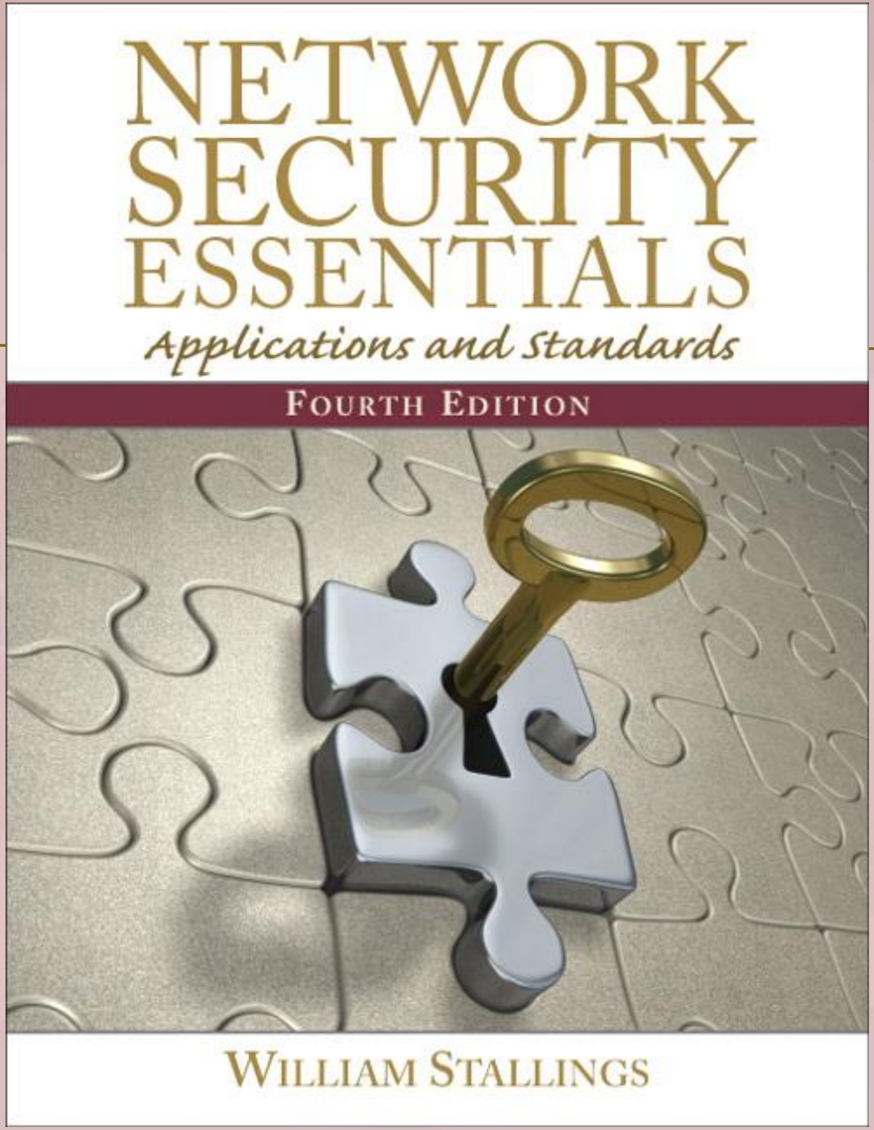
- ***Network Security Essentials, Fourth Edition, William Stallings.***



The Content of Slides are the Summary from the Main Reference by Lawrie Brown with Some Edits by Me. ;)

Chapter Two:

Symmetric Encryption and Message Confidentiality



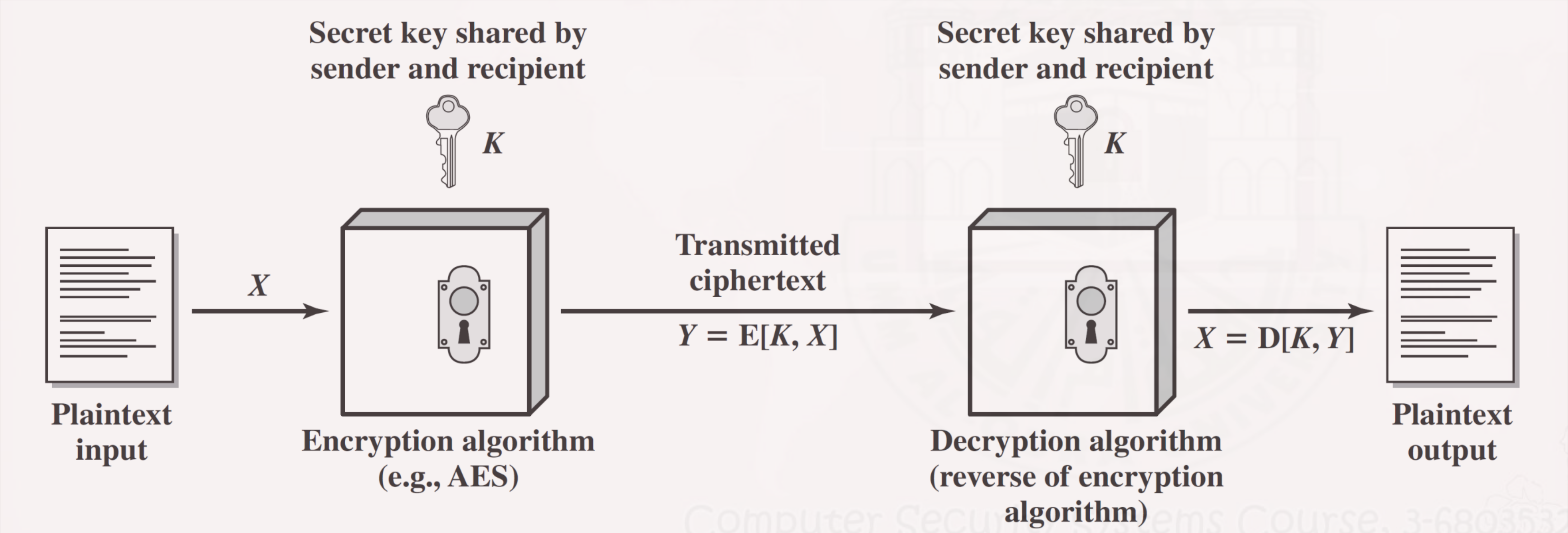
Symmetric Encryption

- or conventional / private-key / single-key
- sender and recipient share a common key
- all classical encryption algorithms are private-key
- was only type prior to invention of public-key in 1970's
- and by far most widely used

Some Basic Terminology

- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

Symmetric Cipher Model



Requirements

- two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- mathematically have:

  $Y = E(K, X)$

  $X = D(K, Y)$
- assume encryption algorithm is known
- implies a secure channel to distribute key

cryptography

❖ can characterize cryptographic system by:

- **type of encryption operations used**
  - ✓ substitution
  - ✓ transposition
  - ✓ product
- **number of keys used**
  - ✓ single-key or private
  - ✓ two-key or public
- **way in which plaintext is processed**
  - ✓ block
  - ✓ stream

cryptanalysis

- **objective to recover key not just message**
- **general approaches:**
  - cryptanalytic attack
  - brute-force attack
- **if either succeed all key use compromised**

Cryptanalytic Attacks

- **ciphertext only**
  - only know algorithm & ciphertext, is statistical, know or can identify plaintext
- **known plaintext**
  - know/suspect plaintext & ciphertext
- **chosen plaintext**
  - select plaintext and obtain ciphertext
- **chosen ciphertext**
  - select ciphertext and obtain plaintext
- **chosen text**
  - select plaintext or ciphertext to en/decrypt

More Definitions

- **unconditional security**
  - no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- **computational security**
  - given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Brute Force Search

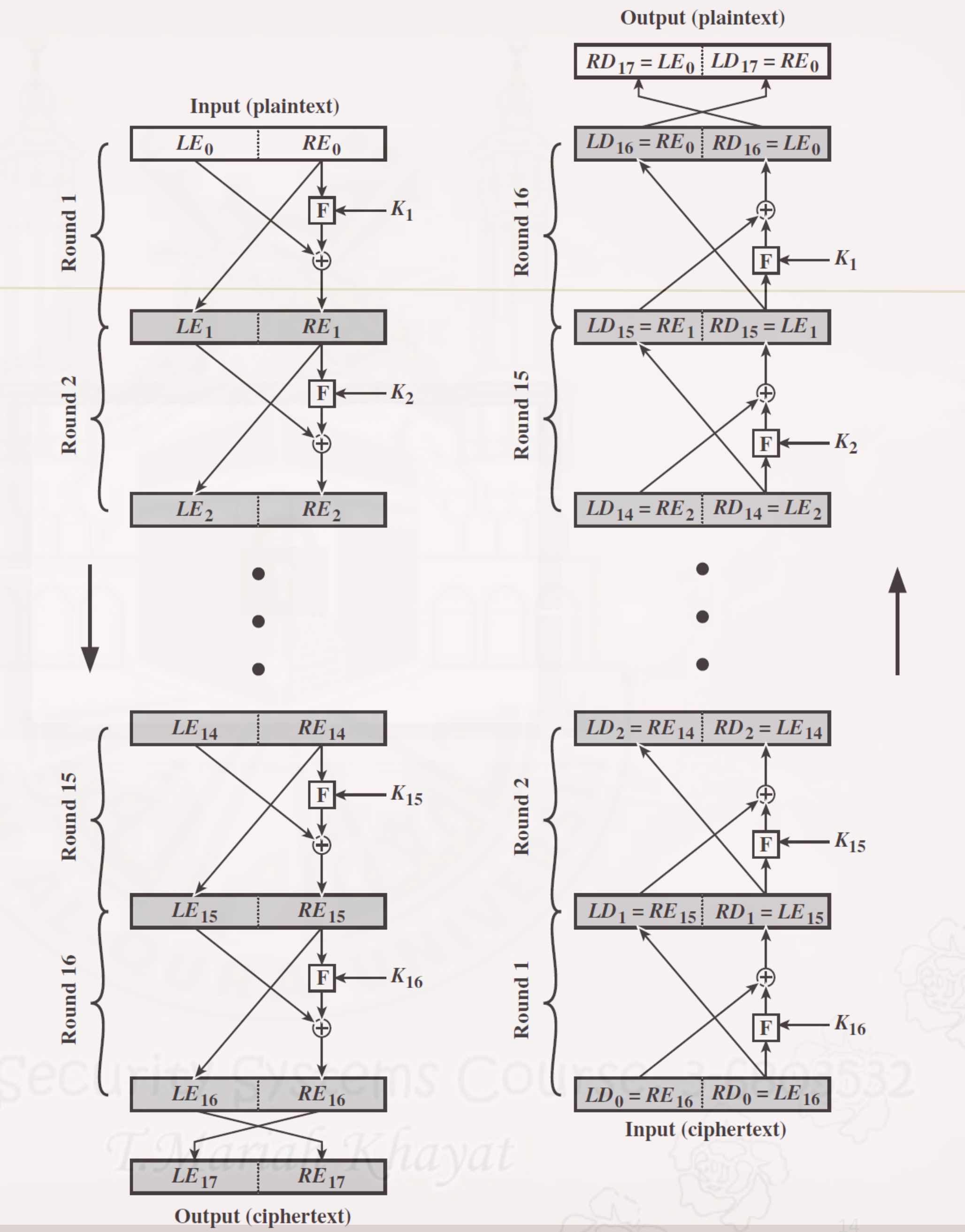
- always possible to simply try every key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

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

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept

Feistel Cipher Structure



Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis

Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security

DES History

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES

DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- use of DES has flourished
  - especially in financial applications
  - still standardised for legacy application use

Multiple Encryption & DES

- clear a replacement for DES was needed
  - theoretical attacks that can break it
  - demonstrated exhaustive key search attacks
- AES is a new cipher alternative
- prior to this alternative was to use multiple encryption with DES implementations
- Triple-DES is the chosen form

Double-DES?

- could use 2 DES encrypts on each block
  - $C = E_{K2}(E_{K1}(P))$
- issue of reduction to single stage
- and have "meet-in-the-middle" attack
  - works whenever use a cipher twice
  - since $X = E_{K1}(P) = D_{K2}(C)$
  - attack by encrypting P with all keys and store
  - then decrypt C with keys and match X value
  - can show takes $O(2^{56})$ steps

Triple-DES with Two-Keys

- **hence must use 3 encryptions**
  - would seem to need 3 distinct keys
- **but can use 2 keys with E-D-E sequence**
  - $C = E_{K1}(D_{K2}(E_{K1}(P)))$
  - nb encrypt & decrypt equivalent in security
  - if K1=K2 then can work with single DES
- **standardized in ANSI X9.17 & ISO8732**
- **no current known practical attacks**
  - several proposed impractical attacks might become basis of future attacks

Triple-DES with Three-Keys

- although are no practical attacks on two-key Triple-DES have some indications
- can use Triple-DES with Three-Keys to avoid even these
  - $C = E_{K3}(D_{K2}(E_{K1}(P)))$
- has been adopted by some Internet applications, eg PGP, S/MIME

Origins

- ❖ **clear a replacement for DES was needed**
  - have theoretical attacks that can break it
  - have demonstrated exhaustive key search attacks
- ❖ **can use Triple-DES – but slow, has small blocks**
- ❖ **US NIST issued call for ciphers in 1997**
- ❖ **15 candidates accepted in Jun 98**
- ❖ **5 were shortlisted in Aug-99**
- ❖ **Rijndael was selected as the AES in Oct-2000**
- ❖ **issued as FIPS PUB 197 standard in Nov-2001**

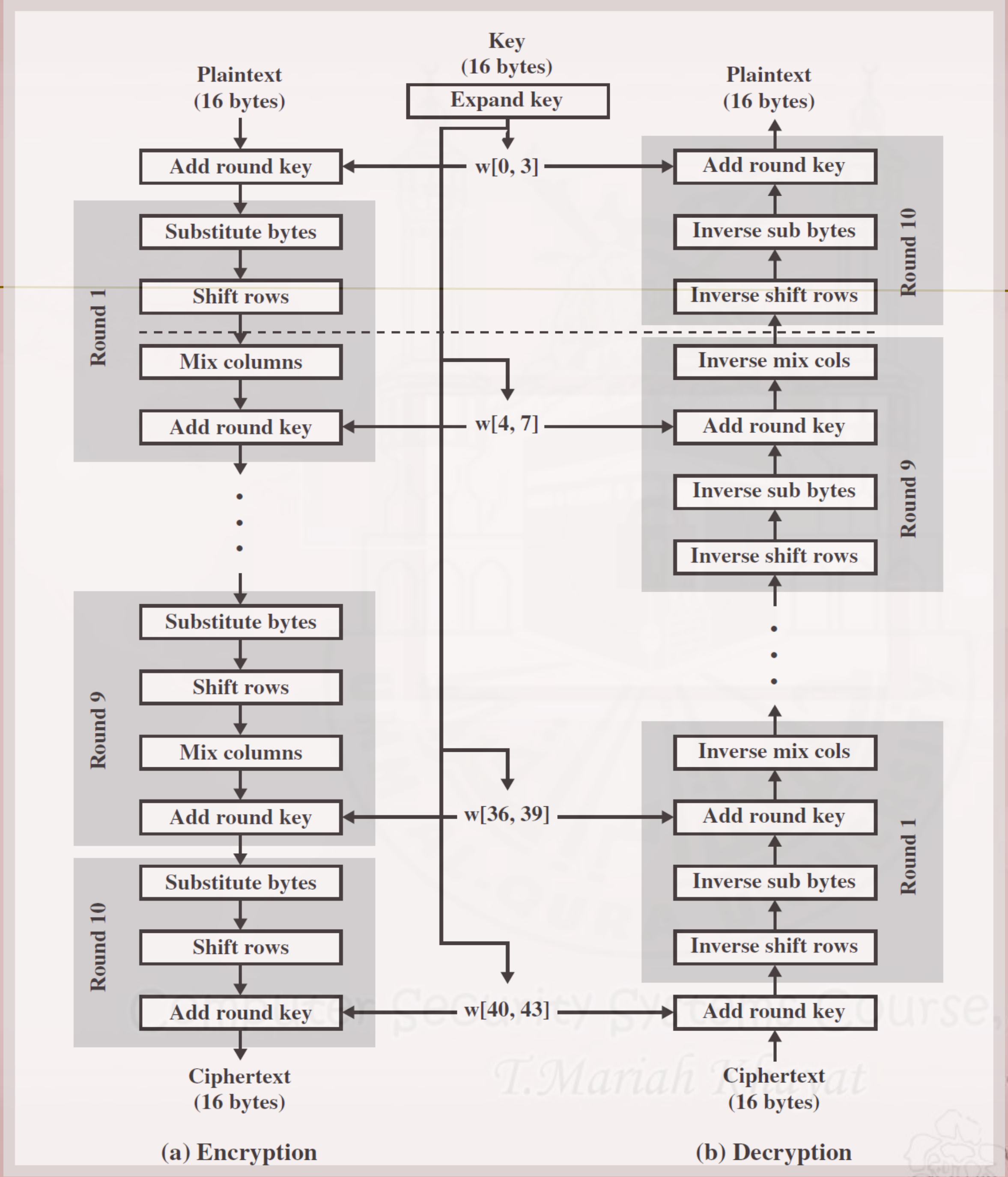
The AES Cipher - Rijndael

- **designed by Rijmen-Daemen in Belgium**
- **has 128/192/256 bit keys, 128 bit data**
- an **iterative** rather than **feistel** cipher
  - processes data as block of 4 columns of 4 bytes
  - operates on entire data block in every round
- **designed to be:**
  - resistant against known attacks
  - speed and code compactness on many CPUs
  - design simplicity

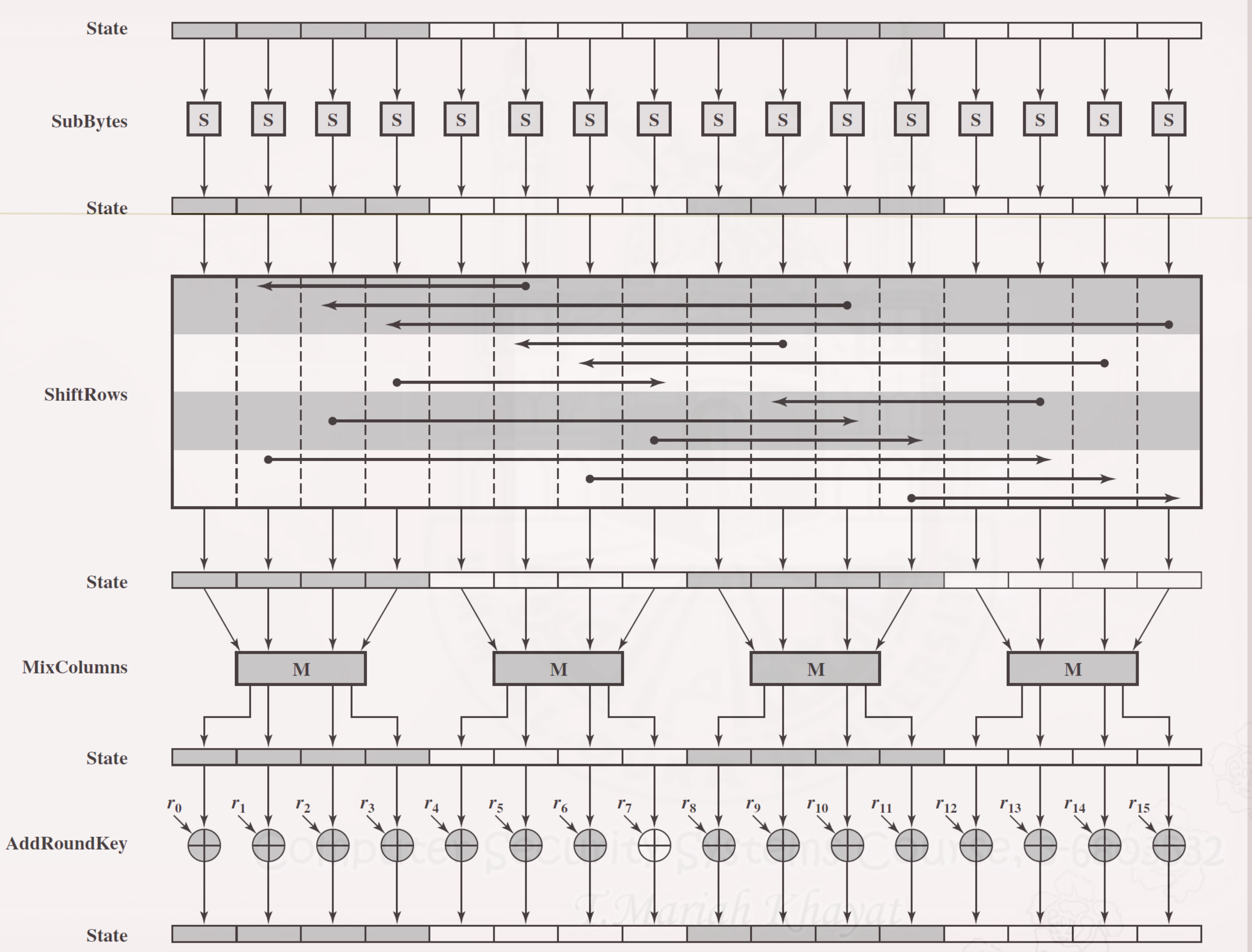
AES Structure

- ❖ **data block of 4 columns of 4 bytes is state**
- ❖ **key is expanded to array of words**
- ❖ **has 9/11/13 rounds in which state undergoes:**
  - byte substitution (1 S-box used on every byte)
  - shift rows (permute bytes between groups/columns)
  - mix columns (subs using matrix multiply of groups)
  - add round key (XOR state with key material)
  - view as alternating XOR key & scramble data bytes
- ❖ **initial XOR key material & incomplete last round**
- ❖ **with fast XOR & table lookup implementation**

AES Structure



AES Round


State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

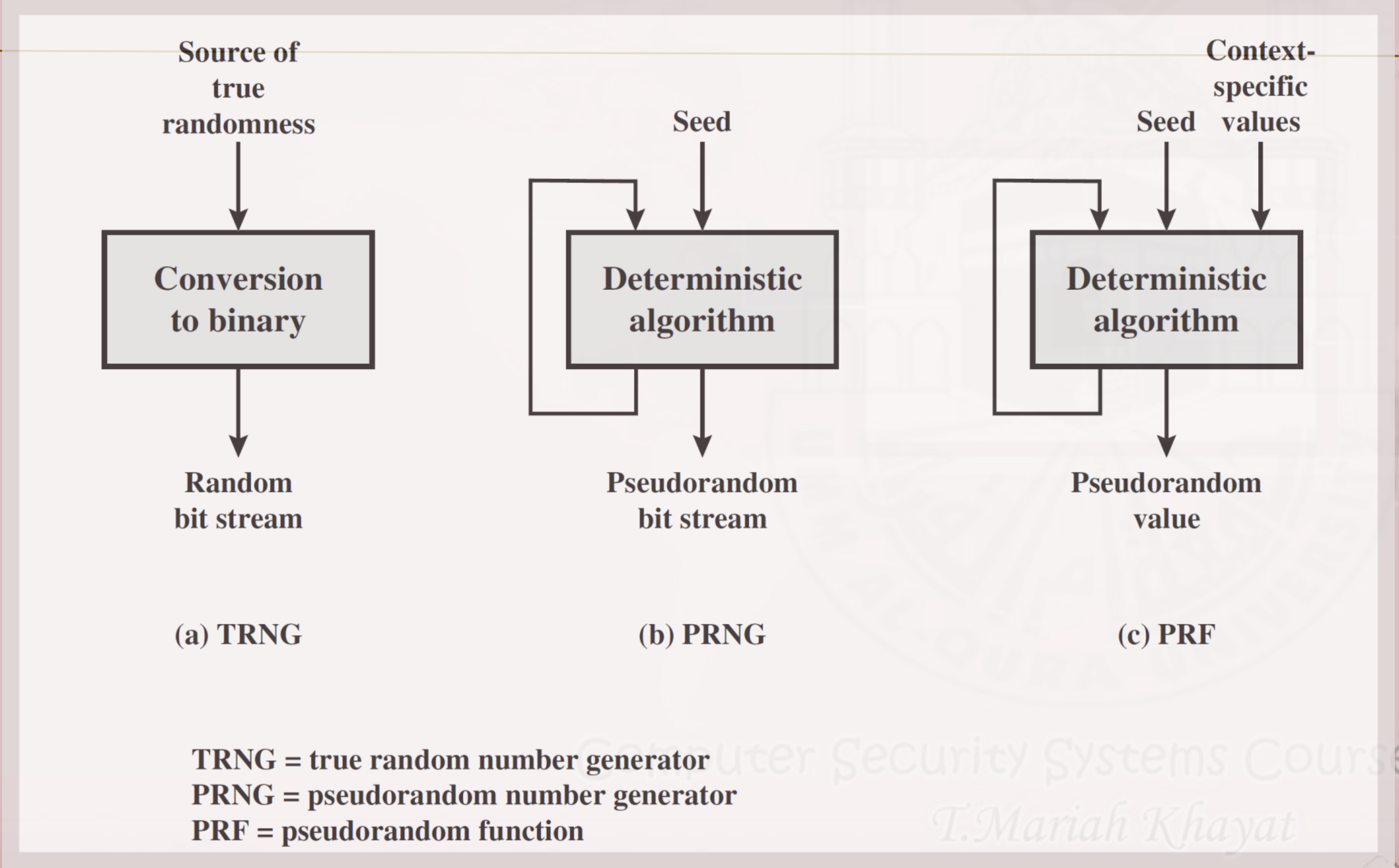
Random Numbers

- many uses of **random numbers** in cryptography
  - nonces in authentication protocols to prevent replay
  - session keys
  - public key generation
  - keystream for a one-time pad
- **in all cases its critical that these values be**
  - statistically random, uniform distribution, independent
  - unpredictability of future values from previous values
- **true random numbers provide this**
- **care needed with generated random numbers**

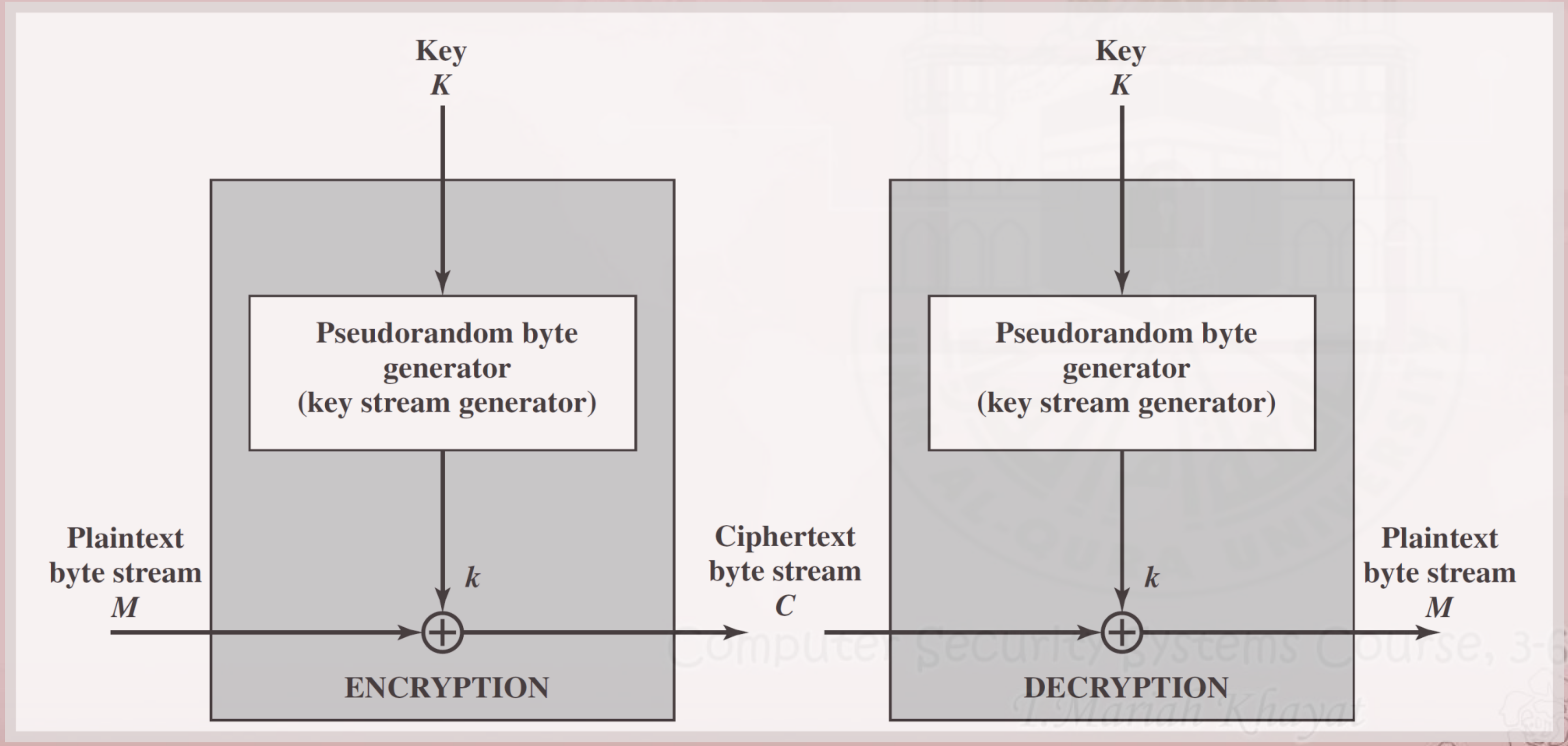
Pseudorandom Number Generators (PRNGs)

- often use deterministic algorithmic techniques to create "random numbers"
  - although are not truly random
  - can pass many tests of "randomness"
- known as "pseudorandom numbers"
- created by "Pseudorandom Number Generators (PRNGs)"

Random & Pseudorandom Number Generators



Stream cipher Structure



Stream cipher

- For example, if the next byte generated by the generator is **01101100** and the next plaintext byte is **11001100**, then the resulting ciphertext byte is

$$\begin{array}{rl} & 11001100 \text{ plaintext} \\ \oplus & 01101100 \text{ key stream} \\ \hline & 10100000 \text{ ciphertext} \end{array}$$

- Decryption requires the use of the same pseudorandom sequence:

$$\begin{array}{rl} & 10100000 \text{ ciphertext} \\ \oplus & 01101100 \text{ key stream} \\ \hline & 11001100 \text{ plaintext} \end{array}$$

Stream cipher Properties

- ❖ some design considerations are:
  - The encryption sequence should have a large period, the longer the period of repeat the more difficult it will be to do cryptanalysis.
  - statistically random
  - depends on large enough key
  - large linear complexity
- ❖ **properly designed, can be as secure as a block cipher with same size key**
- ❖ **but usually simpler & faster**

Rc4

- a proprietary cipher owned by RSA DSI
- another Ron Rivest design, simple but effective
- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS, wireless WEP/WPA)
- key forms random permutation of all 8-bit values
- uses that permutation to scramble input info processed a byte at a time

RC4 Key Schedule

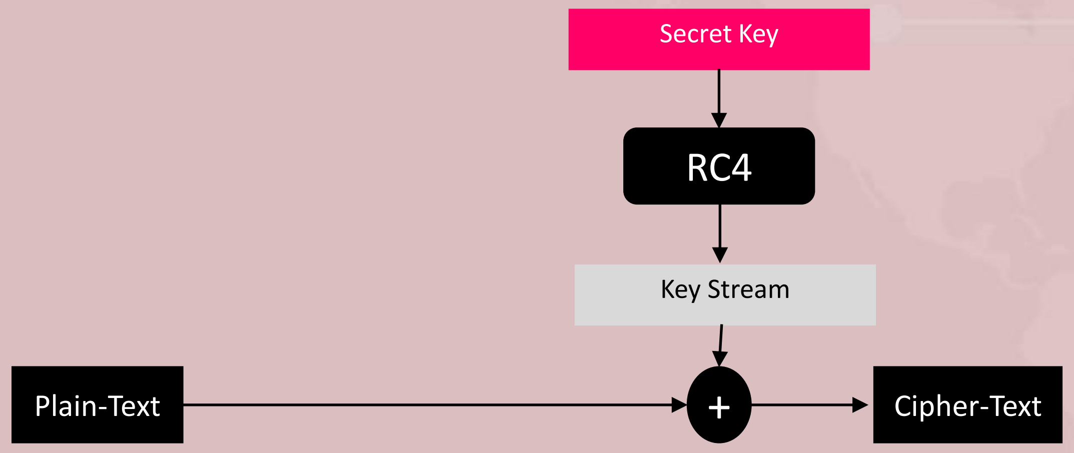
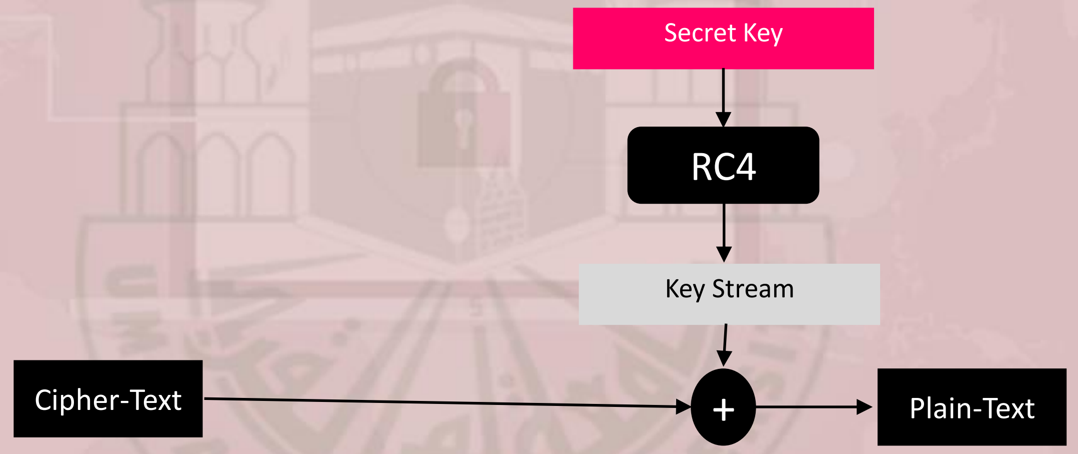
- starts with an array S of numbers: 0..255
- use key to well and truly shuffle
- S forms **internal state** of the cipher

```
for i = 0 to 255 do
  S[i] = i
  T[i] = K[i mod keylen])
j = 0
for i = 0 to 255 do
  j = (j + S[i] + T[i]) (mod 256)
  swap (S[i], S[j])
```

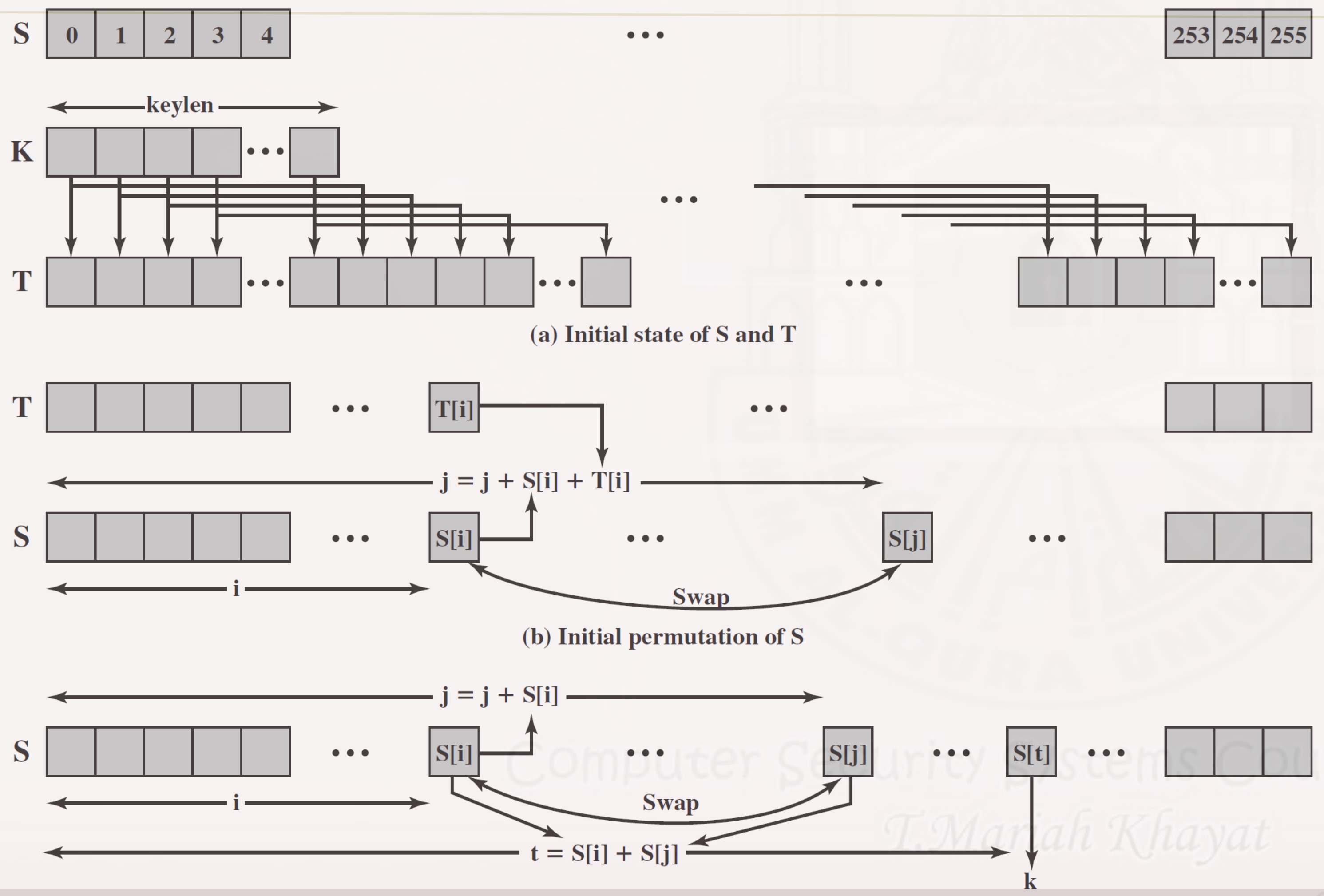
Rc4 Key Stream Generation

```
/* Stream Generation */
i, j = 0;
while (true)
i = (i + 1) mod 256;
j = (j + S[i]) mod 256;
Swap (S[i], S[j]);
t = (S[i] + S[j]) mod 256;
K = S[t];
```

RC4 Encryption and Decryption

Encryption	Decryption
	

RC4 Overview



RC4 Security

- ❖ claimed secure against known attacks
  - have some analyses, none practical
- ❖ result is very non-linear
- ❖ since RC4 is a stream cipher, must **never reuse a key**
- ❖ have a concern with WEP, but due to key handling rather than RC4 itself

Modes of Operation

- block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks with 56-bit key
- need some way to en/decrypt arbitrary amounts of data in practise
- NIST SP 800-38A defines 5 modes
- have **block** and **stream** modes
- to cover a wide variety of applications
- can be used with any block cipher

Electronic Codebook Book (ECB)

- message is broken into independent blocks which are encrypted
- each block is a value which is substituted, like a codebook, hence name
- each block is encoded independently of the other blocks

  $C_i = E_K(P_i)$
- uses: secure transmission of single values

Advantages and Limitations of ECB

- message repetitions may show in ciphertext
  - if aligned with message block
  - particularly with data such graphics
  - or with messages that change very little, which become a code-book analysis problem
- weakness is due to the encrypted message blocks being independent
- main use is sending a few blocks of data

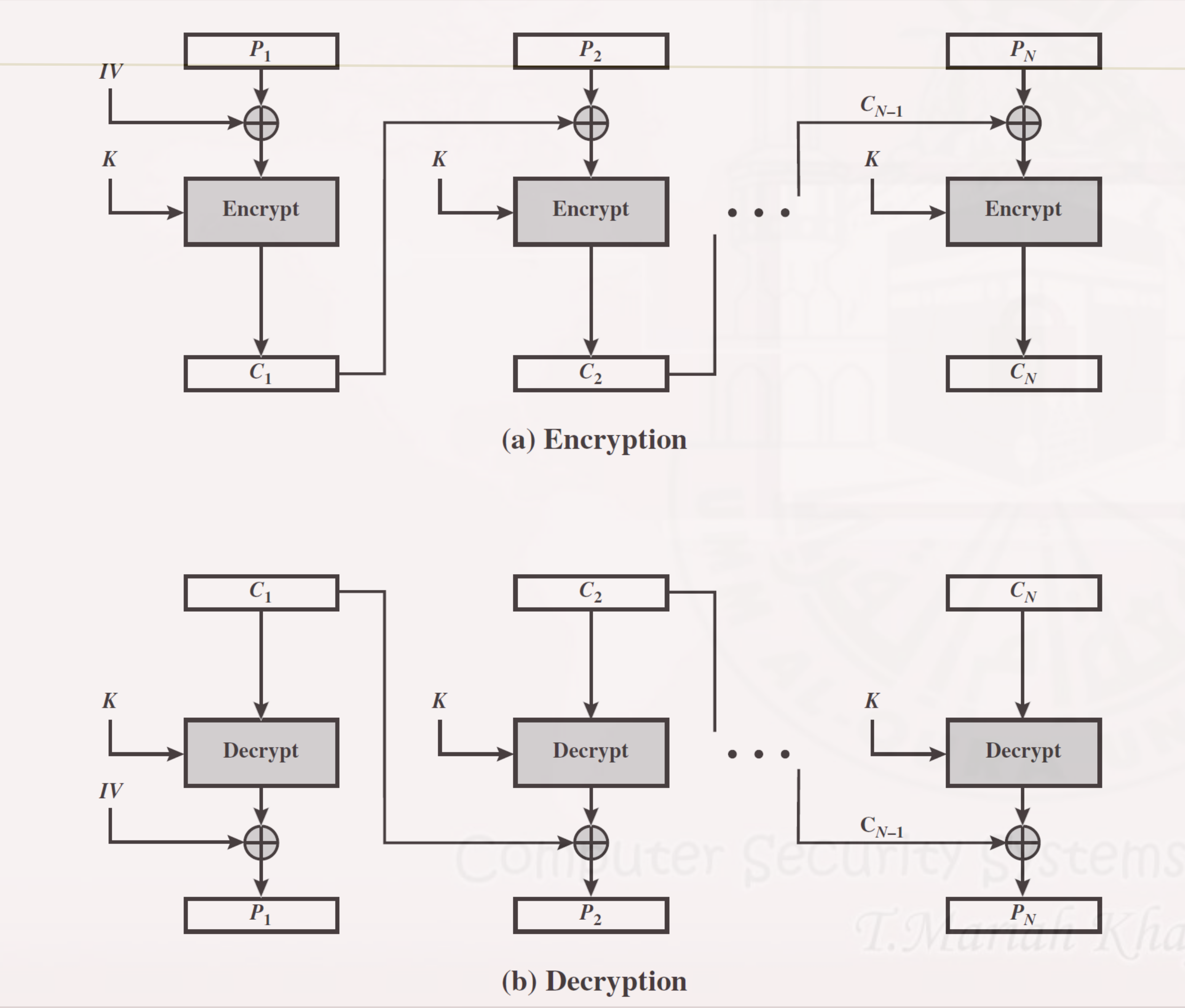
Cipher Block Chaining (CBC)

- message is broken into blocks
- linked together in encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process

  $C_i = E_K(P_i \text{ XOR } C_{i-1})$

  $C_{-1} = IV$
- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



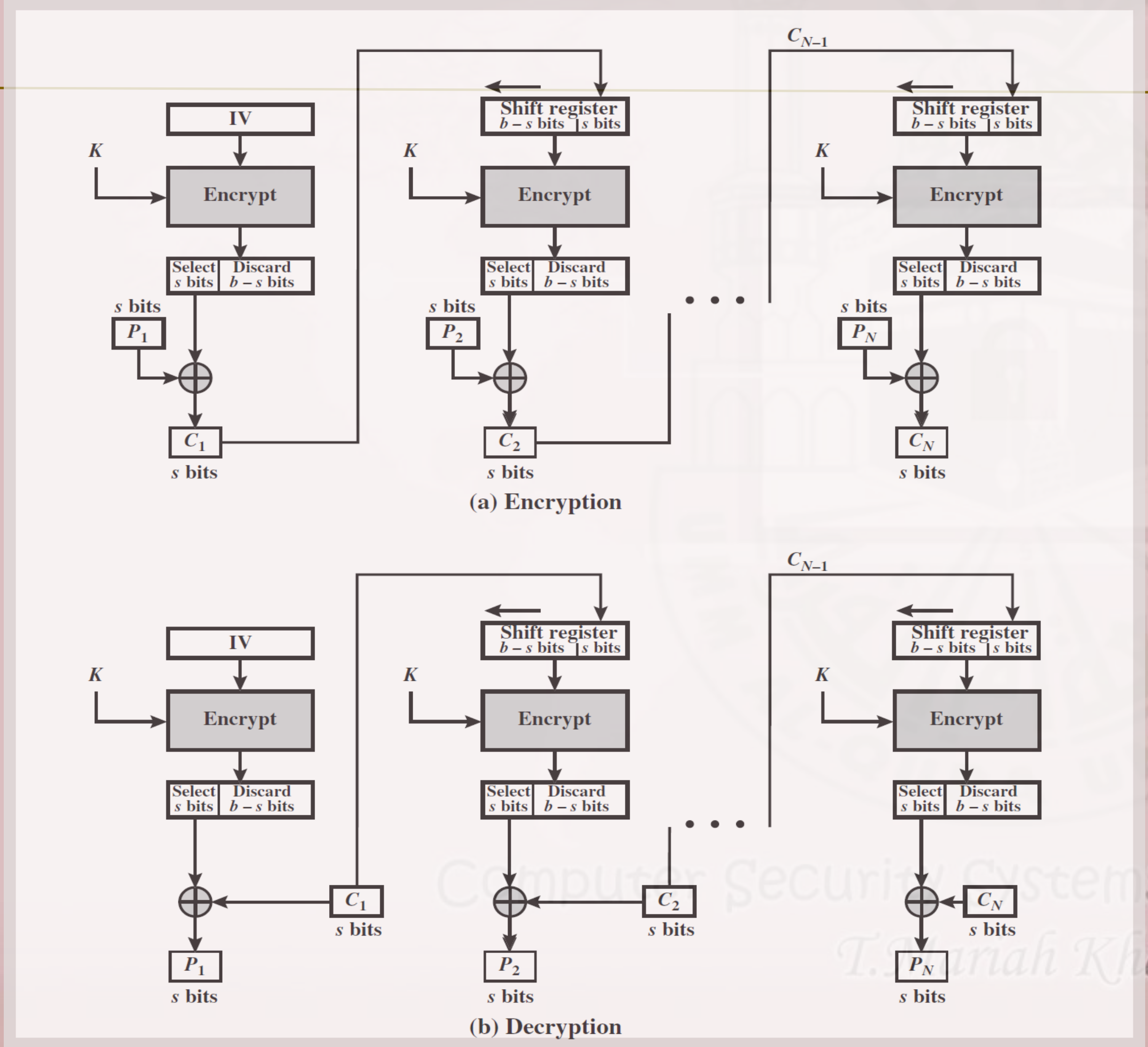
Cipher FeedBack (CFB)

- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8, 64 or 128 etc) to be feed back
  - denoted CFB-1, CFB-8, CFB-64, CFB-128 etc
- most efficient to use all bits in block (64 or 128)

  $C_i = P_i \text{ XOR } E_K(C_{i-1})$

  $C_{-1} = IV$
- uses: stream data encryption, authentication

S-bit Cipher FeedBack (CFB-S)



Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- limitation is need to stall while do block encryption after every n-bits
- note that the block cipher is used in **encryption** mode at **both** ends
- errors propagate for several blocks after the error

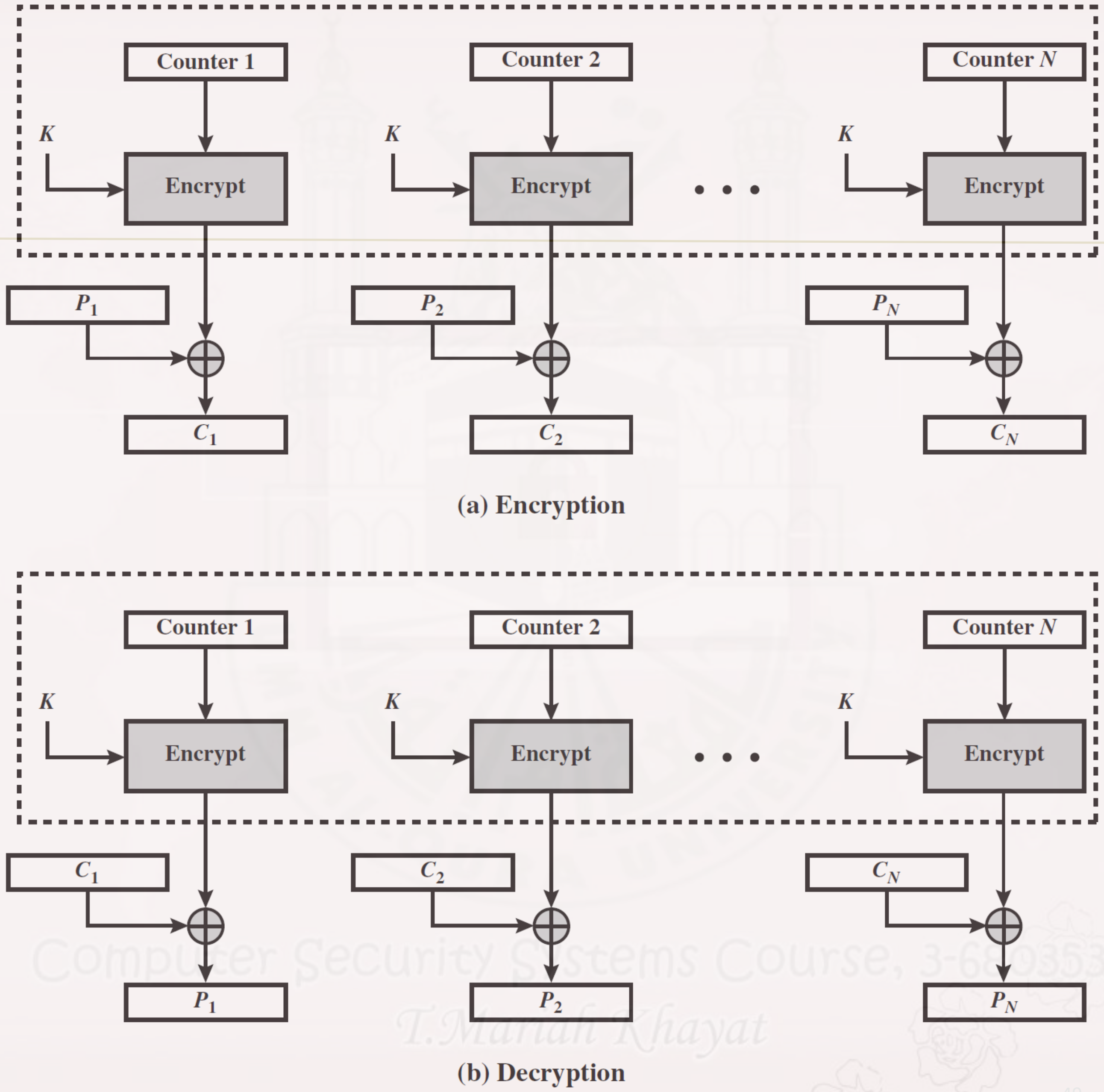
Counter (CTR)

- a "new" mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

$$O_i = E_K(i)$$
$$C_i = P_i \text{ XOR } O_i$$

- uses: high-speed network encryptions

Counter (CTR)



Counter 1
Counter 2
Counter N
K
Encrypt
P1
P2
PN
C1
C2
CN
(a) Encryption
Counter 1
Counter 2
Counter N
K
Encrypt
C1
C2
CN
P1
P2
PN
(b) Decryption

Advantages and Limitations of CTR

- **efficiency**
  - can do parallel encryptions in h/w or s/w
  - can preprocess in advance of need
  - good for bursty high speed links
- **random access to encrypted data blocks**
- **provable security (good as other modes)**
- **but must ensure never reuse key/counter values, otherwise could break (cf OFB)**

وصلّى الله وبارك على نبيّنا محمد

The End Summary of Chapter Two

T.Mariah Khayat
الأستاذة/ مارية خياط
Adham University College
الكلية الجامعية بأضم

mskhayat@uqu.edu.sa