Computer Security and Cryptography

CS381

来学嘉
计算机科学与工程系 电院3-423室
34205440 1356 4100825 laix@sjtu.edu.cn

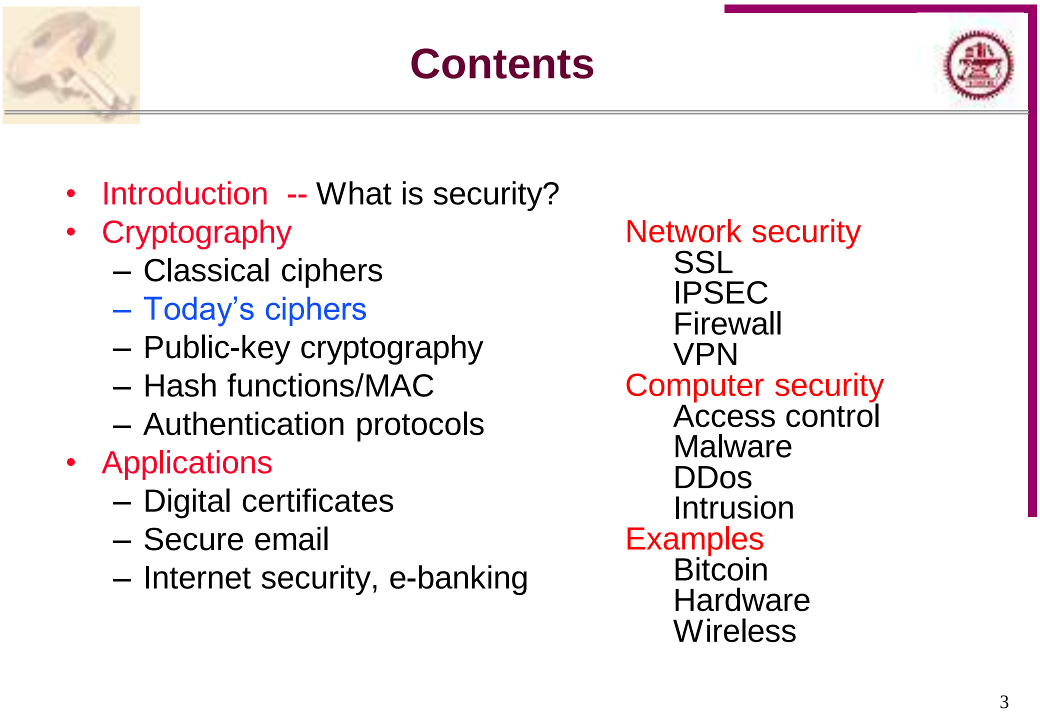
2016-03

Organization

- Week 1 to week 16 (2016-02-24 to 2016-06-08)
- 东上院502
- Monday 3-4节; week 9-16
- Wednesday 3-4节; week 1-16
- lecture 10 + exercise 40 + random tests 40 + other 10
- Ask questions **in** class – counted as points
- Turn ON your mobile phone (after lecture)
- Slides and papers:
  - http://202.120.38.185/CS381
    - **computer-security**
  - http://202.120.38.185/references
- TA: ‘薛伟佳’ xue_wei_jia@163.com，'黄格仕' <huang.ge.shi@foxmail.com>
- Send homework to: laix@sjtu.edu.cn and to TAs

Rule: do not disturb others!

2



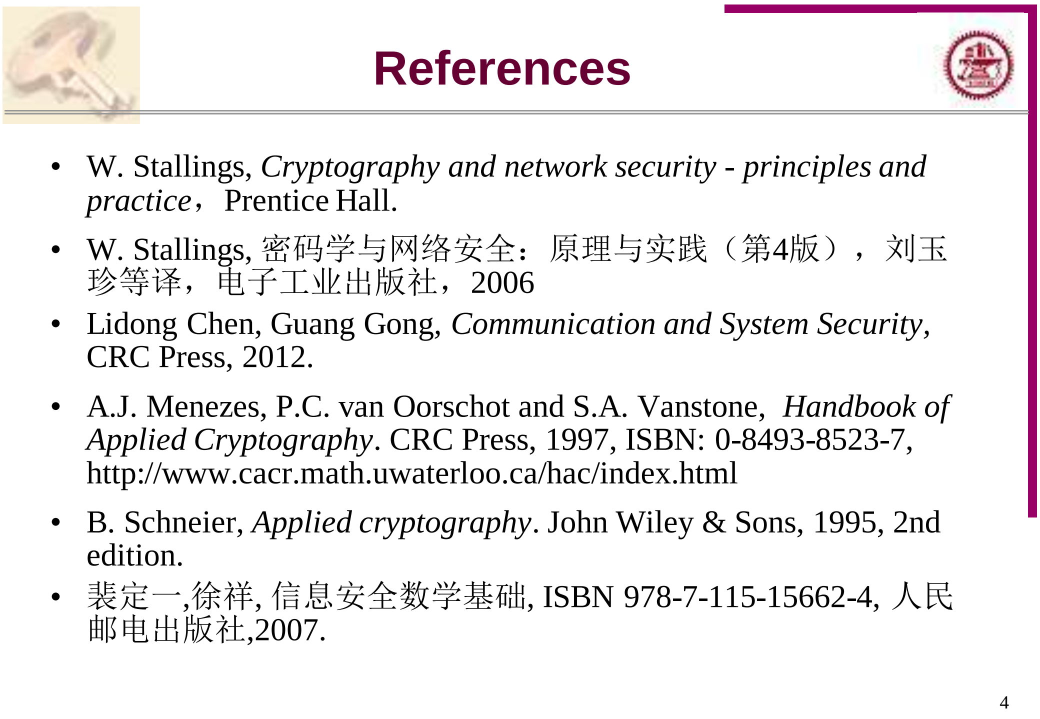
Contents

- Introduction -- What is security?
- Cryptography
  - – Classical ciphers
  - – Today's ciphers
  - – Public-key cryptography
  - – Hash functions/MAC
  - – Authentication protocols
- Applications
  - – Digital certificates
  - – Secure email
  - – Internet security, e-banking

Network security
SSL
IPSEC
Firewall
VPN

Computer security
Access control
Malware
DDos
Intrusion

Examples
Bitcoin
Hardware
Wireless



References

- W. Stallings, *Cryptography and network security - principles and practice*，Prentice Hall.
- W. Stallings, 密码学与网络安全：原理与实践（第4版），刘玉珍等译，电子工业出版社，2006
- Lidong Chen, Guang Gong, *Communication and System Security*, CRC Press, 2012.
- A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997, ISBN: 0-8493-8523-7, http://www.cacr.math.uwaterloo.ca/hac/index.html
- B. Schneier, *Applied cryptography*. John Wiley & Sons, 1995, 2nd edition.
- 裴定一,徐祥, 信息安全数学基础, ISBN 978-7-115-15662-4, 人民邮电出版社,2007.

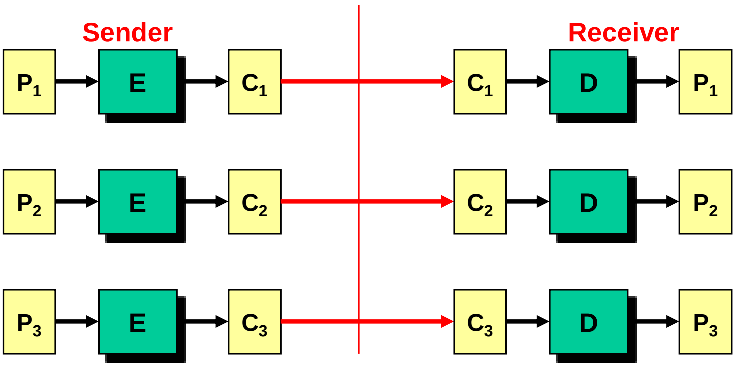
One-way functions

- Oneway function f: X ->Y, given x, easy to compute f(x); but for given y in f(X), it is hard to find x, s.t., f(x)=y.
  - Prob[f(A(f(x))=f(x))] < 1/p(n)   (TM definition, existence unknown)
  - Example: hash function, discrete logarithm;
- Keyed function f(X,Z)=Y, for known key z, it is easy to compute f(.,z)
  - Block cipher (fix c, f(c,.) is a oneway function)
- Keyed oneway function: f(X,Z)=Y, for known key z, it is easy to compute f(.,z) but for given y, it is hard to x,z, s.t., f(x,z)=y.
  - MAC function: keyed hash h(z,X), block cipher CBC
- Trapdoor oneway function $f_T(x)$: easy to compute and hard to invert, but with additional knowledge T, it is easy to invert.
  - Public-key cipher; RSA: $y=x^e$ mod N, T: N=p*q

Modes of Operation

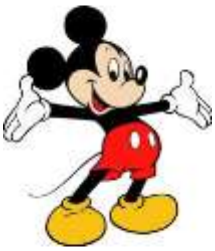
- block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks with 56-bit key
- need some way to en/decrypt arbitrary amounts of data in practise
- **ANSI X3.106-1983 Modes of Use** (now FIPS 81) defines 4 possible modes
- subsequently 5 defined for AES & DES
- have **block** and **stream** modes

Electronic Code Book Mode (ECB)

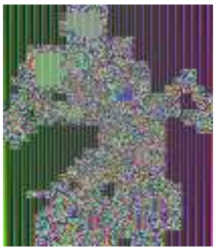


- each block is encoded independently
  $C_i = DES_{K1}(P_i)$
- Main use: secure transmission of single values such as keys
- Weakness: cannot hide data pattern.

Block cipher in stream mode



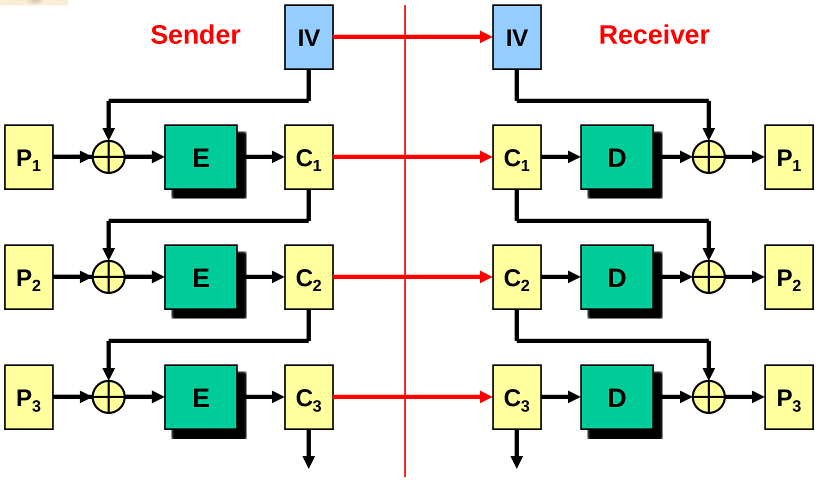
original



ECB mode

CBC mode

Cipher Block Chaining Mode (CBC)



use Initial Vector (IV)

$C_{-1} = IV, C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$;

uses: bulk data encryption, authentication-code, integrity

Advantages and Limitations of CBC

- a ciphertext block depends on **all** blocks before it
- change to a block affects all following ciphertext blocks (integrity, hide-pattern)
- Error propagation: an error in ciphertext affect only 2 plaintext blocks
- need **Initialization Vector** (IV)
  - which must be known to sender & receiver
  - if sent in clear, attacker can change bits of first block, and change IV to compensate
  - hence IV must either be a fixed value (as in EFTPOS)
  - or must be sent encrypted in ECB mode before rest of message

Message Padding

- at end of message must handle a possible last short block
  - which is not as large as blocksize of cipher
  - pad either with known non-data value (eg nulls)
  - or pad last block along with count of pad size
    - eg. [b1 b2 b3 0 0 0 0 5]
    - means have 3 data bytes, then 5 bytes pad+count
  - this may require an extra entire block over those in message
- there are other, more esoteric modes, which avoid the need for an extra block

Using CBC for MACs

- can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
    - or the leftmost M bits (16≤M≤64) of final block
- MAC is used for message authentication and integrity

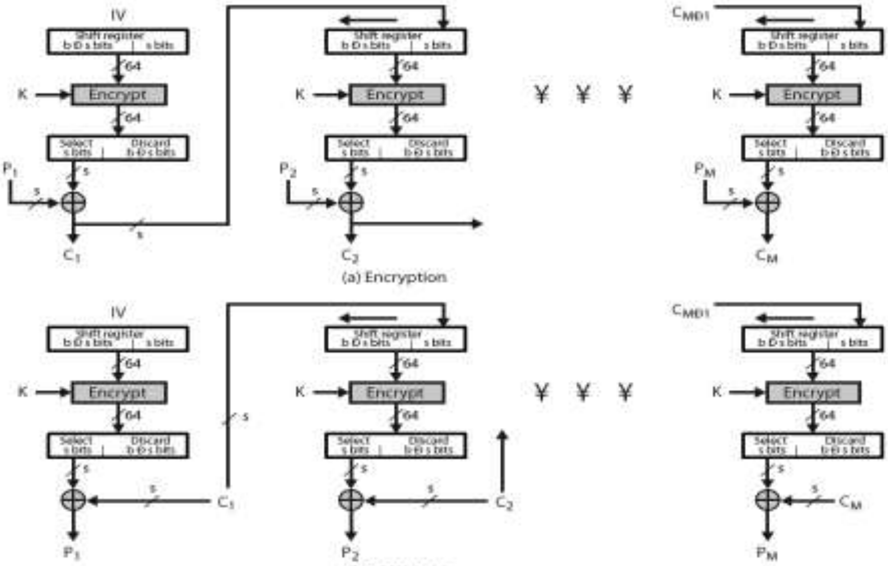
Cipher FeedBack (CFB)

- 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 XOR $DES_{K1}(C_{i-1})$

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

Cipher FeedBack (CFB)



(a) Encryption

(b) Decryption

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

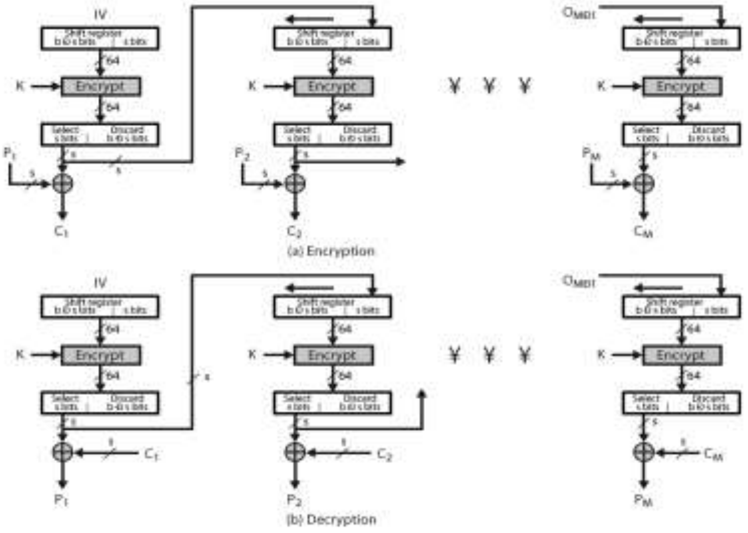
Output FeedBack (OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

$$C_i = P_i \text{ XOR } O_i$$
$$O_i = \text{DES}_{K1}(O_{i-1})$$
$$O_{-1} = \text{IV}$$

- uses: stream encryption on noisy channels

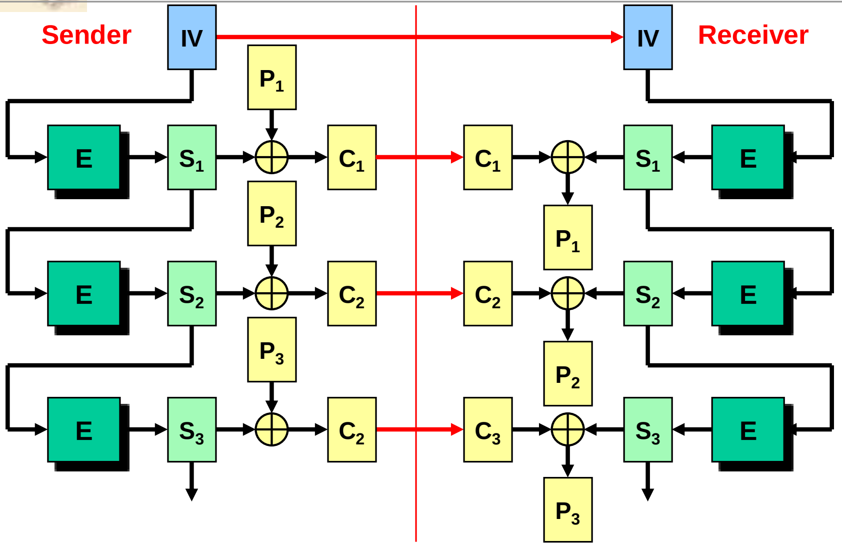
Output FeedBack (OFB)



(a) Encryption

(b) Decryption

Output FeedBack (OFB)



Advantages and Limitations of OFB

- bit errors do not propagate
- more vulnerable to message stream modification
- a variation of a Vernam cipher
  - hence must **never** reuse the same sequence (key+IV)
- sender & receiver must remain in sync
- originally specified with m-bit feedback
- subsequent research has shown that only **full block feedback** (ie CFB-64 or CFB-128) should ever be used

Counter (CTR)

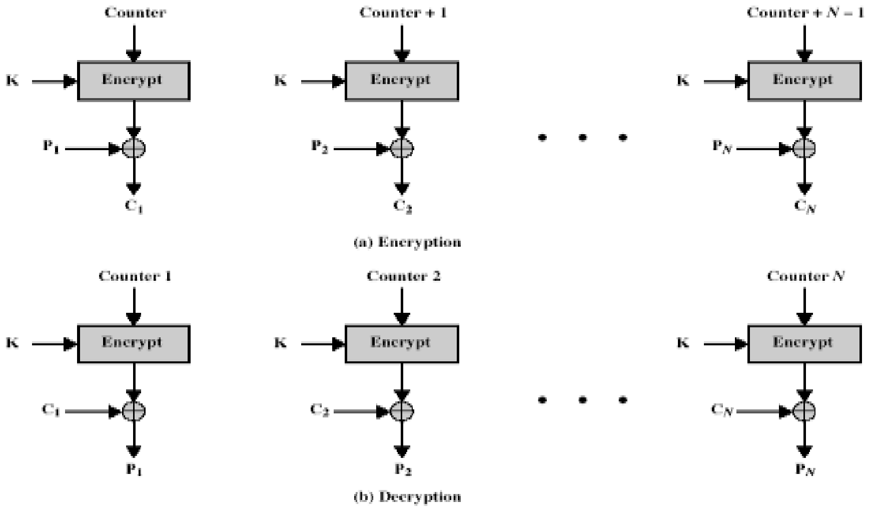
- a “new” mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

$$C_i = P_i \text{ XOR } O_i$$

$$O_i = DES_{K1}(i)$$

- uses: high-speed network encryptions

Counter (CTR)



(a) Encryption

(b) Decryption

Advantages and Limitations of CTR

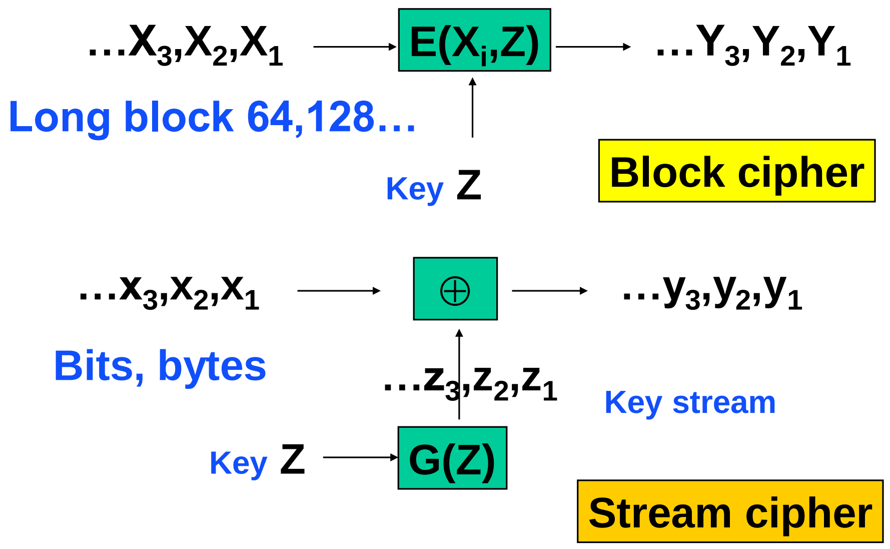
- efficiency
  - can do parallel encryptions in h/w or s/w
  - can preprocess in advance of need
  - good for bursty high speed links
- random access to encrypted data blocks
- provable security (good as other modes)
- but must ensure never reuse key/counter values, otherwise could break (cf OFB)

Symmetric cipher system

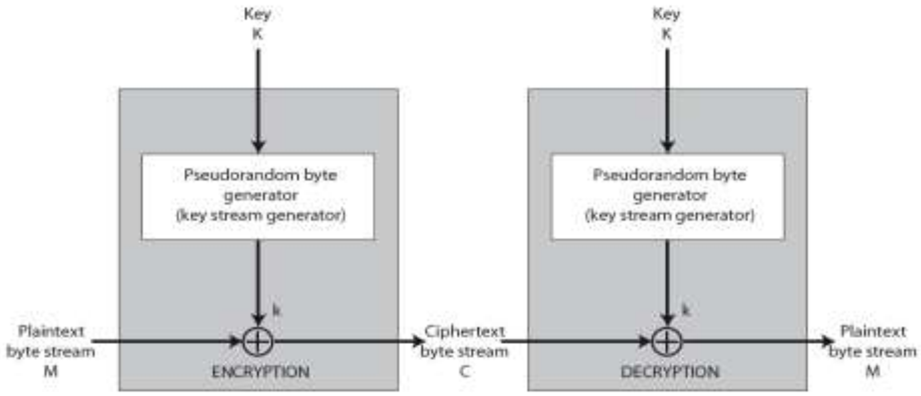
- block cipher(分组密码): plaintexts are devided into blocks of the same length (64, 128 bits), each block is encrypted to ciphertext block with the same length, under the control of the same secret key.
  - The same function is used to encrypt successive blocks ⇒ memoryless
- stream cipher(流密码,序列密码): plaintext is processed as a sequence of bits or bytes.
  - Encryption transformation varies with time ⇒ have memory

23

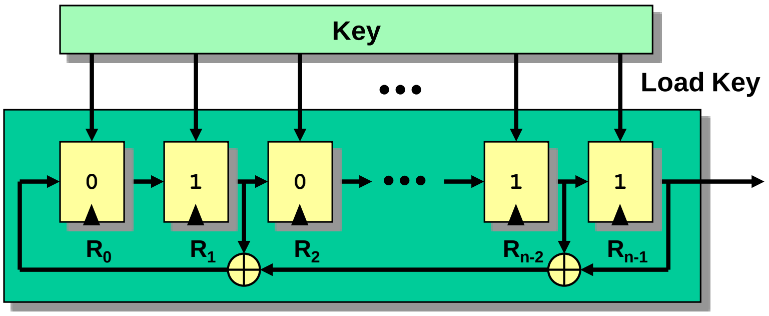
Block cipher and stream cipher



Stream Cipher Structure

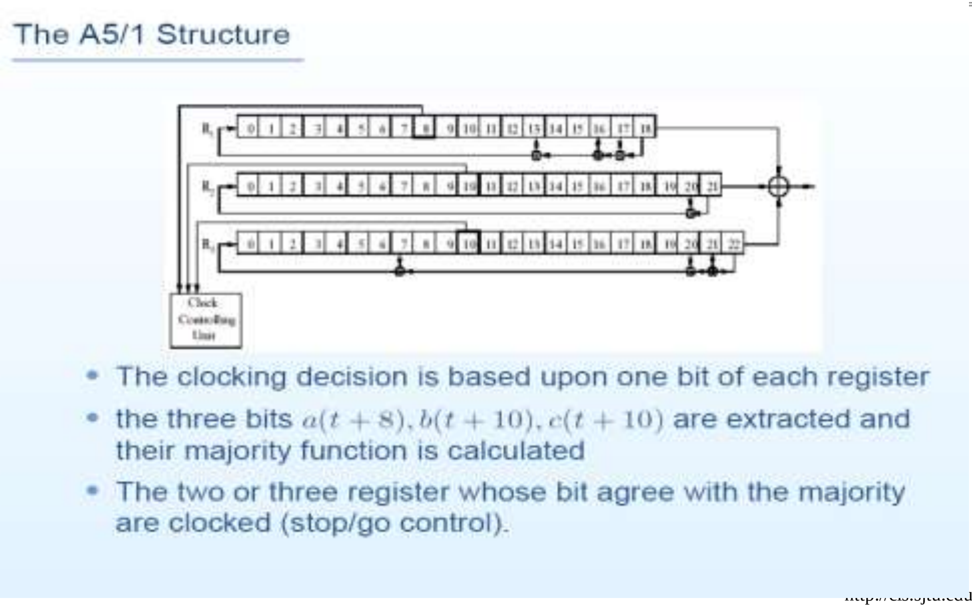


Linear Feedback Shift Register (LFSR)



- Maximum possible sequence length is $2^n - 1$ with n registers
- LFSRs are often used as building blocks for stream ciphers
- GSM A5 is a cipher with 3 LFSRs of lengths 19, 22, and 23
- Berlekamp-Massey algorithm (1969): can recover whole sequence with complexity: data 2n, comp n^2

A5/1



The A5/1 Structure

- The clocking decision is based upon one bit of each register
- the three bits $a(t+8)$, $b(t+10)$, $c(t+10)$ are extracted and their majority function is calculated
- The two or three register whose bit agree with the majority are clocked (stop/go control).

Stream cipher A5 in GSM

- A5/1 is used in GSM (1987) to protect the confidentiality of voice communications
- A5/2 was (1989) a weaker version for export .
- Both were initially kept secret.
- The general design was leaked in 1994, and the algorithms were entirely reverse engineered in 1999 by Marc Briceno from a GSM telephone.
- In 2000, around 130 million GSM customers, by 2011, it was 4 billion (not in China)
- [Barkan-Biham-Keller 2006] attacks on A5/1, A5/3, or even GPRS that allow attackers to tap GSM mobile phone conversations and decrypt in real-time, or at any later time.

RC4

- Designed by Ron Rivest for RSA DSI in 1987
- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS, wireless WEP)
- key forms random permutation of all 8-bit values
- uses that permutation to scramble input info processed a byte at a time

- RC4 is a stream cipher, must **never reuse a keystream**
- various weakness found in various implementations
- If implemented carefully, RC4 can provide adequate security

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])
```

Encryption

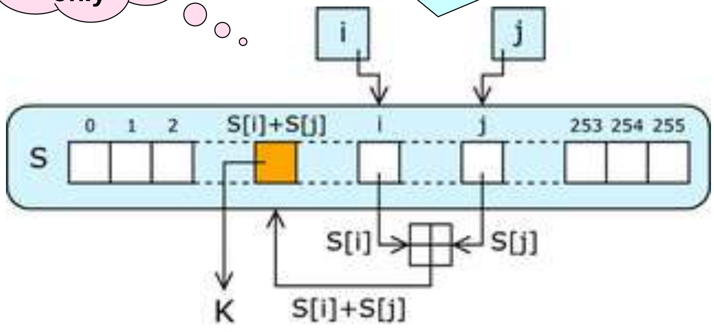
- encryption continues shuffling array values
- sum of shuffled pair selects "stream key" value from permutation
- XOR S[t] with next byte of message to en/decrypt

```
i = j = 0
for each message byte $M_i$
    i = (i + 1) (mod 256)
    j = (j + S[i]) (mod 256)
    swap(S[i], S[j])
    t = (S[i] + S[j]) (mod 256)
    $C_i$ = $M_i$ XOR S[t]
```

RC4

```
i := 0
j := 0 while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(S[i],S[j])
    output S[(S[i] + S[j]) mod 256]
```

continues shuffling only



$C_i \ldots = \leftarrow \oplus \leftarrow M_i, \ldots$

Random Numbers

- usage of random numbers in cryptography
  - nonce in authentication protocols to prevent replay
  - session keys
  - public key generation
  - Key-stream for one-time pad
- True random numbers
  - uniform distribution, independent
  - By Coin flipping, Quantum, physical noise
- Pseudo-random number
  - The generation is deterministic, but requires
  - Unpredictable: cannot infer future sequence on previous values

Examples of PRNG

- **Fibonacci sequence (1202):**

$$f_{n+1} = f_n + f_{n-1}$$

- **Linear Congruential Generator** (Lehmer 1948)

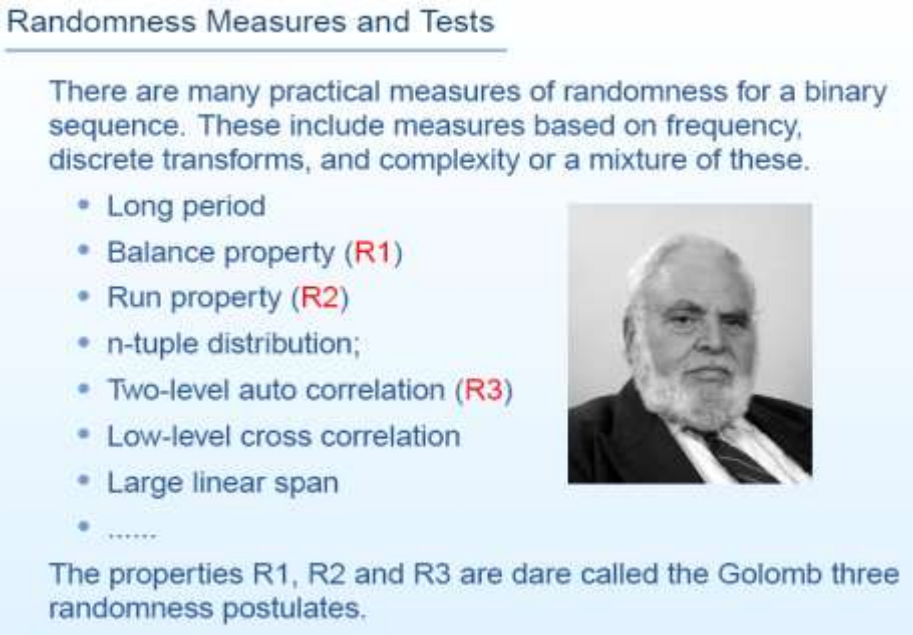
$$f_{n+1} = af_n + c \quad mod\ m$$

- **linear feedback shift register** (ore 1934)

$$f_{k+n} = c_0 f_k + c_1 f_{k+1} + \ldots + c_{n-1} f_{k+n-1} \quad c_i \in F_2$$

- **Blum Blum Shub (1986)**

$$f_{n+1} = f_n^2 \quad mod\ m \qquad (m=p*q,\ output\ some\ bits)$$

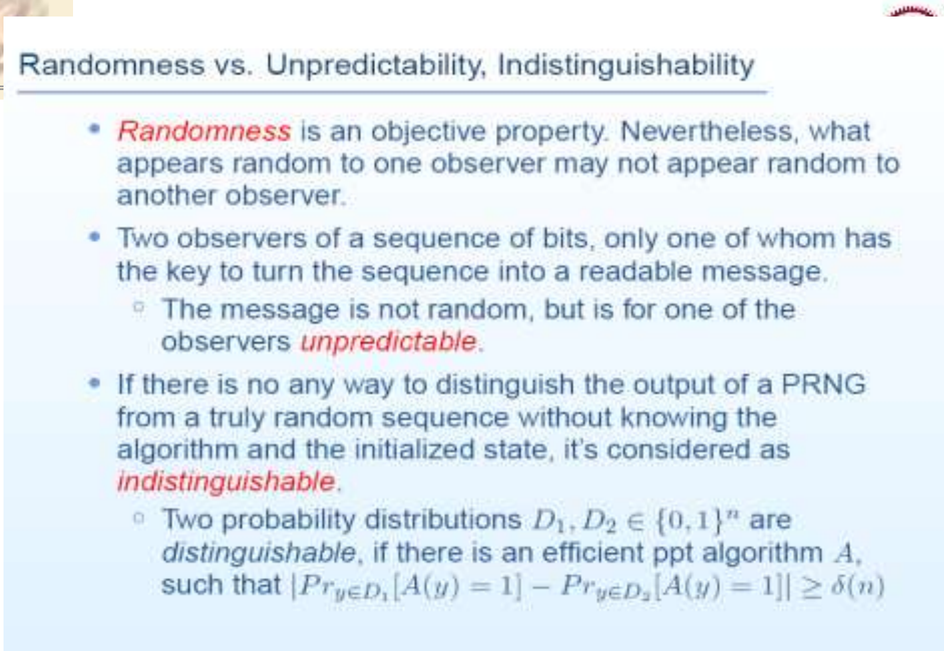


Randomness Measures and Tests

There are many practical measures of randomness for a binary sequence. These include measures based on frequency, discrete transforms, and complexity or a mixture of these.

- Long period
- Balance property (R1)
- Run property (R2)
- n-tuple distribution;
- Two-level auto correlation (R3)
- Low-level cross correlation
- Large linear span
- ……

The properties R1, R2 and R3 are dare called the Golomb three randomness postulates.



Randomness vs. Unpredictability, Indistinguishability

- *Randomness* is an objective property. Nevertheless, what appears random to one observer may not appear random to another observer.
- Two observers of a sequence of bits, only one of whom has the key to turn the sequence into a readable message.
  - The message is not random, but is for one of the observers *unpredictable*.
- If there is no any way to distinguish the output of a PRNG from a truly random sequence without knowing the algorithm and the initialized state, it's considered as *indistinguishable*.
  - Two probability distributions $D_1, D_2 \in \{0,1\}^n$ are *distinguishable*, if there is an efficient ppt algorithm A, such that $|Pr_{y \in D_1}[A(y)=1] - Pr_{y \in D_2}[A(y)=1]| \geq \delta(n)$

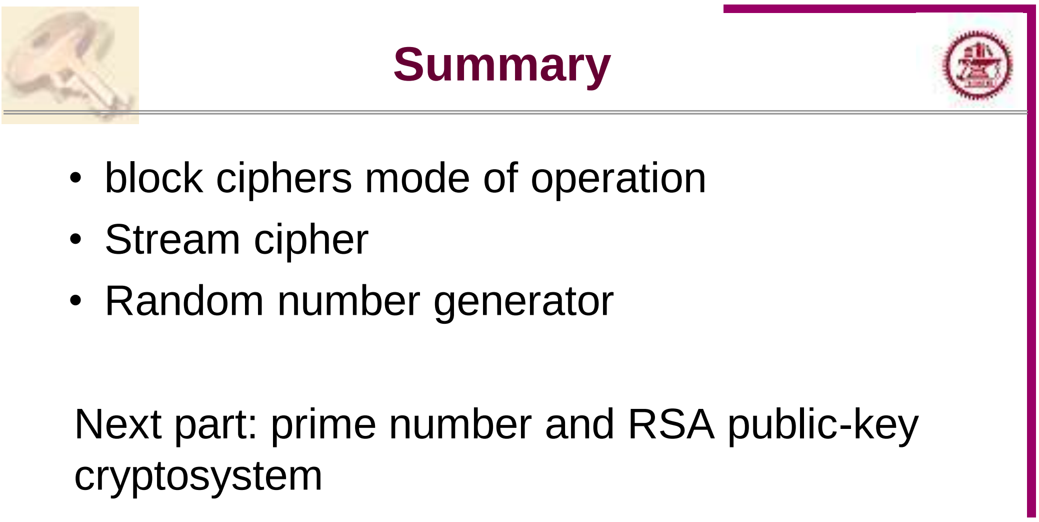
Exercise 6 – stream

1. Design a method of padding for DES, so that you can encrypt an 80-bit plaintext into 80-bit ciphertext in ECB mode. Can we do this with AES?

2. Prove that in CBC mode, an error in ciphertext affects only 2 plaintext blocks;

3. Is pseudo-random number generator a one-way function, and why?

Deadline: 1 day before next lecture
Format: Subject: CS381--EX.#-某某某



Summary

- block ciphers mode of operation
- Stream cipher
- Random number generator

Next part: prime number and RSA public-key cryptosystem