Computer Security and Cryptography

CS381

4. Block cipher DES

来学嘉　　计算机科学与工程系　电院3-423室
34205440　13564100825　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 function

- The intrinsic problem of information security is the "one-wayness".
- Cryptography studies one-way function
  - The measure of one-way: difficulty
  - The design of one-way function
  - The attacks on one-way function
  - The use of one-way function

5

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,K)=Y, for known key z, it is easy to compute f(.,k)
  - Block cipher (fix c, f(c,.) is a oneway function)
- Keyed oneway function: f(X,K)=Y, for known key k, it is easy to compute f(.,k) but for given y, it is hard to x,k, s.t., f(x,k)=y.
  - MAC function: keyed hash h(k,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

6

Block cipher

- A block cipher is a mapping E: $\left\{ \begin{array}{l} F_2^m \times F_2^k \rightarrow F_2^m \\ (X, K) \rightarrow Y \end{array} \right.$

  s.t. $E(;k)$ is invertible for each k

- A block cipher should be easy to use and hard to break.
  - easy to use: the encryption function $E(;k)$ and decryption function $D(;k)$ are easy to compute for all k
  - hard to break: It is hard for attacker

    to determine the key k (totally break) or

    to recover plaintexts often (partially break)

7

Cipher parameters

- Plaintext size m:
  - Large enough to defeat statistical analysis
  - Multiple of 8 (byte size)
  - 64-bit (DES, IDEA): adequate for encryption, too small for hash;
  - 128-bit (AES): current standard size, too small for hash;
  - 256-bit: big enough but speed and security could suffer, need more rounds for confusion and diffusion.
- Key size k:
  - 40-bit: so government can read your secret, still in use.
  - 64-bit is too small for today,
  - 128-bit is secure enough against exhaustive search
  - 256, 512, 1024-bit: good for hash, fit into RSA for key-encryption; security up bound can be hard to achieve.
  - If someone is sell a cipher with 5000-bit key “my cipher is more secure), then he is probably selling snake oil (忽悠)

8

Design principles

implementation

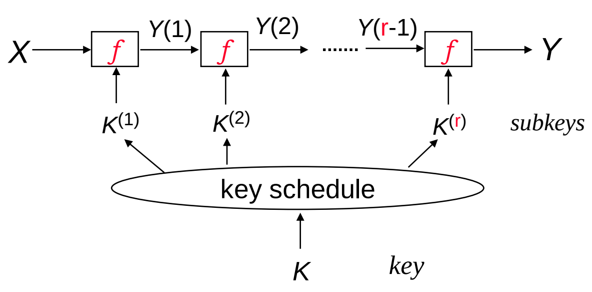
- software
  - use operations on subblocks
  - simple and available operations
  - use small look-up tables
- hardware
  - regular structure
  - similarity of encryption and decryption
    - differ in the way of using the key (key schedule)

security

- confusion
  - hard to determine the key from known plaintext and ciphertext
    - e.g: "highly nonlinear"
- diffusion
  - no useful dependence between plaintext X and ciphertext $Y=E(X,k)$ for virtually all k
    - e.g. function E(.,.) should be "complete", i.e., each bit of (X,K) influences every bit of Y.
- Resist known attacks

9

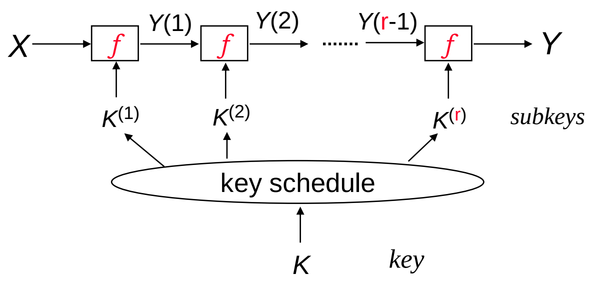
Iterated block ciphers



- f: round function is easy to implement
- k: the key
- $k^{(i)}$ the round subkey generated from k by an algorithm called key schedule
- r: number of rounds (iterations)

10

Reasons for Iteration



- a simple round function f is easy to implement
- iterations provide confusion and diffusion
  - complexity of differential analysis $\approx c^r$ (under certain conditions)
  - complexity of implementation $\approx rc(f)$

DES (Data Encryption Standard)

- 1972 the NBS (National Bureau of Standards), now NIST (the National Institute of Standards and Technology), call for encryption algorithm that could be standardized.
- 1974, IBM responded with a design based on their 'Lucifer' algorithm.
  - Data Encryption Standard. FIPS PUB 46, Appendix A, Federal Information Processing Standards Publication, January 15, 1977, US Dept. of Commerce, National Bureau of Standards.
  - C. Meyer and S. Matyas: Cryptography, John Willy & Sons, 1982.

DES

- DES has a effective key-length of 56 bits and a block-length of m = 64 bits.
- A DES key consists of 64 bits, of which 56 bits are randomly generated and used directly by the algorithm.
  - The effective key length is 56 bits
  - The other 8 bits are the parity check: there is an odd number of "1"s in each 8-bit byte.

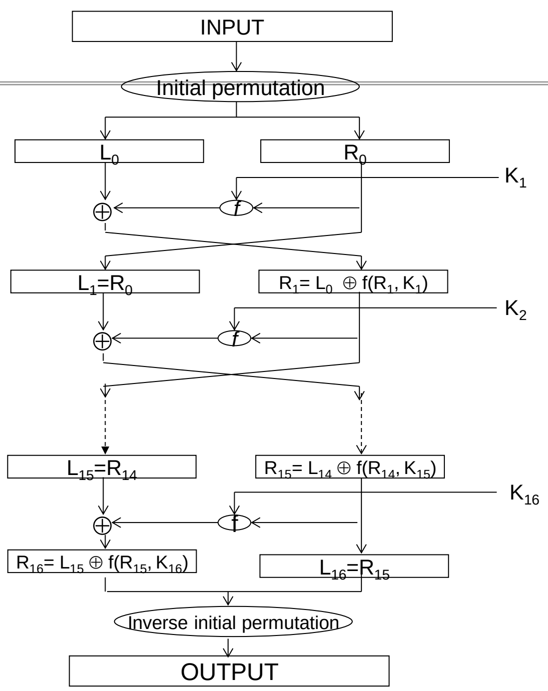
☆ DES

r=16 rounds

w=32

block length 64 bits

Subkeys K_i, 48bits



Initial permutation

the first bit of L_0R_0 is the 58th bit of *M*,
the second bit is the 50th bit of M
and so on,
the last, the 64th bit of L_0R_0 being the 7th bit of *M*

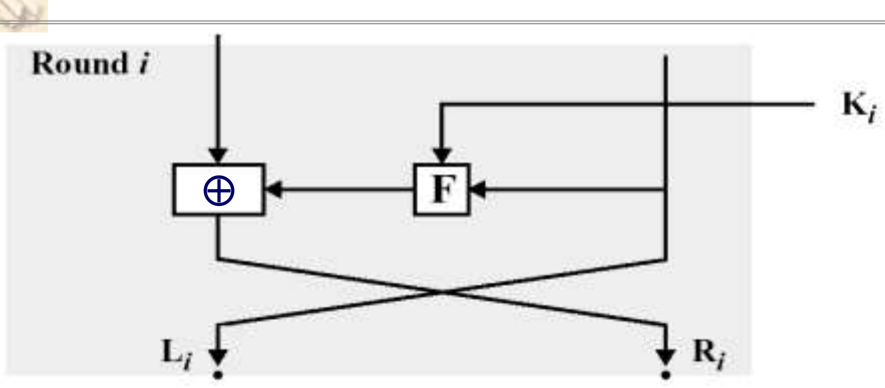
Based on implementation consideration;
no cryptographic significance

- IP

58 50 42 34 26 18 10 2
60 52 44 36 28 20 12 4
62 54 46 38 30 22 14 6
64 56 48 40 32 24 16 8
----
57 49 41 33 25 17 9 1
59 51 43 35 27 19 11 3
61 53 45 37 29 21 13 5
63 55 47 39 31 23 15 7

15

☆ Feistel structure



•encryption: $L_i = R_{i-1}$; $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

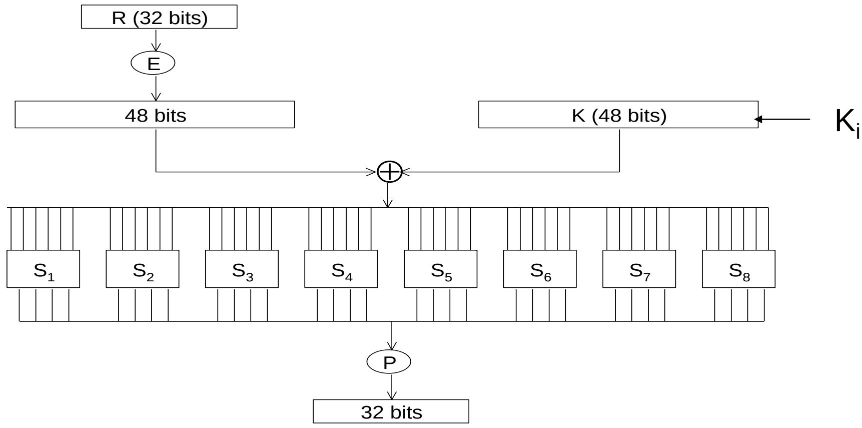
•decryption: $R_{i-1} = L_i$

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

$= R_i \oplus F(L_i, K_i)$

16

DES F function



17

The expansion E

The 32 bits of R are expanded into 48 bits as 6x8 inputs for 8 S-boxes

1st and 6th bits are used to choose the row number, the middle 4 bits determine the column

- To achieve better diffusion effect of input bits
- 1-bit change in input difference causes >1 bits change in output difference

1. 32 | 01 02 03 04 | 05
2. 04 | 05 06 07 08 | 09
3. 08 | 09 10 11 12 | 13
4. 12 | 13 14 15 16 | 17
5. 16 | 17 18 19 20 | 21
6. 20 | 21 22 23 24 | 25
7. 24 | 25 26 27 28 | 29
8. 28 | 29 30 31 32 | 01

18

8 S-boxes

S1
14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7
0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8
4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0
15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13
S2
15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10
3 13 4 7 15 2 8 14 12 0 1 10 6 9 11 5
0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15
13 8 10 1 3 15 4 2 11 6 7 12 0 5 14 9
S3
10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8
13 7 0 9 3 4 6 10 2 8 5 14 12 11 15 1
13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7
1 10 13 0 6 9 8 7 4 15 14 3 11 5 2 12
S4
7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15
13 8 11 5 6 15 0 3 4 7 2 12 1 10 14 9
10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4
3 15 0 6 10 1 13 8 9 4 5 11 12 7 2 14

S5
2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9
14 11 2 12 4 7 13 1 5 0 15 10 3 9 8 6
4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14
11 8 12 7 1 14 2 13 6 15 0 9 10 4 5 3
S6
12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11
10 15 4 2 7 12 9 5 6 1 13 14 0 11 3 8
9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6
4 3 2 12 9 5 15 10 11 14 1 7 6 0 8 13
S7
4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1
13 0 11 7 4 9 1 10 14 3 5 12 2 15 8 6
1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2
6 11 13 8 1 4 10 7 9 5 0 15 14 2 3 12
S8
13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7
1 15 13 8 10 3 7 4 12 5 6 11 0 14 9 2
7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8
2 1 14 7 4 10 8 13 15 12 9 0 3 5 6 11

Permutations of S1

S1

Column Number

Row No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S1-0

0 >14> 0; 1>4>2 >13>9 >10>6 >11>12 >5 >15 >7 >8 >3 >1

S1-1

0>0; 1>15>8>10>12>9>6>13>5>2>7>1; 4>14>3>4; 11>11

Design principles

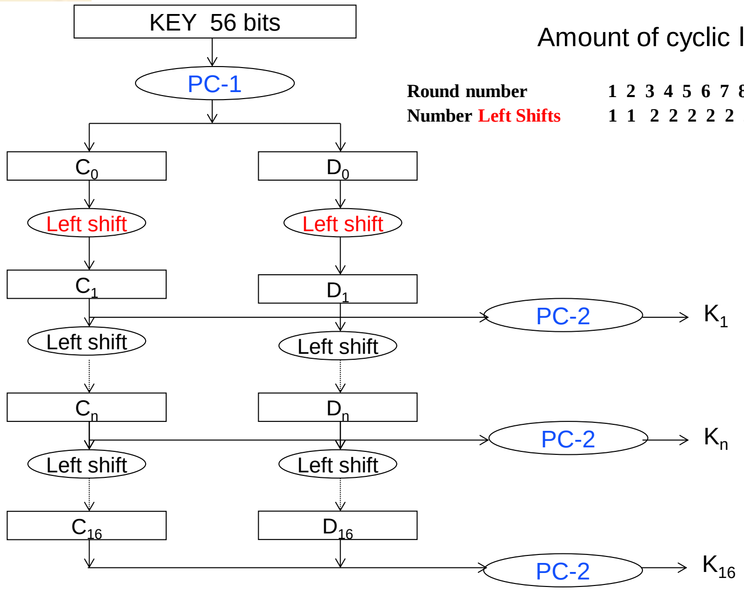
Design principles are not published (removed by NSA)

Claims

- Non-linear binary representation of the permutation has the highest number of terms [Meyer & Matyas, Cryptography, 1982]
- Each permutation consists of a long cycle, plus some short cycles (why? Uniform for random permutation)
- Change of each input will cause at least 2 bits of output to change (to against differential attacks?) [Coppersmith, 1993]

21

DES key schedule



Amount of cyclic left-shift

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number Left Shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

22

DES key schedule

PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

PC-1: select 56 bits from 64 bits key (with parity)

- 1-st bit of C_0D_0 is the 57th bit of *master key,*
- 2-nd bit is the 49th bit of *master key* and so on,
- the last, the 56th bit of C_0D_0 being the 4th bit of *master key*

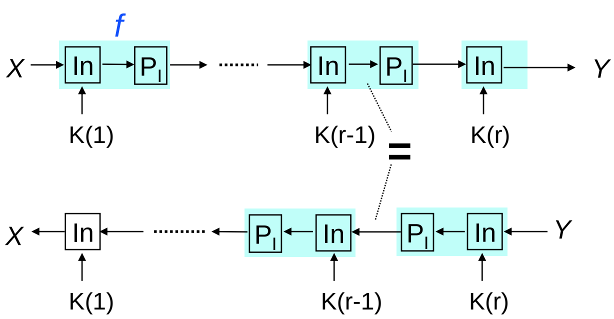
PC-2 : select 48 bits from CnDn

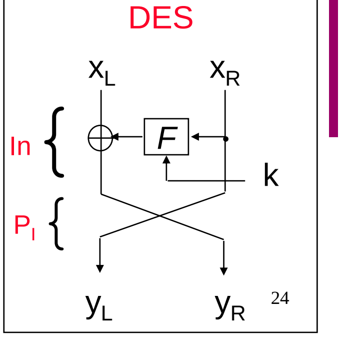
the first bit of K_n is the 14th bit of C_nD_n,
the second bit is the 17th bit of C_nD_n,
and so on,
the last, the 48th bit of K_n being the 32th bit of C_nD_n

23

DES-E/D similarity

- Same process for encryption and decryption, only subkeys are different
- round function: $Y = P_I[\, In(X,K)\,]$,
- Fix k, $In(In(x,k),k)=x$; $P_I^2 =$ Identity





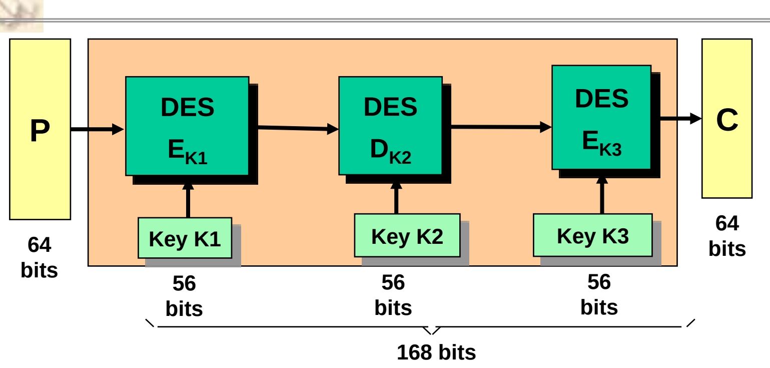
24

DES –key length

- DES key-length of k = 56 bits is too short [Diffie-Hellman 77]
- A design for a DES key search machine [Wiener 96] for $1M can find key in 3.5 hours, on average.
- 1997 version of this machine would be capable of finding DES keys in 35 minutes, on average.
- A $10,000 version would be capable of finding DES keys in 2.5 days.
- distributed Internet computing project DESCHALL[97], key search was done in running in idle time. 3 months with 10,000 computers
- A solution is multiple encryption

25

☆ Triple–DES (3DES)



- Triple DES $C=E_{K3}(D_{K2}(E_{K1}(P)))$
- Two-key Triple DES $C=E_{K1}(D_{K2}(E_{K1}(P)))$
- Backwards compatible: single DES for k1=k2=k3
- FIPS 46-3, 1999 October 25, DES-EDE3
  - NIST no longer support the use of single DES for many applications

Meet-in-the-middle attack

- Double DES

$$C=E_{K2}(E_{K1}(P)))$$

Meet-in-the-middle attack (known plaintext):

From known P, compute 2^{56} values of $E_{K1}(P)$ for all choices of k_1

From given C, compute 2^{56} values of $D_{K2}(C)$ for all choices of k_2

then $D_{K2}(C) = E_{K1}(P)$ holds for the correct k1 and k2.

$$P \longrightarrow \underset{k1}{\boxed{E}} \longrightarrow \; = \; \longleftarrow \underset{k2}{\boxed{D}} \longleftarrow C$$

Complexity: time 2^{56}, memory 2^{56}

Key size is 112-bit, but real strength is still 56-bit

How about 3DES, 4DES, 5DES,…?

27

Attacks on DES

- Differential Cryptanalysis [Biham-Shamir C91, Murphy 90, NSA 70s?]
  - Complexity:
    - Data: 2^{47} chosen texts, Process: 2^{37}
- Linear analysis [Matsui EC93]
  - Linear approximations with prob $p \neq ½$

    $P[i1,..,ia] \oplus C[j1,..,jb] = K[k1,..,kc]$, i_a, j_b, k_c are bit locations in P,C,K
  - Complexity:
    - data: 2^{43} known texts, Process.: 2^{43}
- Key search
  - Special hardware (1M$): 1 hour
  - Computers: 3 months with 10,000 computers

28

Weak keys

- DES Weak keys ($E_k=D_k$, E_k^2=Identity)
  - 0000000 0000000
  - 0000000 FFFFFFF
  - FFFFFFF 0000000
  - FFFFFFF FFFFFFF
- 16 semi-weak keys $E_{k1}(.) = D_{k2}(.)$:

```
01FE01FE01FE01FE and FE01FE01FE01FE01
1FE01FE00EF10EF1 and E01FE01FF10EF10E
01E001E001F101F1 and E001E001F101F101
1FFE1FFE0EFE0EFE and FE1FFE1FFE0EFE0E
011F011F010E010E and 1F011F010E010E01
E0FEE0FEF1FEF1FE and FEE0FEE0FEF1FEF1
```

- 48 keys which produce only 4 distinct subkeys (instead of 16)

29

Exercise 4 – DES

1. Let M’ be the bitwise inversion of M. Prove that, for DES, if $Y=E_k(X)$, then $Y'=E_{k'}(X')$

- Hint: $(A \oplus B)'=A' \oplus B$.

2. prove that for DES, if key k is all-0, then $E_k=D_k$. Can you find a method to avoid this problem?

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