Network Information Security

Lecture five: block cipher

Rachel yuan
2019.10

1

填空题 4分 设置

1. 古典加密方法包括 [填空1]和 [填空2] 。
2. OTP的全称是[填空3]。
3. OTP具备香农提出的 [填空4] 性质，它能够完美的抵御唯密文攻击。

正常使用填空题需3.0以上版本雨课堂

作答

2

多选题 2分 设置

Drawbacks of OTP are ( )。

A **Truly Random**

B **As long as the message**

C **Secure exchange of OTP**

D **One time only use of the key**

提交

3

主观题 7分 设置

请说出OTP与Stream cipher（流密码）的不同。

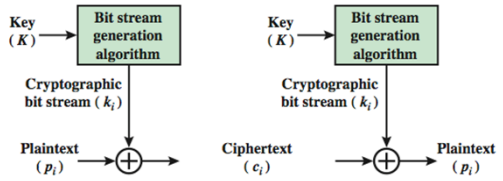
正常使用主观题需2.0以上版本雨课堂

作答

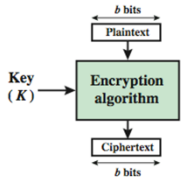
4

Stream cipher vs. block cipher

5



(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

6

How to build a *block cipher*?

7

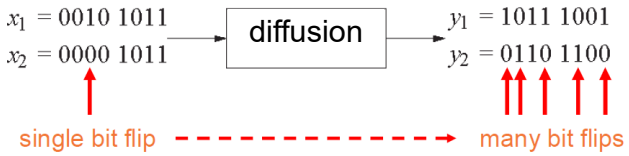
Two primitive operations

- Claude shannon
- Confusion(混淆): 密文与明文之间的关系十分复杂，无法从数学上去描述，或从统计上去分析
  - S-box
- Diffusion(扩散): 明文中的每一位二元数字都对密文中的多个二元数字有直接影响

8

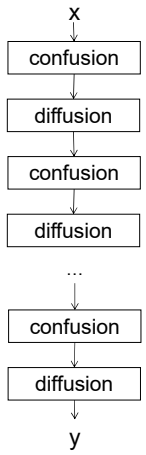
Diffusion

- Example:



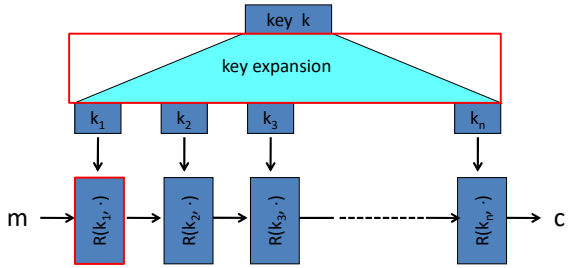
9

Product cipher



10

Block Ciphers Built by Iteration (迭代)



R(k,m) is called a round function

for DES (n=16), for 3DES (n=48), for AES-128 (n=10)

Dan Boneh

11

The Data Encryption Standard (DES)

- Early 1970s: Horst Feistel designs Lucifer at IBM
  key-len = 128 bits ; block-len = 128 bits
- 1973: NBS asks for block cipher proposals.
  IBM submits variant of Lucifer.
- 1976: NBS adopts DES as a federal standard
  key-len = 56 bits ; block-len = 64 bits
- 1997: DES broken by exhaustive search
- 2000: NIST adopts Rijndael as AES to replace DES

Widely deployed in banking and commerce

NBS: national bureau of standard
NIST: *National Institute of Standards and Technology*

12

Feistel Ciphers Overview

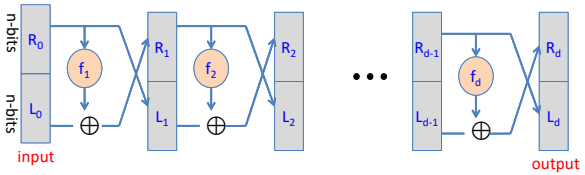
- Feistel cipher has been a very influential "template" for designing a block cipher.
- Major benefit: can do encryption and decryption with the same hardware
- Examples: DES, RC5

13

DES: core idea – Feistel Network

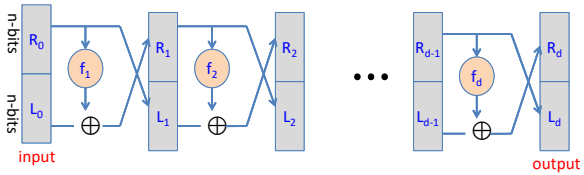
Given functions $f_1, \ldots, f_d$: $\{0,1\}^n \rightarrow \{0,1\}^n$

Goal: build invertible function $F: \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$



In symbols: $R_i = f_i(R_{i-1}) \oplus L_{i-1}$, $L_i = R_{i-1}$

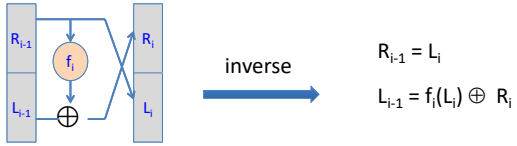
14



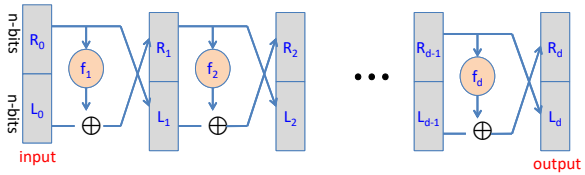
Claim: for all $f_1, \ldots, f_d$: $\{0,1\}^n \rightarrow \{0,1\}^n$

Feistel network $F: \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ is invertible(可逆的)

Proof: construct inverse



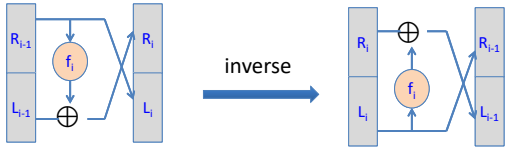
15



Claim: for all $f_1, \ldots, f_d$: $\{0,1\}^n \rightarrow \{0,1\}^n$

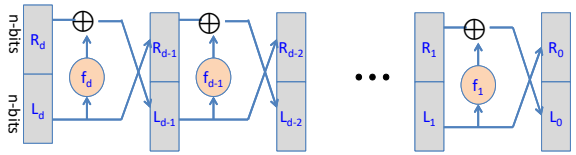
Feistel network $F: \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ is invertible(可逆的)

Proof: construct inverse



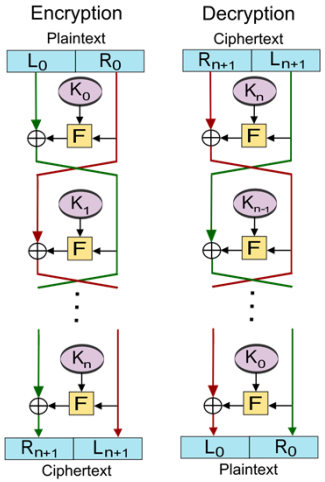
16

Decryption circuit



- Inversion is basically the same circuit, with $f_1, \ldots, f_d$ applied in reverse order
- General method for building invertible functions (block ciphers) from arbitrary functions.
- Used in many block ciphers … but not AES

17

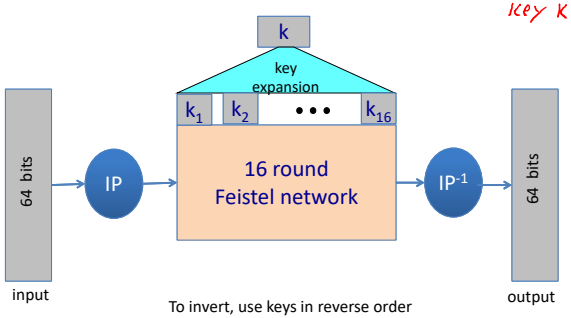


18

DES: 16 round Feistel network

$f_1, \ldots, f_{16}$: $\{0,1\}^{32} \rightarrow \{0,1\}^{32}$, $f_i(x) = F(k_i, x)$

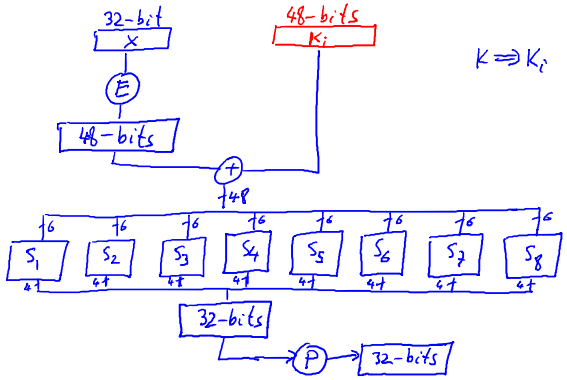
From key K



To invert, use keys in reverse order

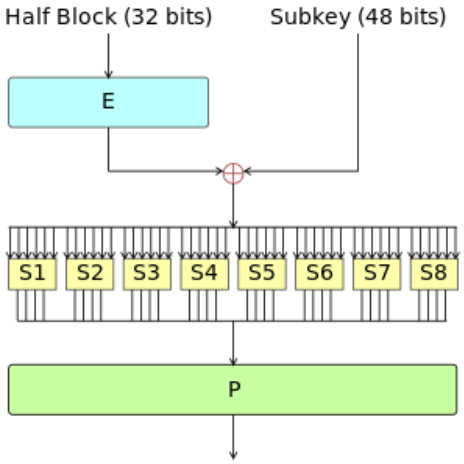
19

The function F(ki, x)



S-box: function $\{0,1\}^6 \rightarrow \{0,1\}^4$, implemented as look-up table.

20



21

The S-boxes

$S_i: \{0,1\}^6 \rightarrow \{0,1\}^4$

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

22

Example: a bad S-box choice

Suppose:

$S_i(x_1, x_2, \ldots, x_6) = (x_2 \oplus x_3,\ \ x_1 \oplus x_4 \oplus x_5,\ \ x_1 \oplus x_6,\ \ x_2 \oplus x_3 \oplus x_6)$

or written equivalently: $S_i(\mathbf{x}) = A_i \cdot \mathbf{x} \pmod 2$

We say that S_i is a linear function.

$$\begin{pmatrix} 0&1&1&0&0&0 \\ 1&0&0&1&1&0 \\ 1&0&0&0&0&1 \\ 0&1&1&0&0&1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} x_2 \oplus x_3 \\ x_1 \oplus x_4 \oplus x_5 \\ x_1 \oplus x_6 \\ x_2 \oplus x_3 \oplus x_6 \end{pmatrix}$$

23

Example: a bad S-box choice

Then entire DES cipher would be linear: ∃fixed binary matrix B s.t.

$$DES(k,m) = \underset{64}{\overset{832}{B}} \cdot \begin{pmatrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{pmatrix} = c \pmod 2$$

24

Choosing the S-boxes

Choosing the S-boxes at random would result in an insecure block cipher (key recovery after $\approx 2^{24}$ outputs) [BS'89]

Several rules used in choice of S boxes:

- No output bit should be close to a linear func. of the input bits
- S-boxes are 4-to-1 maps

⋮

25

Strengthening DES against exhaustive attack

Method 1: **Triple-DES**

•Let $E : K \times M \longrightarrow M$ be a block cipher

•Define **3E**: $K^3 \times M \longrightarrow M$ as

$$\mathbf{3E}((k_1,k_2,k_3), m) = \mathbf{E}(k_1, \mathbf{D}(k_2, \mathbf{E}(k_3, m)))$$

$k_1 = k_2 = k_3 \rightarrow$ Single DES

For 3DES: key-size = $3 \times 56 = 168$ bits. $3\times$ slower than DES.

(simple attack in time $\approx 2^{118}$)

26

The AES block cipher

27

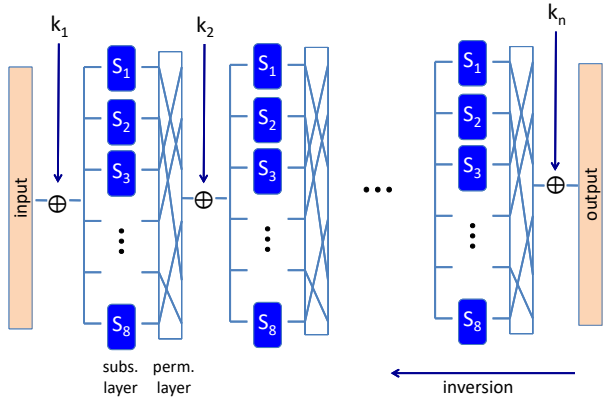
The AES process

- 1997: NIST publishes request for proposal
- 1998: 15 submissions. Five claimed attacks.
- 1999: NIST chooses 5 finalists
- 2000: NIST chooses Rijndael as AES (designed in Belgium)

Key sizes: 128, 192, 256 bits. Block size: 128 bits

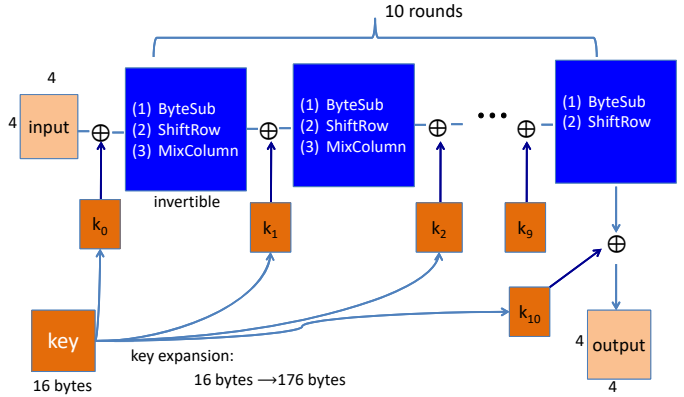
28

AES is a Subs-Perm network (not Feistel)



29

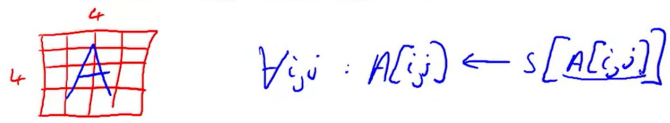
AES-128 schematic



30

The round function

- **ByteSub**: a 1 byte S-box. 256 byte table (easily computable)



31

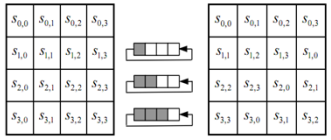
S-box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

32

The round function

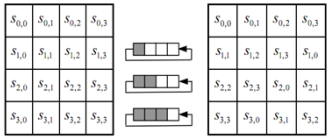
- **ByteSub**: a 1 byte S-box. 256 byte table (easily computable)
- **ShiftRows**:



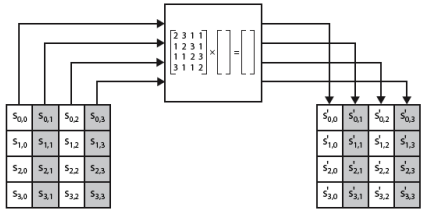
33

The round function

- **ByteSub**: a 1 byte S-box. 256 byte table (easily computable)
- **ShiftRows**:



- **MixColumns**:



34