第 2 章：分组密码体制

2.2 数据加密标准（DES）

赵俊舟

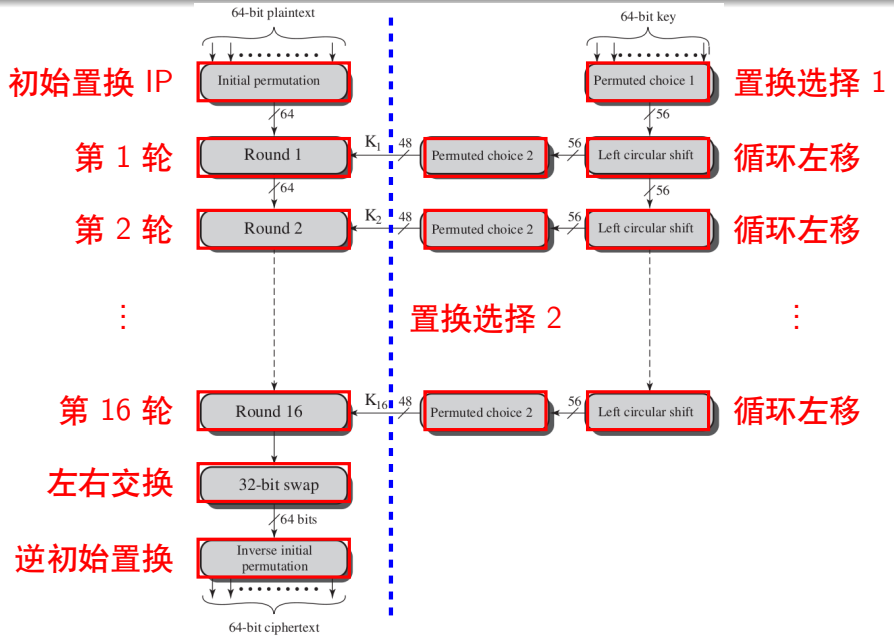
junzhou.zhao@xjtu.edu.cn

2025 年 2 月 28 日

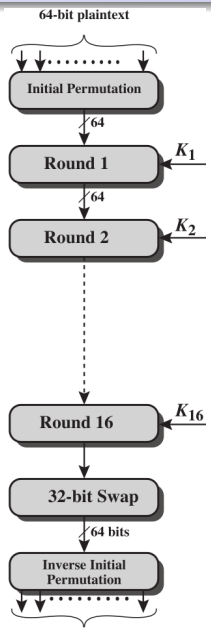
DES 的历史

- IBM 公司在 1971 年由 Horst Feistel 领导开发了 Lucifer Cipher，使用 128 位密钥加密 64 位的分组。
- 1974 年，IBM 与 NSA 合作开发了 Lucifer 的一个修订版，易于在芯片上实现，抗密码分析能力更强，且密钥缩短为 56 位。
- 1977 – 1998 期间，这个加密方案成为美国国家密码标准，称为 DES。
- DES 在密码学领域被深入分析。
- 目前 DES 已经不安全，已经被淘汰，替代算法包括 3DES、AES 等。

DES 加密过程


64-bit plaintext
64-bit key
初始置换 IP
Initial permutation
Permuted choice 1
置换选择 1
64
56
第 1 轮
Round 1
K_1
48
Permuted choice 2
56
Left circular shift
循环左移
64
56
第 2 轮
Round 2
K_2
48
Permuted choice 2
56
Left circular shift
循环左移
置换选择 2
第 16 轮
Round 16
K_{16}
48
Permuted choice 2
56
Left circular shift
循环左移
左右交换
32-bit swap
64 bits
逆初始置换
Inverse initial permutation
64-bit ciphertext

DES 加密过程



- DES 的明文长 64 位，密钥长 56 位（虽然输入 64 位密钥，但内部仅使用了 56 位）；
- 明文处理经过三个阶段：
  - 首先 64 位明文经过初始置换（IP）而被重新排列；
  - 然后进行 16 轮相同函数的作用，每轮都进行代替和置换；
  - 最后一轮输出 64 位分组，左右互换产生预输出，经过逆初始置换（IP^{-1}）产生 64 位密文。
- 除了初始和末尾的置换操作，DES 的结构与 Feistel 密码结构完全相同。

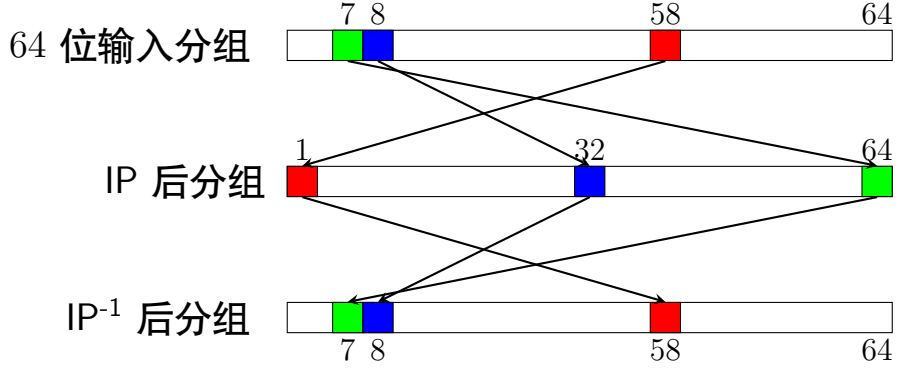
初始置换 IP 和逆初始置换 IP^{-1}

(a) Initial Permutation (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

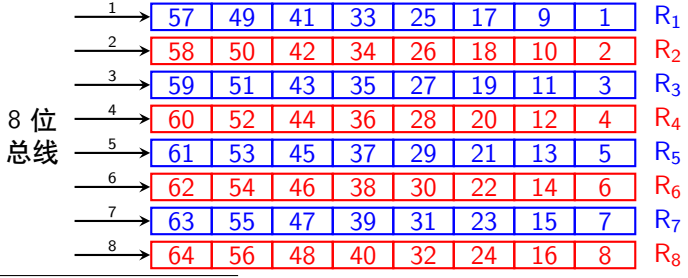
(b) Inverse Initial Permutation (IP^{-1})

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



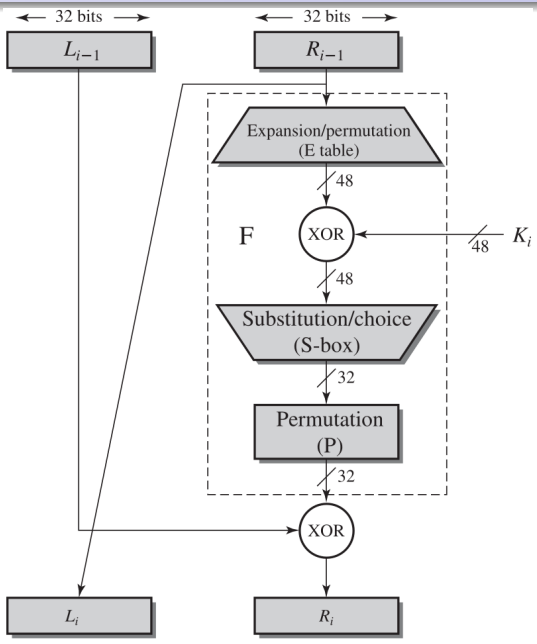
初始置换 IP 和逆初始置换 IP^{-1}

- 初始置换和逆置换并不能增强 DES 的安全性，而是为了方便硬件电路实现[1]。
- 当总线宽度为 8 位时，需要配合使用 8 个 8 位移位寄存器，经过 8 个时钟得到 64 位输入分组。
- 如果不使用初始置换，那么从 8 个寄存器取前 32 位和后 32 位时，会在电路连线中产生很多交叉。



[1] https://crypto.stackexchange.com/a/6

每一轮的运算



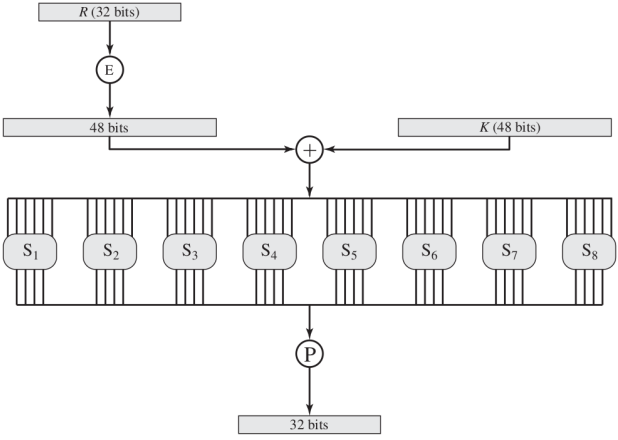
- 第 i 轮的输入分成左右两部分 L_{i-1} 和 R_{i-1}；
- 做如下运算得到本轮输出的左右两部分 L_i 和 R_i：

$$L_i = R_{i-1}$$

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

其中 $\oplus$ 表示异或运算。

轮函数 F

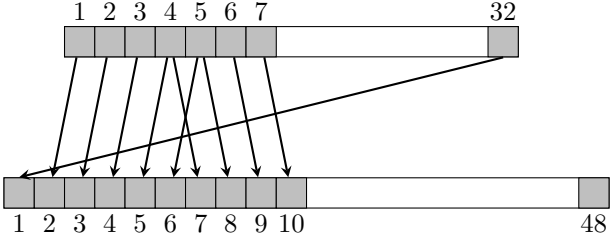


轮函数 F 是 DES 的核心运算函数，包含 4 步运算：扩展置换函数 E、与子密钥异或、S 盒替换和置换函数 P。

扩展置换函数 E

使用置换表 E 将 32 位 R 扩展成 48 位，起扩散作用。

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



S 盒替换

- 48 位结果送给 8 个替换盒 $S_1, \ldots, S_8$，得到 32 位结果；

S_1

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

S_2

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

S_3

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

S_4

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

S 盒替换

S_5

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

S_6

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

S_7

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

S_8

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

S 盒替换

- S 盒是轮函数 F 的核心，每个 S 盒输入 6 位，输出 4 位。
- 作用是混淆，即通过 S 盒替换使密文和密钥之间的关系尽可能复杂。
- 每个 S 盒输入的第一位和最后一位组成一个 2 位二进制数用来选择 S 盒 4 行中的某一行，中间 4 位用来选择 16 列中的某一列。
- 行列对应的十进制数转换为二进制后可得到输出的 4 位二进制数。

例

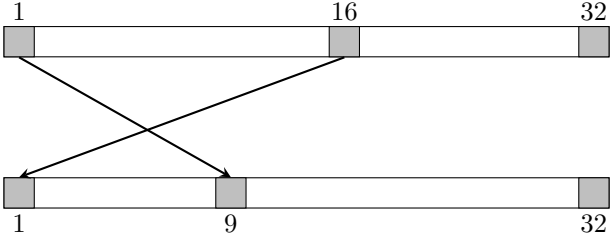
例如，在 S_1 中，若输入为 011001，则行是 $1(01)$，列是 $12(1100)$，该处的值为 9，所以输出 1001。

置换函数 P

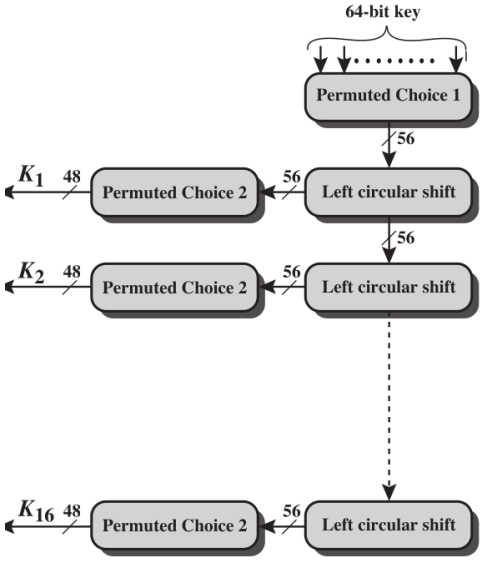
- 最后使用 32 位置换表 P，把 32 位结果再进行一次置换处理。

(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

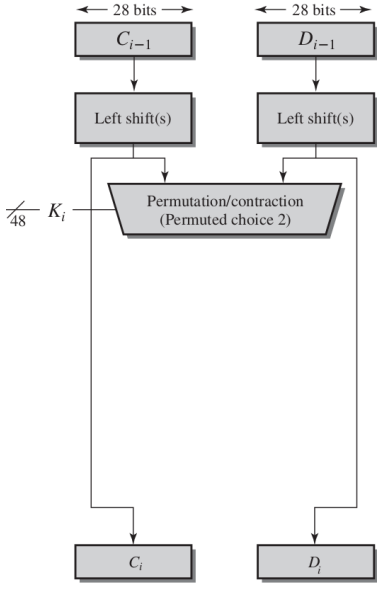


DES 子密钥生成过程



- 密钥经过一个置换，然后循环左移，再经过另一个置换，得到各轮的子密钥 K_i；
- 每轮的置换函数都一样，由于循环左移，使得各轮子密钥各不相同。

子密钥生成算法



- 每一轮都要依据输入密钥生成一个子密钥以供加密使用；
- 输入密钥为 64 位，DES 只使用其中 56 位，其余位可用于奇偶校验；
- 使用置换选择 1（PC-1），将 56 位密钥分成两半 C 和 D，每部分 28 位；
- 根据循环左移表将这两半分别循环左移 1 位或 2 位；
- 使用置换选择 2（PC-2），形成 48 位子密钥，用在轮函数 F 中。

DES 只使用输入密钥中的 56 位

(a) Input Key

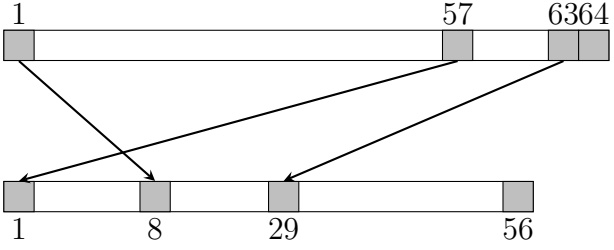
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

- 为了与输入明文分组长度一致，DES 密钥长度为 64 位，实际只使用 56 位，每个字节的最后一位在 DES 算法中没有使用，可用于校验等其他功能。

置换选择 1

(b) Permuted Choice One (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



循环左移表

(d) Schedule of Left Shifts

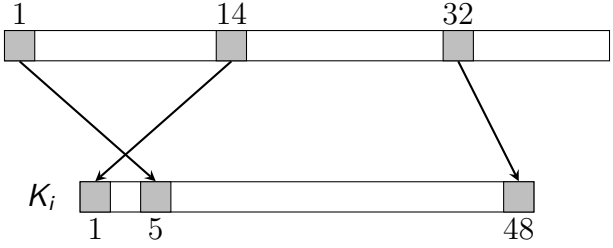
Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

每轮循环左移 $1 \sim 2$ 位。

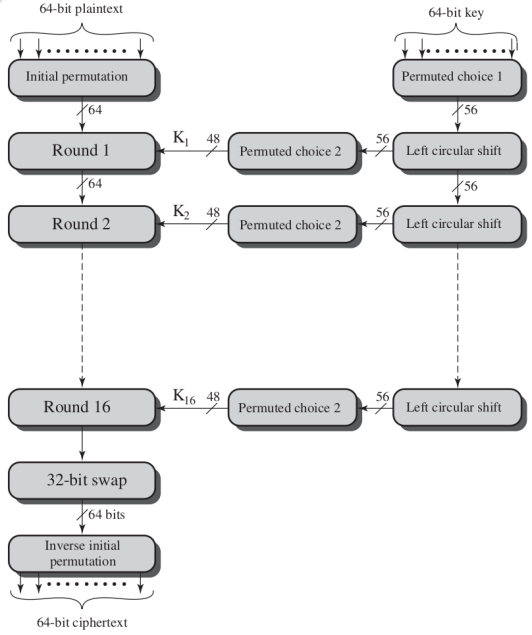
置换选择 2

(c) Permuted Choice Two (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



DES 解密



- 同 Feistel 密码，DES 解密使用与加密相同的算法，只是子密钥的使用顺序相反。

DES 举例

Plaintext:	`02468aceeca86420`
Key:	`0f1571c947d9e859`
Ciphertext:	`da02ce3a89ecac3b`

Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

Note: DES subkeys are shown as eight 6-bit values in hex format

雪崩效应

- 雪崩效应（Avalanche Effect）：明文或密钥的一比特的变化，引起密文许多比特的改变。如果变化太小，就可能找到一种方法减小有待搜索的明文和密文空间的大小。
- 如果用同样密钥加密只差一比特的两个明文：
  0000000000000000.....00000000
  1000000000000000.....00000000
  3 次循环以后密文有 21 个比特不同，16 次循环后有 34 个比特不同。
- 如果用只差一比特的两个密钥加密同样明文：
  3 次循环以后密文有 14 个比特不同，16 次循环后有 35 个比特不同。

DES 的雪崩效应：改变明文

使用相同密钥加密两个只差 1 比特的明文：

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

DES 的雪崩效应：改变密钥

使用相差 1 比特的两个密钥（0f1571c947d9e859 与 1f1571c947d9e859）加密相同明文：

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30