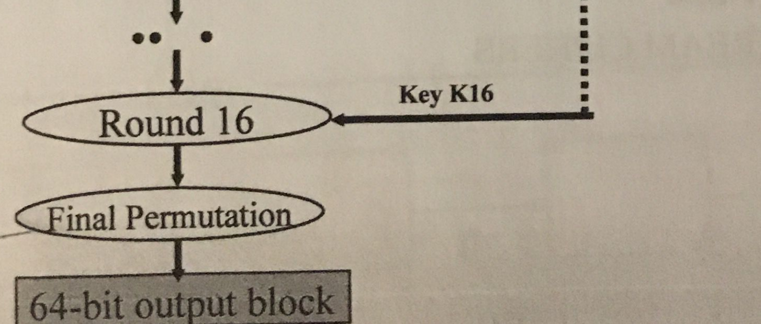
Data Encryption Standard (DES)

- Example of a block cipher (private key cryptosystem).
- Overview
  - Initial permutation
  - 16 rounds of processing
  - Final permutation

  (on each 64-bit input block)
- Permutations
- Key generation
- Single DES round

Overall DES structure



Both Alice & Bob use 56-bit secret key to generate 16 48-bit keys to be used in 16 rounds.

Permutations are used to increase the computation time

Permutations

- Initial and final permutations are not derived from keys and thus do not add to security. Just makes DES less efficient to implement in software.
- Initial and final permutations are inverses of each other. If they were put together, they cancel each other. In this case, they are separated by 16 rounds.

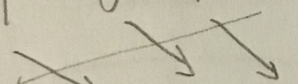
Note:

1. A permutation P has an inverse P^{-1} such that applying P^{-1} to a bit string cancels the effect applying P. In other words, applying P and P^{-1} in sequence to a bit string has no effect on the bit string.
2. A permutation is usually represented by a one-dimensional matrix. The values in the matrix represent the shifted bit positions.

Examples:

bit position: 1 2 3 4

① 1 0 1 1



1 1 0 1

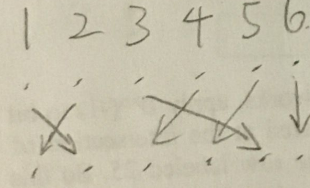
1 2 3 4

$P_1 = (2, 3, 4, 1)$

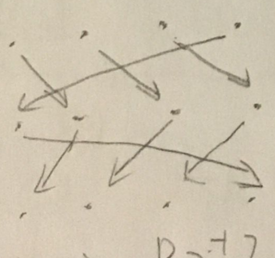
Bit #1 has moved to position #2

#2 - - - - #3

② 1 2 3 4 5 6 $P_2 = (2, 1, 5, 3, 4, 6)$



③ What is P_1^{-1}?



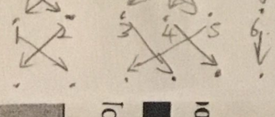
1 2 3 4

$P_1 = (2, 3, 4, 1)$

$P_1^{-1} = (4, 1, 2, 3)$

④ What is P_2^{-1}? $P_2 = (2, 1, 5, 3, 4, 6)$

$P_2^{-1} = (2, 1, 4, 5, 3, 6)$



In DES, the representation is slightly different. The permutation is represented by a 2-d matrix, in which the permuted position can be obtained by the summing the intersecting values of the top row and the first column.

IP: Initial Permutation

Bit	0	1	2	3	4	5	6	7
1	58	50	42	34	26	18	10	2
9	60	52	44	36	28	20	12	4
17	62	54	46	38	30	22	14	6
25	64	56	48	40	32	24	16	8
33	57	49	41	33	25	17	9	1
41	59	51	43	35	27	19	11	3
49	61	53	45	37	29	21	13	5
57	63	55	47	39	31	23	15	7

How to read the permutation:
For example, let's examine how bit 32 is transformed under IP. In the table, bit 32 is located at the intersection of the column labeled 4 and the row labeled 25. So this bit becomes bit 29 of the 64-bit block after the permutation.

Bit #32 ⇒ Bit # (25+4) (row, column) or Bit #29

Bit #51 → Bit #42

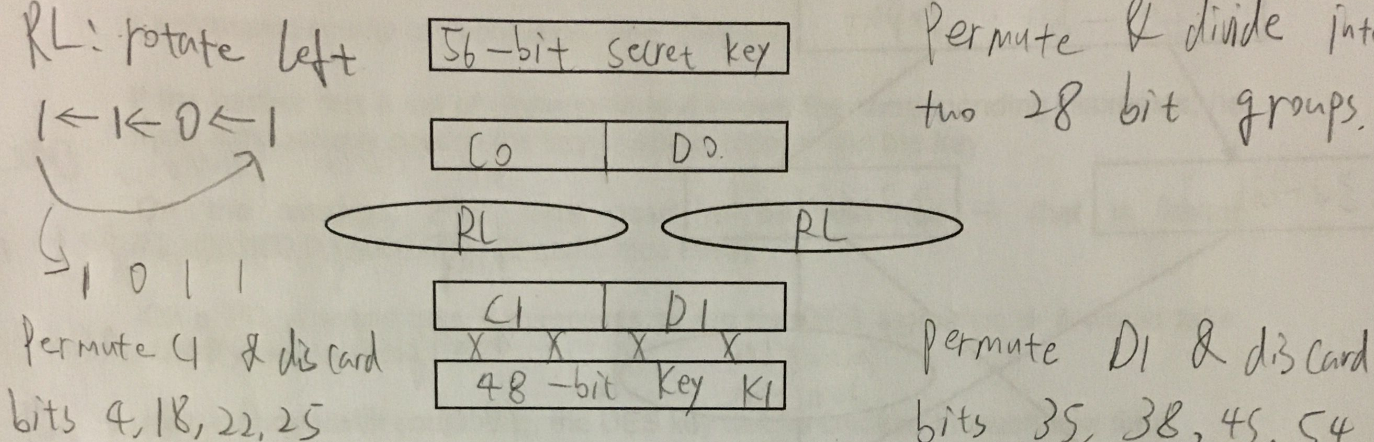
IP^(-1): Inverse Initial Permutation

Bit	0	1	2	3	4	5	6	7
1	40	8	48	16	56	24	64	32
9	39	7	47	15	55	23	63	31
17	38	6	46	14	54	22	62	30
25	37	5	45	13	53	21	61	29
33	36	4	44	12	52	20	60	28
41	35	3	43	11	51	19	59	27
49	34	2	42	10	50	18	58	26
57	33	1	41	9	49	17	57	25

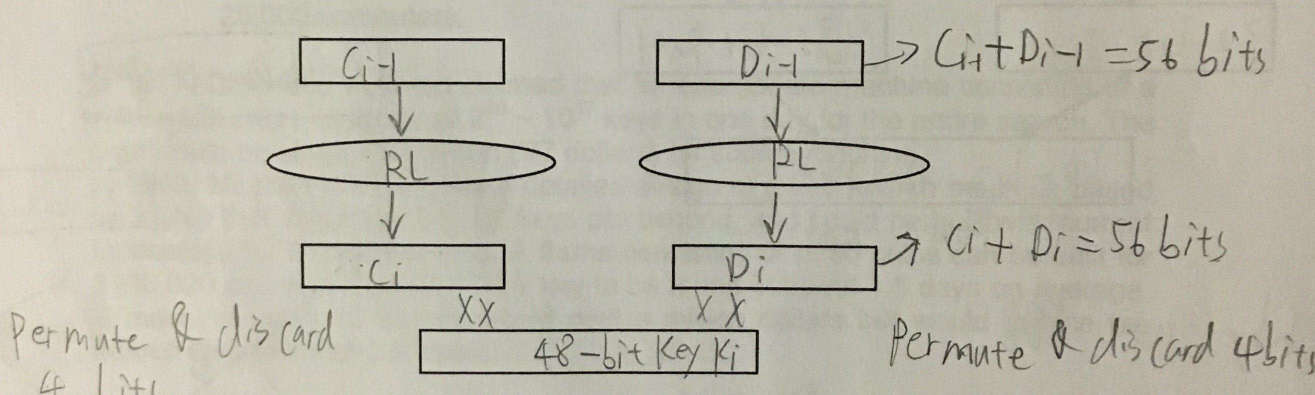
Bit #29 ⇒ Bit #32 (25+7) (row, column)

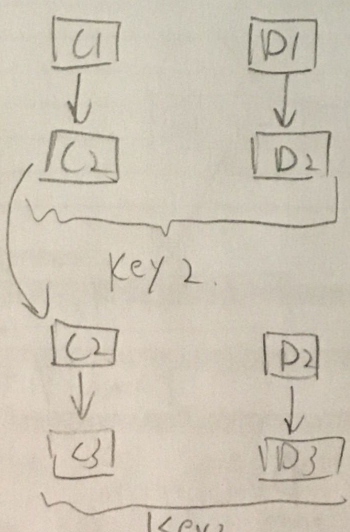
To see how the inverse works, apply IP^(-1) to bit 29. In IP^(-1), bit 29 is located at the intersection of the column labeled 7 and the row labeled 25. So this bit becomes bit 32 after the permutation. And this is the bit position that we started with before the first permutation.

Generation of keys from initial key
Generation of K1:

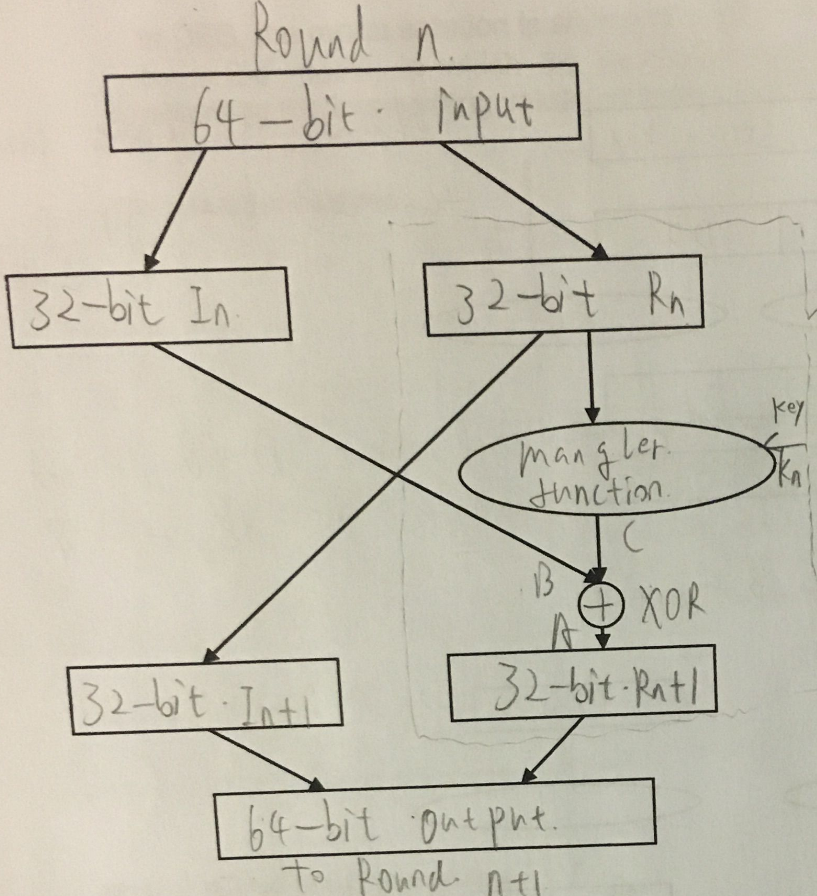


Generation of Ki: where i=2, 3, 4......16.





Single DES round



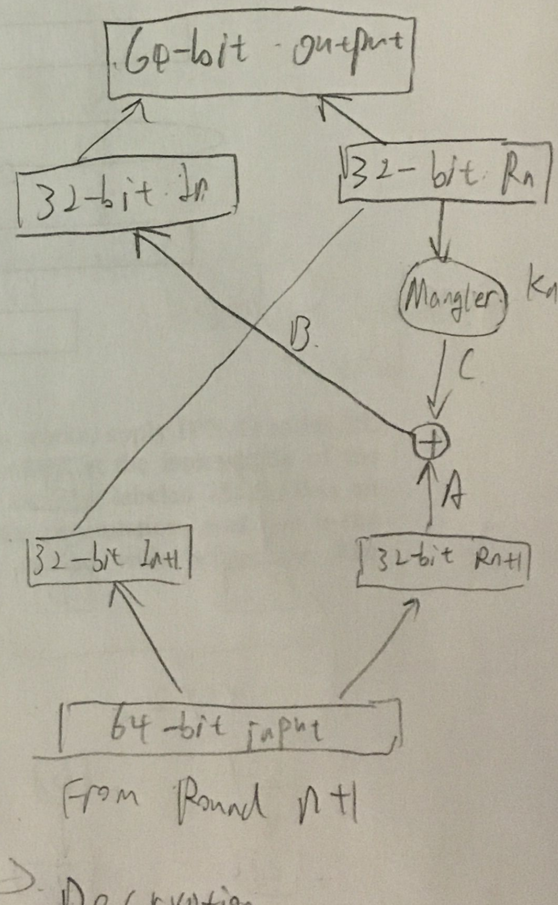
In each round, one half of the input is mangled, In the next round the other half will be mangled.

Why use XOR?

If $A = B \oplus C$ then $B = A \oplus C$

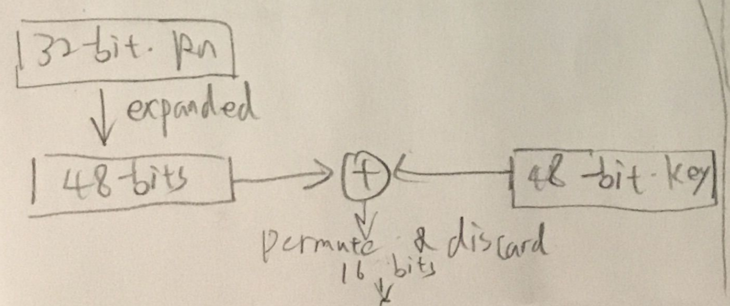
So, it can also be used in decryption

Encryption



Decryption

Mangler



CRACKING THE DES KEY

There is no mathematical proof that DES is secure.

It achieves security by “confusion” and “diffusion”.

If the hacker has a set of ciphertexts and knows the corresponding plaintexts, he must exhaustively search the key space in order to get the key.

On the average, $2^{56}/2$ keys need to be searched → that is about 72,000,000,000,000,000 (72 quadrillion keys).

On a PC, it would take 4 microsecs to run the DES algorithm → it would take 4,500 years to break DES.

But with a parallel computing, the DES key can be cracked in much less time:

- A field programmable gate array was built to crack 100 million keys/ sec.
- A distributed network was employed to crack 70 billion keys/sec with 20,000 computers.

In 1977, Diffie and Hellman claimed that an appropriate machine consisting of a million LSI chips could try all $2^{56} \sim 10^{17}$ keys in one day for the entire search. The cost would be about $20 million ('77 dollars) for such a machine.
In 1993, Michael Wiener gave a detailed design of a key search machine based on a chip that could test 5×10^{7} keys per second, and could be built with current technology for $10.50 per chip. A frame consisting of 5760 chips can be built for $100,000 and would allow a DES key to be found in about 1.5 days on average. A machine using 10 frames would cost a million dollars but would reduce the search time to about 3.5 hours.

DES Key Cracking Record (DES Challenge put out by RSA)

Year	Time to crack 56-bit DES key
1997	4 months
1998	39 days
1998	56 hours
1999	22 hours

References:
www.distributed.net/des/release-desiii.txt

www.cryptography.com/resources/whitepapers/DES.html

http://lasecwww.epfl.ch/memo/memo_des.shtml

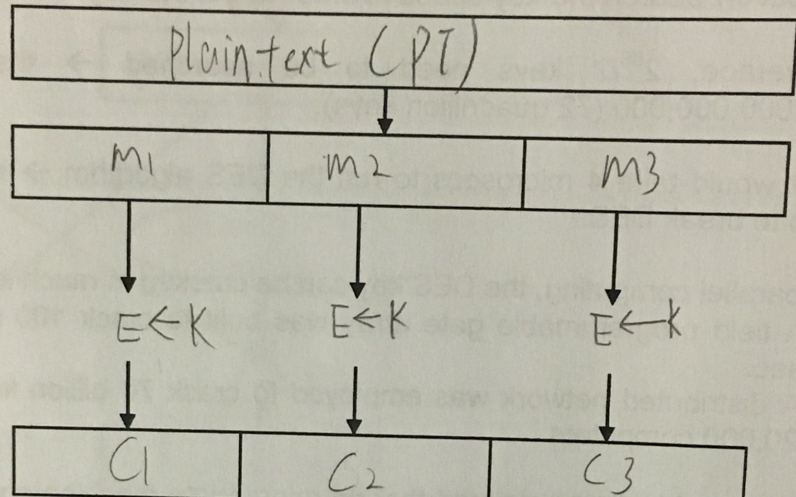
It's the Key Size that Matters!
Minimum Key Size must be 128 bits.

BLOCK CIPHER TECHNIQUES (cont'd.)

Encryption of multiple blocks

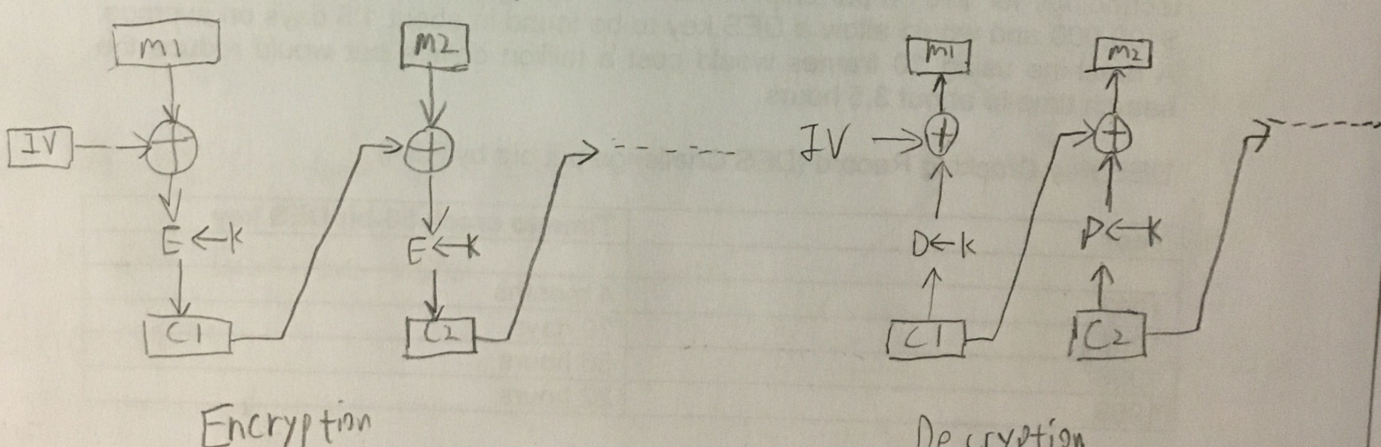
Electronic Code Book (ECB) Method

E←k
encrypt with
Key k.



Drawback: If $m_i = m_j$, then $C_i = C_j$
This can aid the hacker in reverse engineering.

Cipher Block Chaining (CBC) Method



Encryption

Decryption

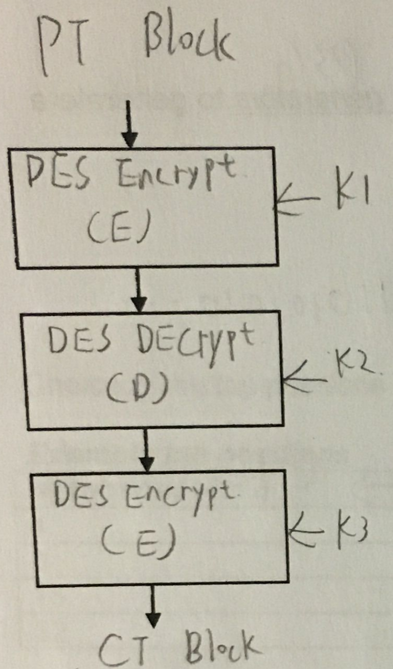
Note: The IV is sent encrypted in the first packet by the sender.

Drawback of CBC: sequential processing.

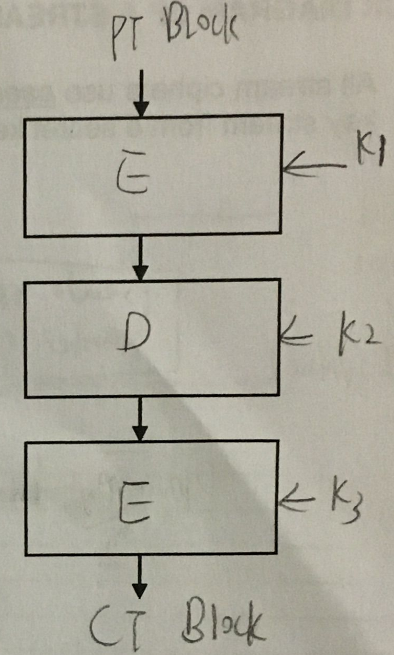
IV: Initialization Vector
(Random binary number chosen by the sender)

3-DES

3-key 3-DES



2-key 3-DES



Why E-D-E and not E-E-E?

1. In E-E-E the permutations which do not depend on

$IP \ldots IP^{-1} IP \ldots IP^{-1} IP \ldots IP$

Keys can each other & thus reduce the brute force attack time.

In E-D-E no cancellation

$IP \ldots IP^{-1} IP^{-1} \ldots IP\, IP \ldots IP^{-1}$

2. Interoperability : A 3-DES machine can communicate with a 2-DES machine by setting $K_1 = K_2 = K_3 = K$

互用性