Modern Block Ciphers and Data Encryption Standard (DES)

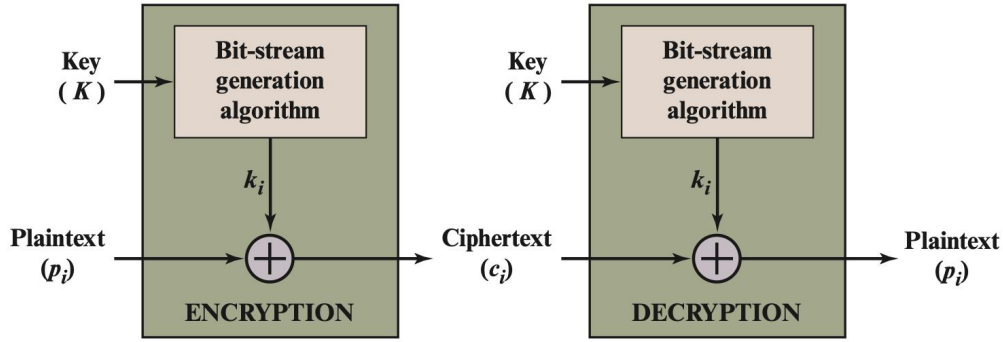
Modern Block Ciphers

- now look at modern block ciphers
- one of the most widely used types of cryptographic algorithms
- provide secrecy /authentication services
- focus on DES (Data Encryption Standard)
- to illustrate block cipher design principles

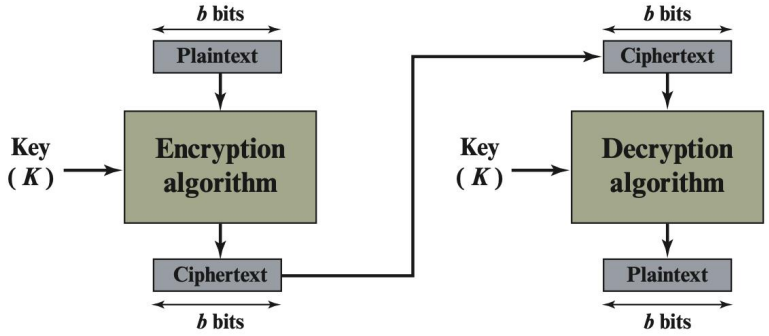
Block vs Stream Ciphers

Feature	Block Cipher	Stream Cipher
Data Processing	Encrypts data in fixed-size blocks (e.g., 128 bits).	Encrypts data bit-by-bit or byte-by-byte.
Speed	Generally slower than stream ciphers (though modern block ciphers like AES are highly optimized).	Generally faster due to simpler operations (mainly XOR).
Error Propagation	An error in one ciphertext bit usually affects the entire block upon decryption.	An error in one ciphertext bit usually affects only the corresponding plaintext bit(localized error).
Mechanisms	Uses Confusion and Diffusion via rounds of substitution and permutation.	Primarily uses Confusion through a highly non-linear keystream generator.
Required Use	Requires Modes of Operation (e.g., CBC, GCM) to handle long messages and ensure security.	Typically operates directly; a single key stream is generated from the key and a unique nonce(or IV).
Use Case	File storage, database encryption, VPNs.	Real-time voice/video, secure protocols (TLS/SSL), wireless communication.

Block vs Stream Ciphers



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations seen before:
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key

Confusion and Diffusion

- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as possible

Confusion: Hiding the Key

- The goal of Confusion is to make the relationship between the secret key and the resulting ciphertext (encrypted text) as complex and mixed-up as possible.
- If you change just one bit of the secret key, the entire resulting ciphertext should change almost completely. This prevents an attacker from being able to guess the key bit-by-bit.
- Confusion is achieved through Substitution, where one piece of data is replaced by another based on a lookup table called an S-Box (Substitution-Box).

Confusion: Hiding the Key

- Imagine you have a 4-bit input from the data being encrypted.

Input (Plaintext)	0101
S-Box Rule: (Lookup)	0101 always becomes 1100
Output (Ciphertext)	1100

Diffusion: Hiding the Plaintext Statistics

- The goal of Diffusion is to spread the influence of every single plaintext bit across the entire ciphertext.
- If you change just one bit of the original plaintext, that single change should cause about half of the bits in the resulting ciphertext to change. This hides statistical patterns, like how often certain letters appear.
- Diffusion is achieved through Permutation (or transposition), which simply rearranges the positions of the bits

Diffusion Example (P-Box)

- Imagine you have an 8-bit block of data that is the output of the confusion step: 1100 0011.

Initial Bit Position	1	2	3	4	5	6	7	8
Data	1	1	0	0	0	0	1	1
P-Box Rule: (Shuffle)	Swap bits 1 and 8, and 4 and 5.							
New Bit Position	8	2	3	5	4	6	7	1
Final Data	1	1	0	0	0	0	1	1

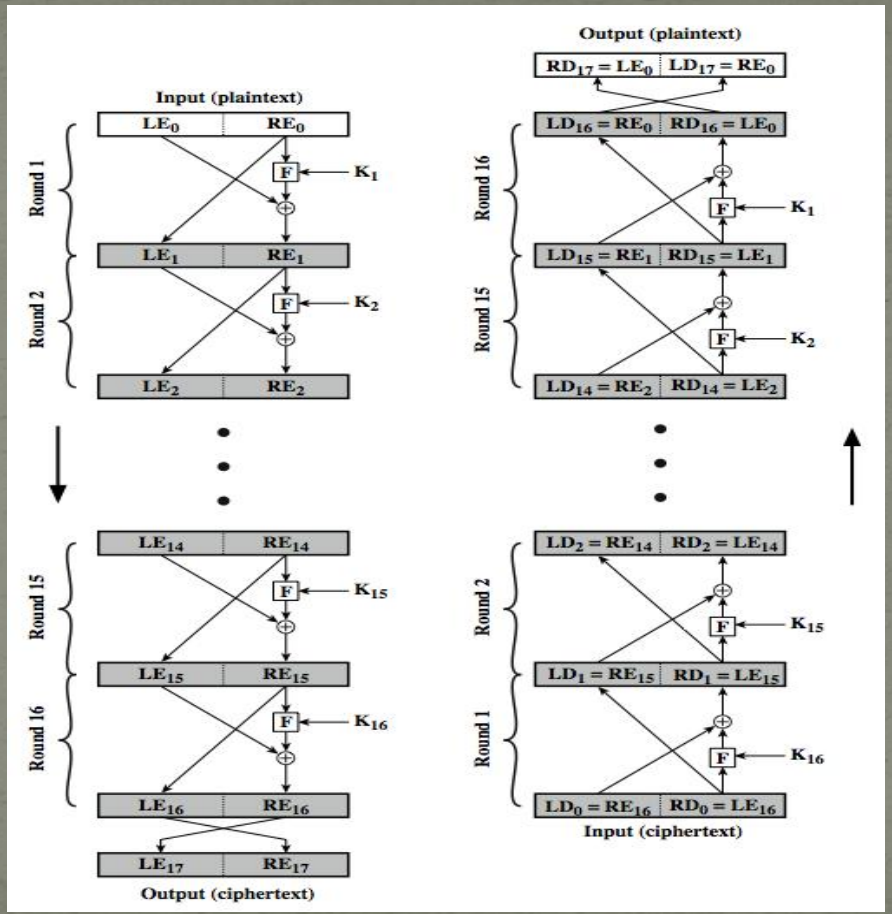
Substitution-Permutation Ciphers (SPNs)

- Shannon realized that using Substitution (Confusion) or Permutation (Diffusion) alone is weak. He proposed chaining these two simple, opposing operations together repeatedly.
- This "product cipher" structure is called a Substitution-Permutation Network (SPN).

SPN Example: The Layer Cake

- A Substitution-Permutation Cipher is like a multi-layered cake (each layer is a "round" of encryption).
- S-Layer (Confusion): You apply a layer of sweet, sticky frosting (S-Boxes) that changes the flavor of the data. This provides confusion.
- P-Layer (Diffusion): Then, you take a knife and rearrange the pieces of the cake (P-Boxes) so that every flavor is now spread across the entire plate. This provides diffusion.
- Repeat: You repeat this process many times (typically 10-14 times) to mix the key and the data so thoroughly that the final ciphertext is completely unrecognizable and impossible to break without the key.

Feistel Encryption and Decryption (16 rounds)



Feistel Cipher Design Elements

- block size: Larger block sizes mean greater security
- key size: Larger key size means greater security but may decrease encryption/ decryption speed.
- number of rounds: The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

Data Encryption Standard (DES)

- DES (Data Encryption Standard) is a symmetric-key block cipher.
- Developed by IBM in the 1970s and adopted by NIST as a federal standard in 1977.
- Operates on 64-bit blocks with a 56-bit key.

DES Key Structure

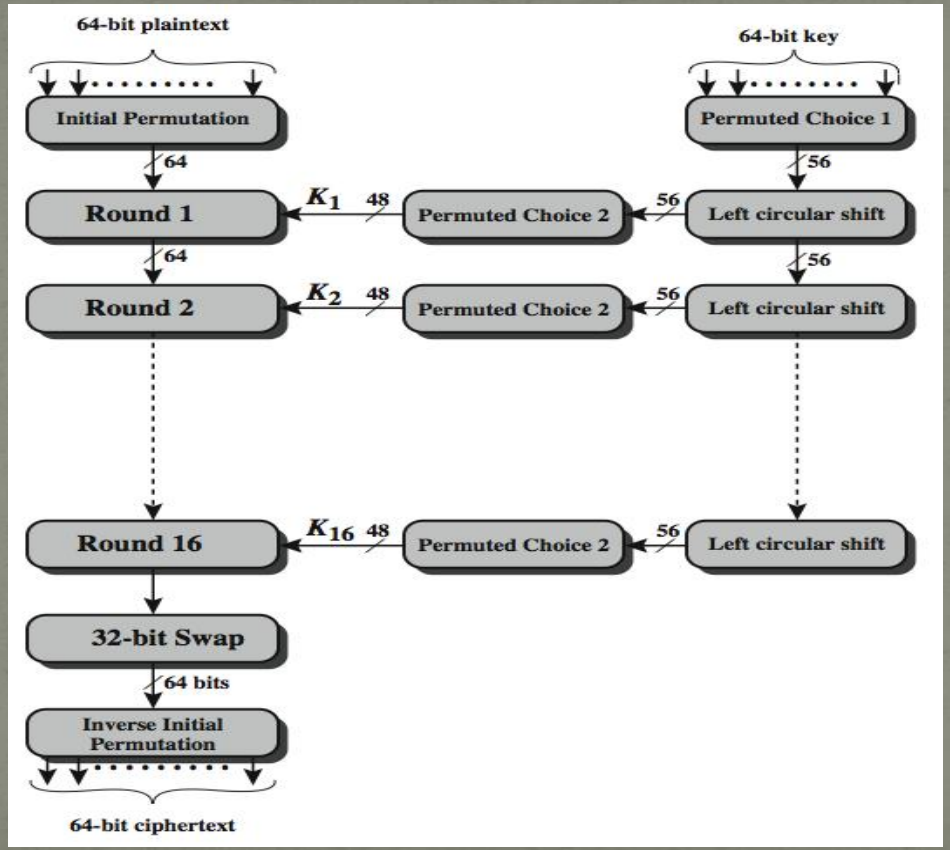
- Symmetric algorithm: same key for encryption and decryption.
- Block cipher: processes fixed-size blocks of data (64 bits).
- Uses multiple rounds of substitution and permutation.

- Input key: 64 bits (8 parity bits removed → 56 bits effective).
- Round keys: 16 subkeys generated from the main key.
- Each subkey is 48 bits.

DES Encryption Process

- 1.Initial Permutation (IP)
- 2.16 rounds of processing:
  - Expansion (E)
  - Key Mixing (XOR)
  - Substitution (S-box)
  - Permutation (P)
- 3.Final Permutation (FP)

DES Encryption Overview



Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- example:

```
IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)
```

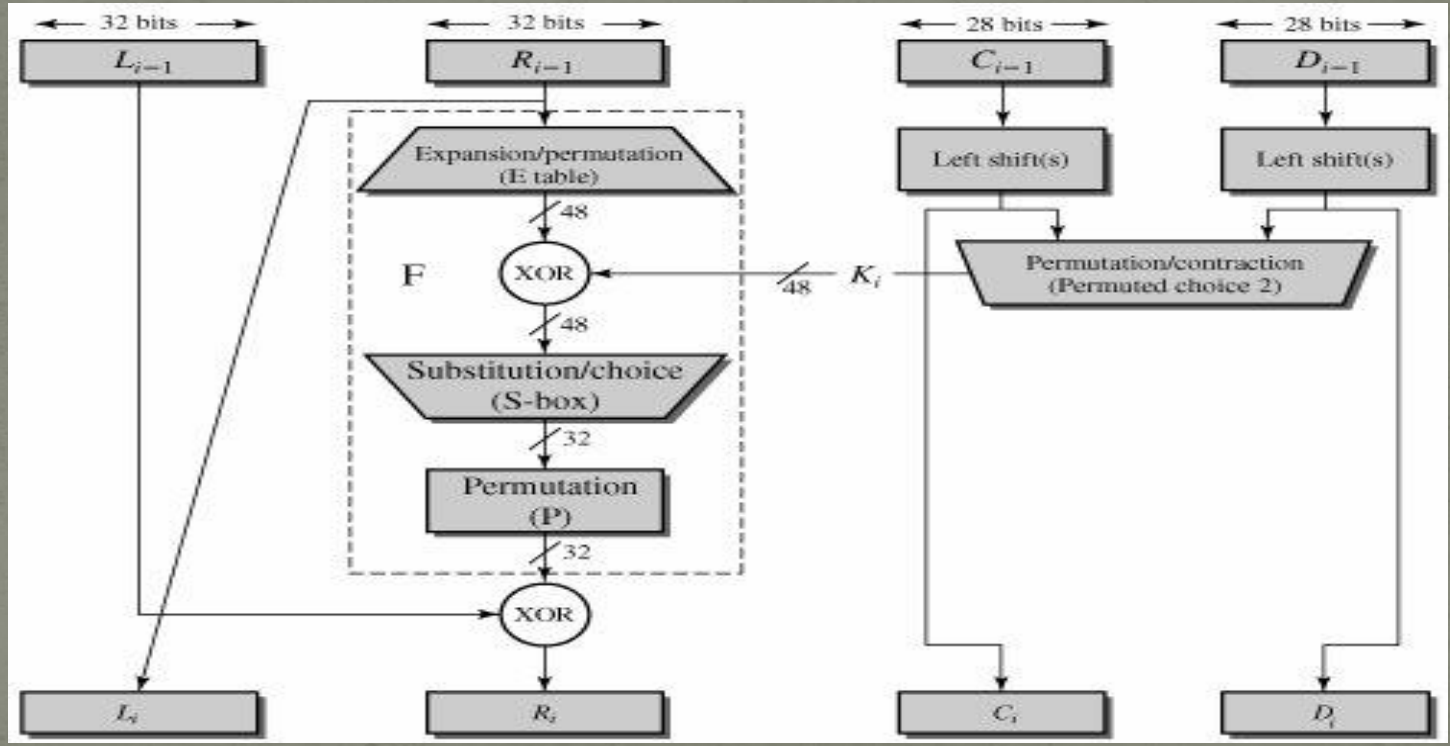
DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

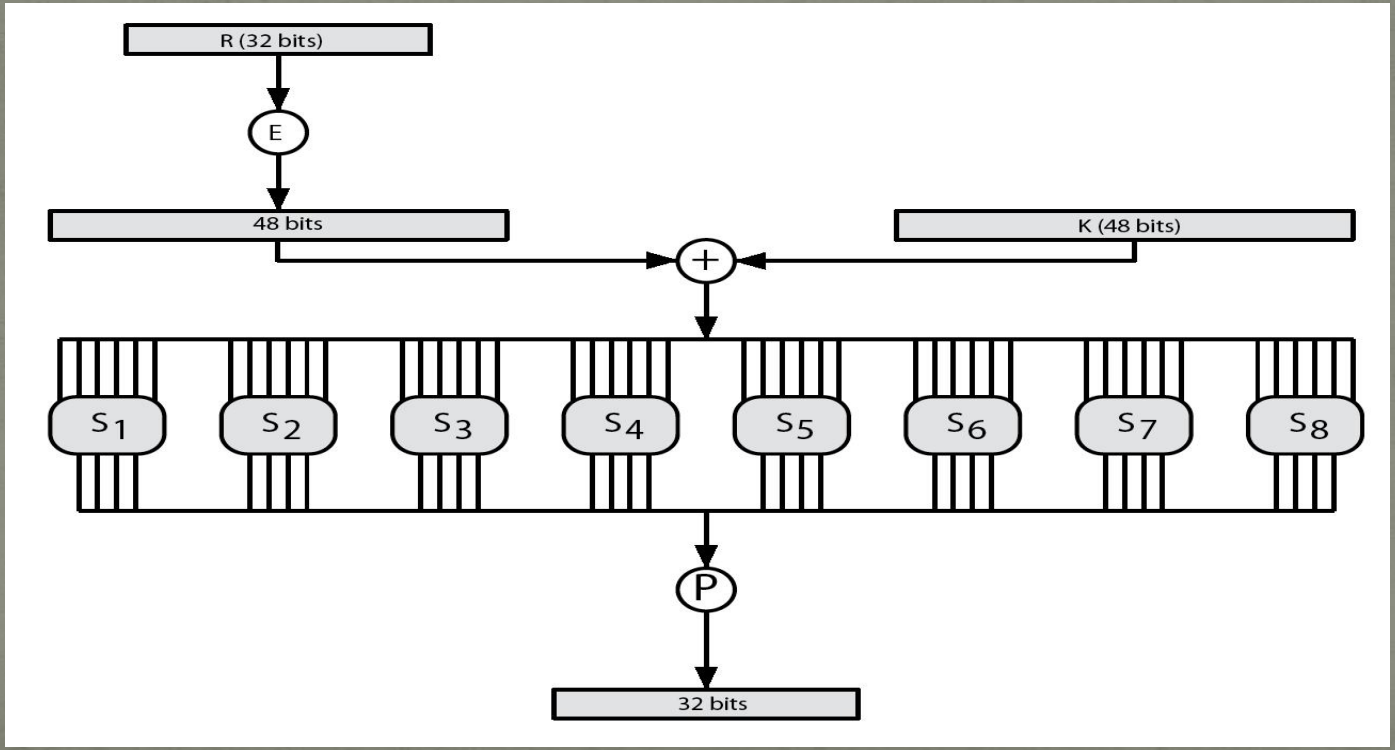
  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$
- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

Single DES Round



DES Round Structure



Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one row of 4
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

שורה	מס' עמודה															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	S_1															
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	3	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	13	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
	S_2															
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
	S_3															
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
	S_4															
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
	S_5															
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	S_6															
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
	S_7															
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
	S_8															
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
- note practical use issues in h/w vs s/w

DES Decryption

- decrypt must unwind steps of data computation
- with Feistel design, do encryption steps again using subkeys in reverse order (SK16 … SK1)
  - IP undoes final FP step of encryption
  - 1st round with SK16 undoes 16th encrypt round
  - ….
  - 16th round with SK1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

DES Example

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

Avalanche in DES

Round		δ	Round		δ
	02468aceeca86420 12468aceeca86420	1	9	c11bfc09887fbc6c 99f911532eed7d94	32
1	3cf03c0fbad22845 3cf03c0fbad32845	1	10	887fbc6c600f7e8b 2eed7d94d0f23094	34
2	bad2284599e9b723 bad3284539a9b7a3	5	11	600f7e8bf596506e d0f23094455da9c4	37
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18	12	f596506e738538b8 455da9c47f6e3cf3	31
4	0bae3b9e42415649 171cb8b3ccaca55e	34	13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
5	4241564918b3fa41 ccaca55ed16c3653	37	14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
6	18b3fa419616fe23 d16c3653cf402c68	33	15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
7	9616fe2367117cf2 cf402c682b2cefbc	32	16	75e8fd8f25896490 1ce2e6dc365e5f59	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33	IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

Avalanche Effect

- DES exhibits strong avalanche
- The Avalanche Effect in the DES is a highly desirable property where a small change in either the plaintext or the key results in a drastic and unpredictable change in the ciphertext (the encrypted output).
- In a high-quality block cipher like DES, the goal is for a single-bit change in the input (plaintext or key) to cause, on average, a change in about half of the output bits (32 out of the 64 ciphertext bits).

Summary

- have considered:
  - block vs stream ciphers
  - Feistel cipher design & structure
  - DES
    - details
    - strength
  - Differential & Linear Cryptanalysis
  - block cipher design principles