Block Cipher Primitives: Confusion and Diffusion

Let's break down these two key ideas from Claude Shannon that are essential for strong encryption algorithms, especially block ciphers like DES and AES.

1. Confusion

- **Definition:** Confusion is about making the relationship between the encryption key and the ciphertext (the encrypted output) as complicated as possible. The goal is to make it very hard for an attacker to figure out the key, even if they have the ciphertext.
- **How is it achieved?**
- The most common way is **substitution**: replacing parts of the data with other values according to a rule or table (like S-boxes in DES and AES).
- In DES, the S-boxes take bits and substitute them with other bits in a non-linear way.
- **Why is it important?**
- If you change just one bit of the key, confusion ensures that many bits of the ciphertext will change unpredictably.
- This makes it very hard to guess the key by looking at patterns in the ciphertext.

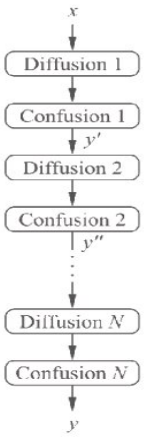
2. Diffusion

- **Definition:** Diffusion is about spreading out the influence of each bit of the plaintext (the original message) over many bits of the ciphertext. The goal is to hide any patterns or statistical properties of the plaintext.
- **How is it achieved?**
- The most common way is **permutation**: rearranging the bits in a block according to a fixed pattern.
- In DES, bit permutation is used in several steps to mix up the bits.
- In AES, a more advanced operation called MixColumns is used for diffusion.
- **Why is it important?**
- If you change just one bit of the plaintext, diffusion ensures that (statistically) about half the bits in the ciphertext will change.
- This makes it very hard for an attacker to find patterns that could reveal the original message.

Why Both Are Needed

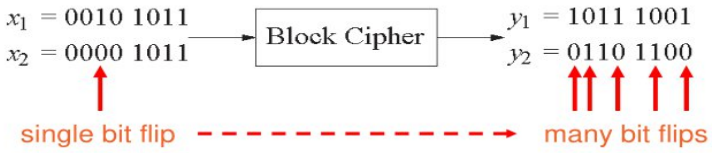
- **Confusion alone** (like simple substitution ciphers) is not enough: attackers can still find patterns in the plaintext.

Product Ciphers



- Most of today's block ciphers are *product ciphers* as they consist of rounds which are applied repeatedly to the data.
- Can reach excellent diffusion: **changing of one bit of plaintext results** *on average* in the **change of half the output bits**.

Example:



-
- **Diffusion alone** (like simple transposition ciphers) is not enough: attackers can still find relationships between the key and ciphertext.
- **Combining both** in multiple rounds (as in DES and AES) creates a **product cipher**—a strong cipher that is much harder to break.

Product Ciphers

- **Product cipher**: A cipher that applies confusion and diffusion operations in several rounds.
- In each round, the data is substituted (confusion) and permuted (diffusion).
- After several rounds, changing one bit of input (plaintext or key) will change many bits of output (ciphertext), making the cipher secure against many types of attacks.

Quick Table: Confusion vs. Diffusion

Feature	Confusion (Substitution)	Diffusion (Permutation)
Purpose	Hide key-ciphertext link	Hide plaintext-ciphertext link
Example	S-box in DES/AES	Bit permutation in DES
Effect	Key change affects many ciphertext bits	Plaintext change affects many ciphertext bits
Used in	Block & stream ciphers	Block ciphers only

Summary

- **Confusion**: Makes it hard to see how the key affects the ciphertext (usually by substitution).
- **Diffusion**: Spreads out the effect of each plaintext bit over the ciphertext (usually by permutation).
- **Product ciphers** like DES and AES use both, in multiple rounds, to achieve strong security.

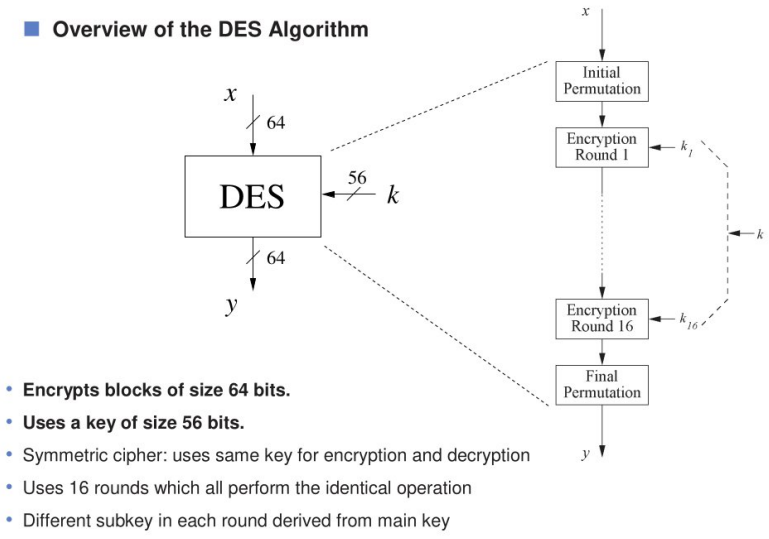
If you want to see a simple example or have questions about any term (like S-box, permutation, round, etc.), let me know and I'll explain with visuals or step-by-step examples!

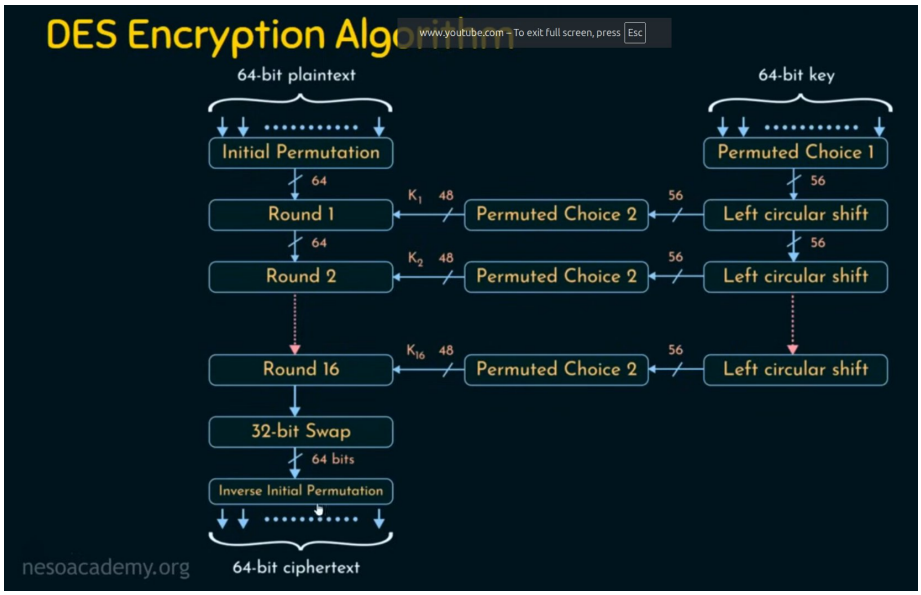
DES (Data Encryption Standard) – Full Details

Let's break down DES step by step, explaining every important term and process in detail. If you get stuck on any word or concept, let me know and I'll clarify further!

1. What is DES?

- **DES** is a symmetric-key block cipher algorithm for encrypting digital data. That means: the same key is used for both encryption and decryption.
- It was developed in the 1970s by IBM and standardized in 1977.
- DES works on **64-bit blocks** of data and uses a **56-bit key** (even though the input key is 64 bits, 8 bits are just for error checking and not used for security).



- 

2. Key Terms Explained

Block Cipher

- DES encrypts data in fixed-size chunks (blocks) of 64 bits. Each block is processed separately.

Symmetric Encryption

- The same key is used for both encrypting and decrypting data. If you have the key, you can reverse the process.

Key Length

- DES uses a 56-bit key for security. The original key is 64 bits, but 8 bits are used for parity (error checking), so only 56 bits actually protect your data.

Feistel Network

- DES uses a structure called a Feistel network. This means:
- Each block is split into two halves: Left (L) and Right (R).
- The algorithm runs for **16 rounds**. In each round, the halves are processed and swapped in a specific way.

Permutation and Substitution

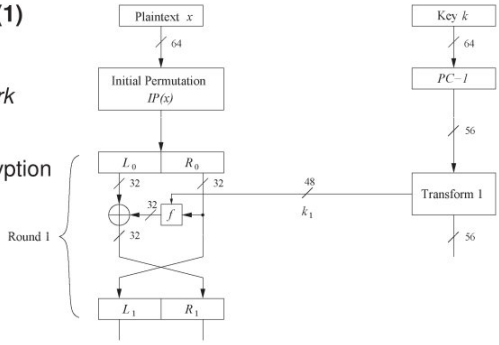
- **Permutation**: Rearranging the order of bits.
- **Substitution**: Replacing bits with other bits using a lookup table (S-box).

- These steps help hide patterns in the data, making it hard to break the encryption.

3. How DES Works – Step by Step

The DES Feistel Network (1)

- DES structure is a *Feistel network*
- Advantage: encryption and decryption differ only in keyschedule

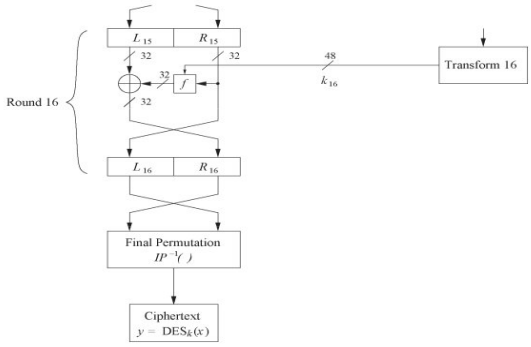


- Bitwise initial permutation, then 16 rounds
  1. Plaintext is split into 32-bit halves L_i and R_i
  2. R_i is fed into the function f, the output of which is then XORed with L_i
  3. Left and right half are swapped
- Rounds can be expressed as:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

The DES Feistel Network (2)

- L and R swapped again at the end of the cipher, i.e., after round 16 followed by a final permutation



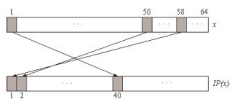
Step 1: Initial Permutation (IP)

- The 64-bit block of plaintext is rearranged according to a fixed table. This spreads out the bits for better security.

- **Initial and Final Permutation**
  - Bitwise Permutations.
  - Inverse operations.
  - Described by tables *IP* and IP^{-1}.

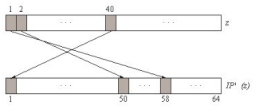
Initial Permutation

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



Final 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



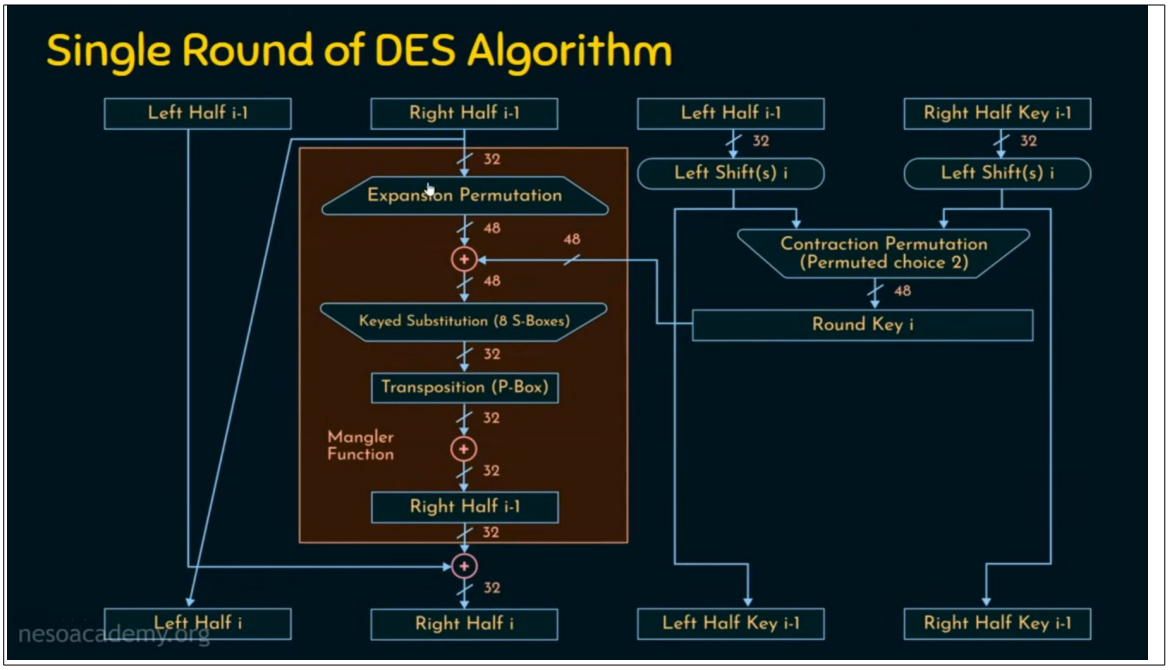
Step 2: Splitting the Block

- The permuted block is split into two 32-bit halves: Left (L0) and Right (R0).

Step 3: 16 Rounds of Processing

For each round (1 to 16):

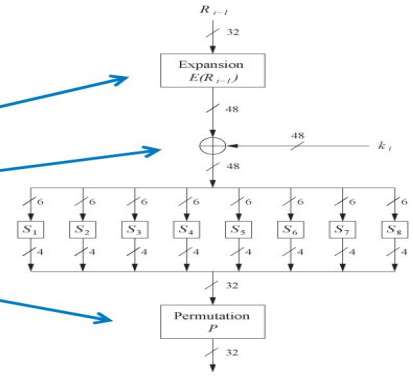
1. **Key Schedule**: A different 48-bit subkey is generated from the main key for each round.
2. **Feistel Function (f)**: The right half (R) is processed:
   - **Expansion (E)**: R is expanded from 32 to 48 bits.
   - **Key Mixing**: The expanded R is XORed with the round subkey.
   - **Substitution (S-boxes)**: The result is split into 8 groups of 6 bits, each passed through an S-box (a lookup table) to produce 4 bits per group (total 32 bits).
   - **Permutation (P)**: The 32 bits are rearranged according to a fixed table.
1. **XOR and Swap**: The output of f is XORed with the left half (L), and then the halves are swapped for the next round.



2.

The f-Function

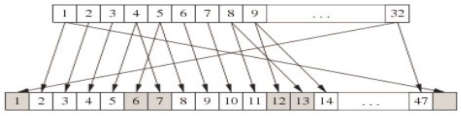
- **main operation of DES**
- *f*-Function inputs: R_{i-1} and round key k_i
- **4 Steps**:
  1. Expansion *E*
  2. XOR with round key
  3. S-box substitution
  4. Permutation

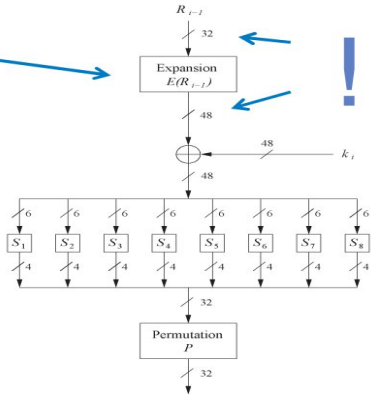


The Expansion Function E

1. **Expansion *E***
- **main purpose: increases diffusion**

E					
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

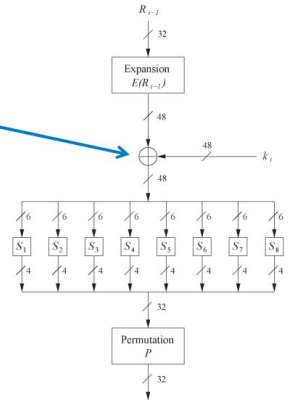




Add Round Key

2. XOR Round Key

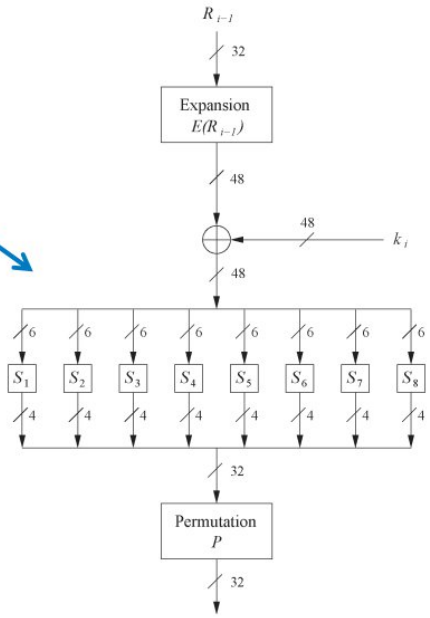
- **Bitwise XOR of the round key and the output of the expansion function *E***
- **Round keys are derived from the main key in the DES keyschedule (in a few slides)**

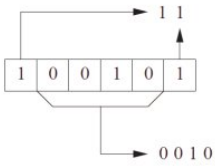


The DES S-Boxes

3. S-Box substitution

- Eight substitution tables.
- 6 bits of input, 4 bits of output.
- Non-linear and resistant to differential cryptanalysis.
- Crucial element for DES security!
- Find all S-Box tables and S-Box design criteria in *Understanding Cryptography* Chapter 3.



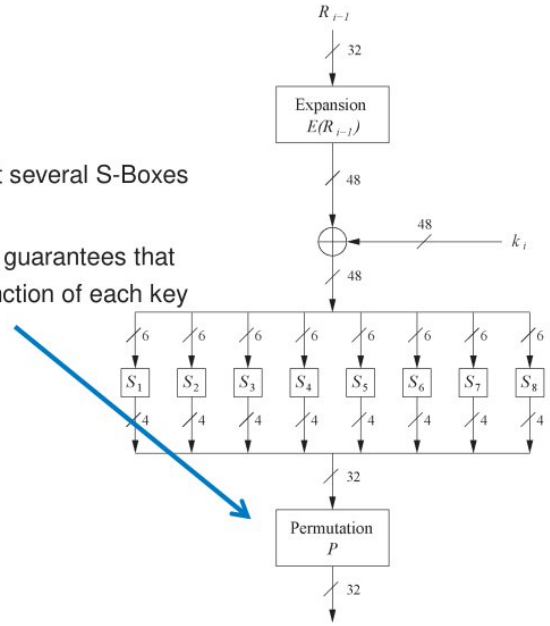


S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

The Permutation P

4. Permutation P

- Bitwise permutation.
- Introduces diffusion.
- Output bits of one S-Box effect several S-Boxes in next round
- Diffusion by E, S-Boxes and P guarantees that after Round 5 every bit is a function of each key bit and each plaintext bit.

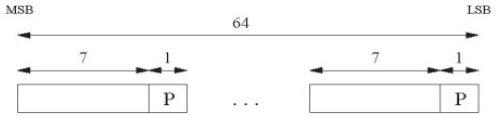


P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

3.

Key Schedule (1)

- Derives 16 round keys (or *subkeys*) k_i of 48 bits each from the original 56 bit key.
- The input key size of the DES is 64 bits: **56 bit key** and 8 bit parity:



P = parity bit

- **Parity bits are removed** in a first **permuted choice** *PC-1*: (note that the bits 8, 16, 24, 32, 40, 48, 56 and 64 are not used at all)

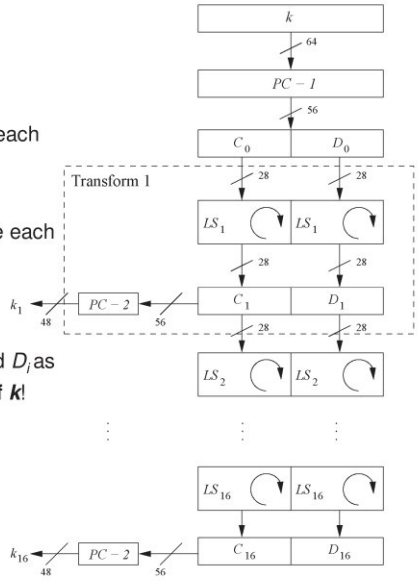
$PC-1$							
57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

4.

5.

Key Schedule (2)

- Split key into 28-bit halves C_0 and D_0.
- In **rounds i = 1, 2, 9 ,16,** the two halves are each rotated left by **one bit**.
- In **all other rounds** where the two halves are each rotated left by **two bits**.
- *In each round i permuted choice* ***PC-2*** selects a permuted subset of 48 bits of C_i and D_i as round key k_i, i.e. **each k_i is a permutation of k**!



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

- **Note:** The total number of rotations:
  4 x 1 + 12 x 2 = 28 $\Rightarrow D_0 = D_{16}$ and $C_0 = C_{16}$!

Chapter 3 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Step 4: Final Permutation (IP^{-1})

- After 16 rounds, the halves are combined and rearranged again using the inverse of the initial permutation table.
- The result is the ciphertext (encrypted data).

4. Key Schedule – How Subkeys Are Made

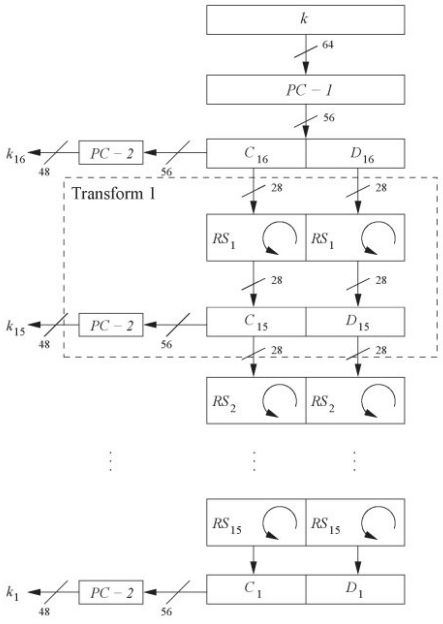
- The original 56-bit key is split into two halves.
- For each round, both halves are shifted left (rotated), and then 48 bits are selected from the combined halves using a fixed table.
- This process creates a unique subkey for each round.

5. Decryption

- Decryption uses the same process as encryption, but the subkeys are applied in reverse order (from round 16 to round 1).
- This is possible because of the Feistel structure.

Decryption

- In **Feistel ciphers** only the keyschedule has to be modified for decryption.
- Generate the same 16 round keys in reverse order.
  (for a detailed discussion on why this works see *Understanding Crptography* Chapter 3)
- **Reversed key schedule:**
  As $D_0=D_{16}$ and $C_0=C_{16}$ the first round key can be generated by applying *PC-2* right after *PC-1* (no rotation here!).
  All other rotations of *C* and *D* can be reversed to reproduce the other round keys resulting in:
  - No rotation in round 1.
  - One bit rotation **to the right** in rounds 2, 9 and 16.
  - Two bit rotations **to the right** in all other rounds.



Chapter 3 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

6. Why DES Is No Longer Secure

- The 56-bit key is too short for modern computers. Brute-force attacks (trying every possible key) are now practical.
- DES has been replaced by more secure algorithms like **AES** (Advanced Encryption Standard).
- **Triple DES (3DES)**: To improve security, DES can be applied three times with different keys, making brute-force much harder.

7. DES Variants

- **Triple DES (3DES)**: Encrypts data three times with two or three keys. Much more secure, but slower.
- **DESX**: Adds extra steps to make DES harder to break, but still not as secure as AES.

8. Where DES Was Used

- **Data Transmission**: Securing data sent over networks.
- **Data Storage**: Protecting files and databases.
- **Authentication**: Verifying passwords and credentials.

9. Common Confusing Terms (Quick Glossary)

- **Cipher**: An algorithm for encrypting or decrypting data.
- **Plaintext**: The original, readable data.
- **Ciphertext**: The encrypted, unreadable data.
- **Key**: A secret value used to encrypt/decrypt data.
- **XOR (Exclusive OR)**: A logical operation used to mix data and keys.
- **S-box**: A substitution box; a table that replaces input bits with output bits in a non-linear way.
- **Permutation Table**: A fixed table that tells how to rearrange bits.

10. Summary Table: DES at a Glance

Feature	Value/Description
Block Size	64 bits
Key Size	56 bits (actual)
Rounds	16
Structure	Feistel Network
Main Steps	Permutation, Substitution
Secure Today?	No (key too short)
Successor	AES, Triple DES

11. Why Learn DES?

- DES is a classic example of how symmetric encryption works.
- It teaches important concepts: block ciphers, Feistel networks, permutations, substitutions, and key schedules.
- Even though it's outdated, understanding DES helps you learn modern cryptography.

If you want to see visual step-by-step explanations in Bangla, check out these YouTube tutorials:

- [DES Algorithm Bangla Explanation]
- [DES Round Key Generator Bangla]
- [DES Function Bangla]

Let me know which part you want to dive deeper into, or if you want a worked example with actual bits and keys!

What is Triple DES?

Triple DES applies the DES algorithm three times to each data block. Instead of using a single 56-bit key, 3DES uses either two or three separate DES keys, effectively increasing the key length and making brute-force attacks much more difficult.

How Triple DES Works

- **Encryption Process:** The plaintext is encrypted with the first key, decrypted with the second key, and then encrypted again with the third key. This sequence is often called Encrypt-Decrypt-Encrypt (EDE).
- **Keying Options:**
- **Three-key 3DES:** Uses three independent 56-bit keys (total 168 bits).
- **Two-key 3DES:** Uses two keys, where the first and third keys are the same (total 112 bits).
- **Decryption:** The process is reversed—decrypt with the third key, encrypt with the second, and decrypt with the first.

Why Triple DES?

- **Security:** Standard DES is vulnerable to brute-force attacks due to its 56-bit key. Triple DES increases the effective key length, making such attacks impractical with current technology.
- **Compatibility:** 3DES is backward compatible with DES, allowing older systems to upgrade security without major changes.
- **Usage:** Still used in many legacy systems, especially in banking, though it is being replaced by more modern algorithms like AES.

Limitations

- **Performance:** Triple DES is slower than DES and AES because it performs three rounds of DES operations for each block.
- **Deprecation:** Modern standards recommend using AES instead, as 3DES is considered less secure and less efficient for new applications.

Summary Table

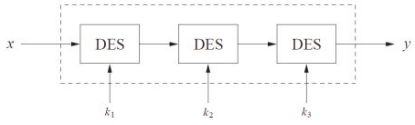
Feature	DES	Triple DES (3DES)
Key Length	56 bits	112 or 168 bits

Security	Weak (brute-force)	Stronger (legacy use)
Speed	Fast	Slower
Usage	Obsolete	Legacy systems
Successor	3DES, AES	AES

Triple DES – 3DES

- Triple encryption using DES is often used in practice to extend the effective key length of DES to 112. For more info on multiple encryption and effective key lengths see Chapter 5 of *Understanding Cryptography*.

$$y = DES_{k_3}(DES_{k_2}(DES_{k_1}(x)))$$



- Alternative version of *3DES:* $y = DES_{k_3}(DES_{k_2}^{-1}(DES_{k_1}(x)))$.

  Advantage: choosing $k_1=k_2=k_3$ performs single DES encryption.
- No practical attack known today.
- Used in many legacy applications, i.e., in banking systems.

How Triple DES (3DES) Works

Triple DES (3DES) is an encryption algorithm that strengthens the original DES by applying it three times with multiple keys. Here's a step-by-step explanation of how it works:

1. Key Generation

- Three separate 56-bit keys are created: K_1K_1, K_2K_2, and K_3K_3.
- Sometimes, only two keys are used (K_1K_1 and K_2K_2), with $K_3=K_1K_3=K_1$ for a slightly lower security level.

2. Data Preparation

- The plaintext message is divided into blocks of 64 bits, just like in regular DES.

3. Encryption Process (EDE Mode)

For each 64-bit block, the following steps are performed:

1. **Encrypt** with K_1:
   - The plaintext block is encrypted using DES and the first key K_1.
1. **Decrypt** with K_2:
   - The result from step 1 is decrypted using DES and the second key K_2.
1. **Encrypt** with K_3:
   - The result from step 2 is encrypted again using DES and the third key K_3.

This sequence is called **Encrypt-Decrypt-Encrypt (EDE)**.

Mathematically:

$$\text{Ciphertext} = E_{K_3}(D_{K_2}(E_{K_1}(\text{Plaintext})))$$

4. Decryption Process

To decrypt, the steps are reversed:

1. **Decrypt** with K_3
2. **Encrypt** with K_2
3. **Decrypt** with K_1

Mathematically:

$$\text{Plaintext} = D_{K_1}(E_{K_2}(D_{K_3}(\text{Ciphertext})))$$

5. Why This Sequence?

- The middle decryption step (with K_2) is used for compatibility with single DES. If all three keys are the same, 3DES acts like regular DES.
- Using three keys increases the effective key length to 168 bits, making brute-force attacks much harder than with single DES.

6. Variants of Triple DES

- **Three-key 3DES:** Uses three independent keys (highest security).
- **Two-key 3DES:** Uses two keys (K_1 and K_2), with $K_3 = K_1$ (lower security, but still much stronger than DES).

7. Advantages and Limitations

- **Advantages:**
- Much stronger than single DES due to longer key length and triple encryption.
- Backward compatible with DES.
- **Limitations:**
- Slower than DES and modern algorithms like AES.
- Considered legacy; AES is now preferred for new systems.

Summary Table: Triple DES Steps

Step	Operation	Key Used
1	Encrypt	K_1
2	Decrypt	K_2
3	Encrypt	K_3

For decryption, the order is reversed: Decrypt with K_3, Encrypt with K_2, Decrypt with K_1.