Differences:

Stream Cipher	Block Cipher
1) convert one symbol of plaintext directly into a symbol of ciphertext	1) encrypt a group of plaintext symbols as one block
2) Speed of transformation: The transformation is fast as algorithms are linear in time and constant in space.	2) Speed of Transformation: The transformation is slow as an entire block must be accumulated before encryption / decryption can begin.
3) Low error propagation: If there is an error while encrypting one symbol,it will not affect subsequent symbols.	3) Error propagation: An error in one symbol may corrupt the entire block
4) Low diffusion: All information of a plaintext symbol is contained in a single ciphertext symbol.	4) High diffusion: Information from one plaintext symbol is diffused into several ciphertext symbols.
5) High Risk of Tampering: An active interceptor who breaks the algorithm might insert fake text that looks authentic	5) Low Risk of Tampering: Difficult for the attacker to insert symbols without detection.

DES(Data Encryption Standard): It is a symmetric key algorithm for the encryption of digital data. It is a one kind Block Cipher algorithm.

Avalanche Effect & DES Encryption:

The Avalanche effect refers to a very small change in the input that will lead to a very big change in the output.

how the Avalanche Effect works in DES encryption:

Bit-Level Transformation: In DES, the plaintext undergoes multiple rounds of permutation and substitution, where each round includes operations such as expansion, XOR with a round key, substitution through S-boxes, and permutation. These operations cause the bits of the plaintext to be transformed in complex ways.

Key Dependency: The transformation process in DES is heavily dependent on the encryption key. Each round uses a different subkey generated from the main encryption key. Thus, even a slight change in the encryption key will lead to a completely different set of subkeys, causing a significant change in the encryption process and consequently, in the ciphertext.

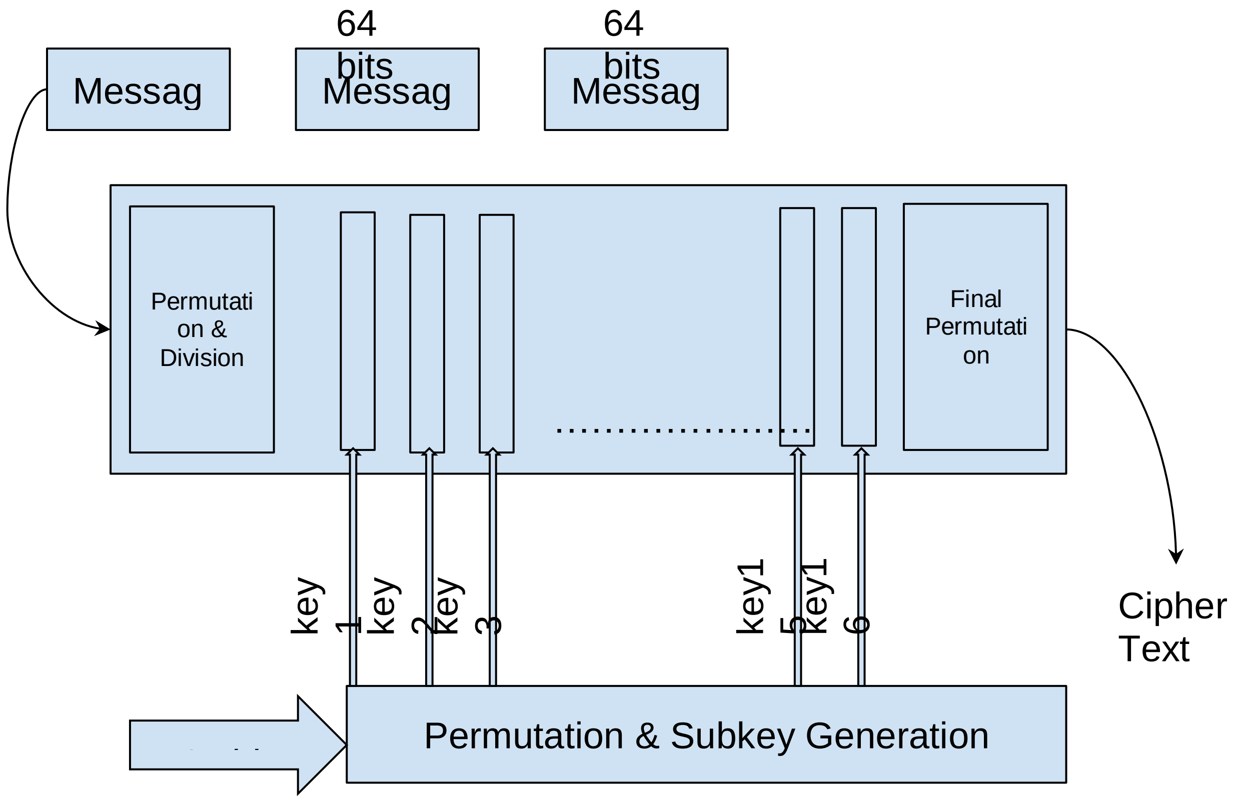
Diffusion: DES ensures that each bit of the plaintext affects many bits in the subsequent rounds. This diffusion property ensures that any change in the plaintext propagates through the encryption process, leading to a completely different ciphertext.

Multiple Rounds: DES employs multiple rounds of encryption (16 rounds in the standard version). Each round contributes to the Avalanche Effect by further mixing and scrambling the data, making it increasingly difficult for an attacker to predict the output ciphertext based on the input plaintext.

XOR Operations: XOR operations with the subkeys and intermediate data ensure that even small changes in the input (plaintext or key) produce significant changes in the intermediate values, which in turn affect the subsequent rounds, magnifying the Avalanche Effect.

https://www.youtube.com/watch?v=-j80aA8q_IQ

DES Encryption Walkthrough:



1st phase(16 subkeys generation from 64 bit key):
2nd phase(Pass a block of message for permutation and division)
3rd phase (Pass the output of 2nd phase to input of 1st round

<u>1st Phase:</u>

length(key1) = 48 bits
length(key2) = 48 bits
.
.
.
length(key16) = 48 bits

64 bit key(hexa), K = 133457799BBCDFF1
(binary) K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

Convert K from 64 bit to 56 bits by using PC-1(8x7) table

PC-1 :
57 49 41 33 25 17 9
1   58 50 42 34 26 18
10 2   59 51 43 35 27
……………………….
.. …………………….
……………………….
……………………….
……………………….

49 57
K(64 bits) = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
K(56 bits) = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

We divide the new K into 2 part CO(28) and DO(28),

C0 (28) =   1111000 0110011 0010101 0101111
D0 (28) =   0101010 1011001 1001111 0001111

Given,

Iteration Number	Number Left Shift
1	1
2	1
3	2
4	2
.	
.	
.	
.	
16	1

After iteration-1(left shift=1),
C1 = 111000 0110011 0010101 01011111
D1 = 101010 1011001 1001111 00011110

………

After

Multiple DES and Block Cipher Modes of Operations (Slide-4)

Drawbacks of DES:

- **Small Key Size:** DES uses a relatively small key size of 56 bits. This limited key space makes DES vulnerable to brute-force attacks, where an attacker can systematically try all possible keys until the correct one is found. With advances in computing power, a 56-bit key can be feasibly cracked within a reasonable amount of time using modern hardware.

- **Vulnerability to Attacks:** DES is susceptible to various cryptographic attacks, including differential cryptanalysis and linear cryptanalysis, which can compromise its security and reveal plaintext information. These attacks exploit patterns in the encryption process to deduce information about the key.

- **Limited Block Size:** DES operates on fixed-size blocks of data (64 bits), which can pose challenges for encrypting large files or streaming data. It requires additional mechanisms such as chaining modes (e.g., Cipher Block Chaining) to encrypt data longer than one block.

- **Aging Design:** DES was developed in the 1970s, and its design predates modern cryptographic standards and best practices. Over time, weaknesses and vulnerabilities in the algorithm have been discovered, reducing its suitability for secure communications in today's digital environment.

AES stands for Advanced Encryption Standard:

Key features and aspects of AES include:

- Symmetric-Key Algorithm: AES is a symmetric-key algorithm, meaning the same key is used for both encryption and decryption of data. This key is kept secret and must be shared securely between parties involved in the communication.

- Key Sizes: AES supports key sizes of 128, 192, or 256 bits. These larger key sizes make AES more resistant to brute-force attacks compared to DES, which had a fixed key size of 56 bits.

- Block Cipher: AES operates on fixed-size blocks of data, with a block size of 128 bits. It encrypts and decrypts data in blocks, making it suitable for encrypting large files or streaming data.
- Strong Security: AES has undergone extensive analysis by cryptographers worldwide, and no practical vulnerabilities have been discovered. It is considered secure against all known cryptographic attacks when implemented correctly with a sufficiently strong key.
- Efficiency: AES is designed to be computationally efficient, making it suitable for use in a wide range of devices and applications, including embedded systems, mobile devices, and cloud computing environments.
- Standardization: AES has been adopted as a standard encryption algorithm by governments, organizations, and industries worldwide. Its standardization ensures interoperability and compatibility across different systems and platforms.

Overall, AES is widely regarded as one of the most secure and efficient encryption algorithms available today.

Meet in the Middle Attack:

Meet-in-the-middle is a known plaintext attack that can greatly reduce the number of brute-force permutations required to decrypt text that has been encrypted by more than one key. Such an attack makes it much easier for an intruder to gain access to data.

A meet-in-the-middle attack targets block cipher cryptographic functions. The intruder applies brute-force techniques to both the plaintext, which is ordinary text before it is encrypted, and the ciphertext, or encrypted text that has been transformed from plaintext, of a block cipher.

The intruder then attempts to encrypt the plaintext according to various keys to achieve an intermediate ciphertext, or text that has only been encrypted by one key. Simultaneously, the intruder attempts to decrypt the ciphertext according to various keys, seeking a block of intermediate ciphertext that is the same as the one created by encrypting the plaintext. If there is a match of intermediate ciphertext, it is highly probable that the key used to encrypt the plaintext and the key used to decrypt the ciphertext are the two encryption keys used for the block cipher.

Modes of Operation:

	Electronic Codebook	Cypher Block Chaining	Cipher Feedback	Output Feedback	Counter
Description	Encrypts each block of plain text independently	Perform XOR operation on each plaintext block with the previous plain text block.	- Produce Keystream block using previous block cipher(or Initialize Vector for 1st block) - Keystream block XOR plaintext block = block ciphertext	- Produce keystream blocks using encrypted initialize vector only. - Keystream block XOR plaintext block = block ciphertext	
Security	Identical plaintext blocks produce identical ciphertext blocks. This can security concern	Provides Confidentiality	Ensure confidentiality	Ensure confidentiality	Ensure confidentiality
Parallelism	Yes. Each block is encrypted independently.	None	Yes	Yes	Highly parallel
Error Propagation	No. Identical plain text blocks produce identical ciphertext blocks.	Yes. Error in one block can affect subsequent blocks	Yes. Due to XOR operation	No.	No.
Random Access	Yes	No. Each block depends on the previous block.	Yes	Yes	Yes. It is easy to access any part of the encrypted data without having to decrypt

					everything before it.
Use case	For encrypting short and non-repetitive data blocks	Network Data encryption, file encryption, database encryption, email encryption	Streaming Encryption, File Encryption	Secure messaging, file, disk, random access encryption etc.	