Five best techniques used in encryption

There are two types of encryption techniques:-
Symmetric: Use of a single key to encrypt/decrypt. It sacrifices security over speed.
Asymmetric: Uses public and private keys. A public key is used for encryption/signature verification. A private key is used for decryption/signature generation along with authentication. It sacrifices speed over security.

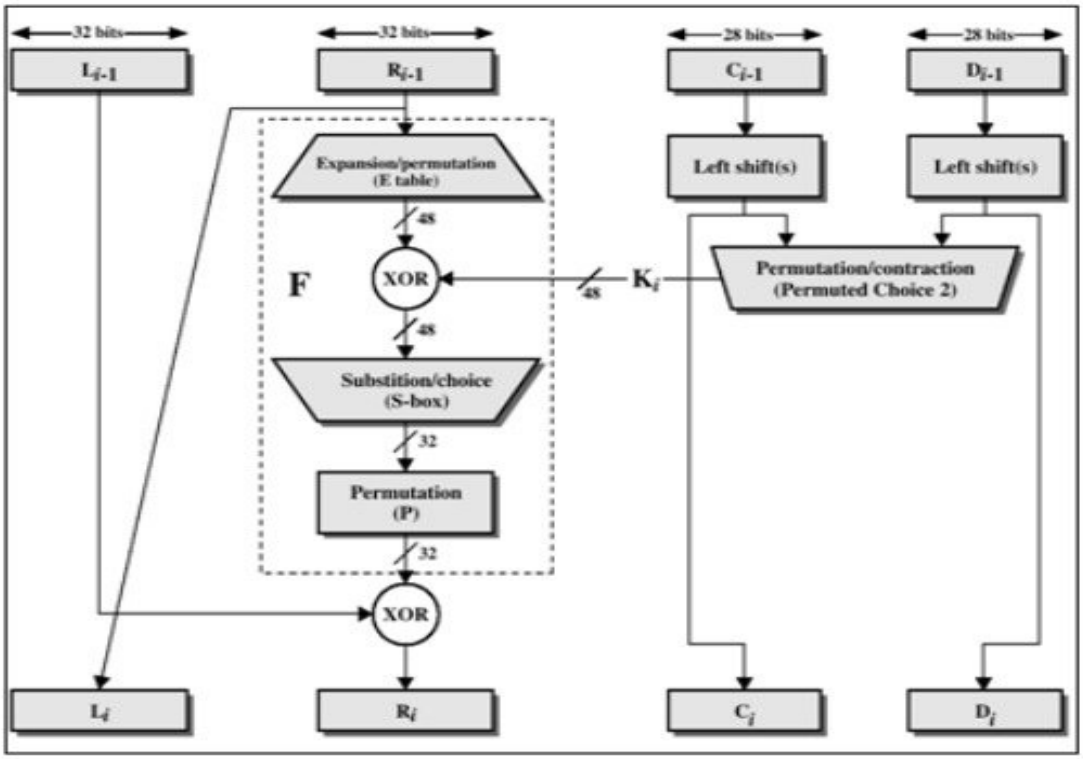
Based on requirements and type of data we use different encryption techniques.
Symmetric Encryption

1. **3DES(Triple Data Encryption Standard)**
   It is the upgraded version of DES.
   3DES uses three or two keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence

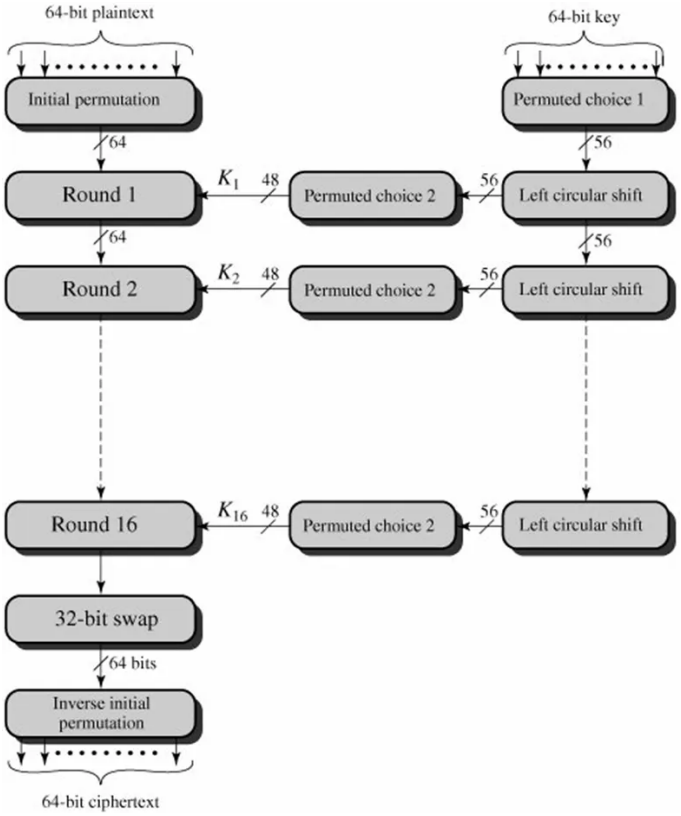
   Given a plaintext P, ciphertext C is generated as C=E(K3, D(K2, E(K1, P)))
   where E[K, X] encryption of X using key K
   D[K, Y] decryption of Y using key K

   Decryption is simply the same operation with the keys reversed:
   P = D(K1 , E(K2 , D(K3 , C)))

   Single round DES



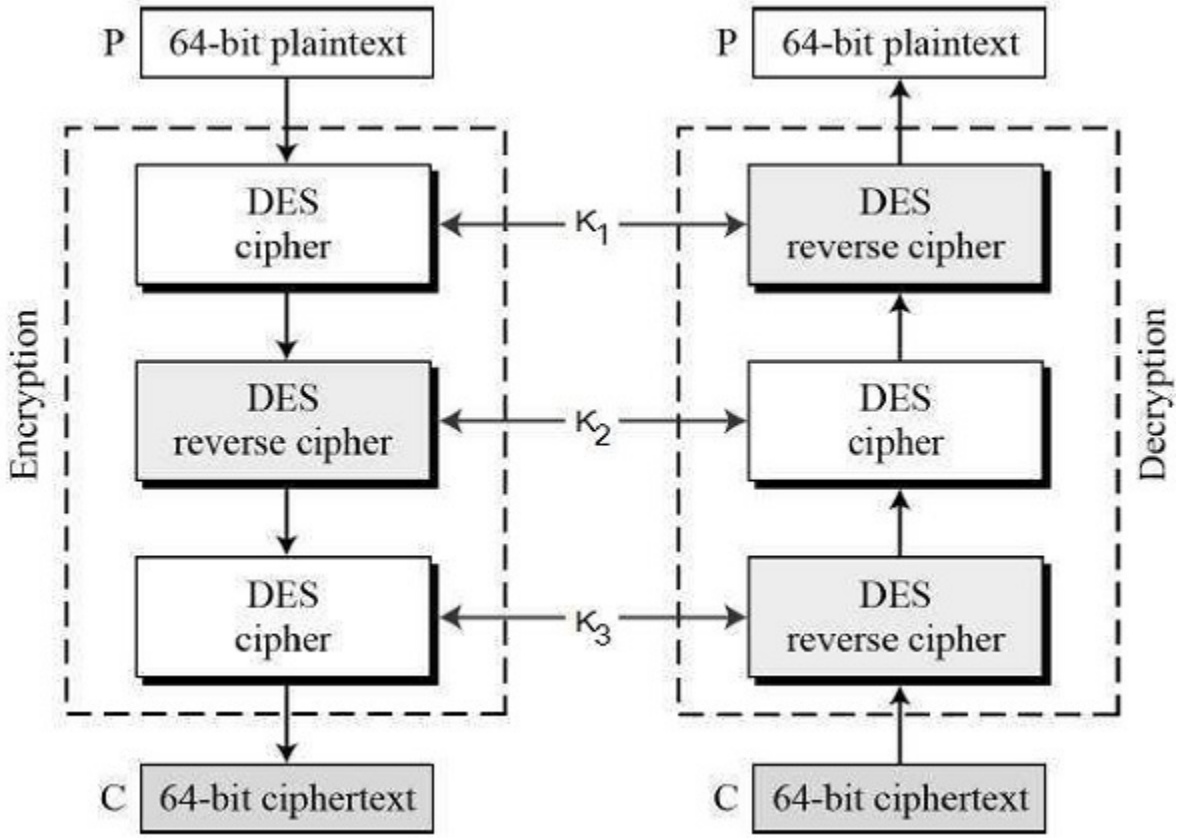
DES Encryption:



Converts 64-bit blocks of text into ciphertext by dividing blocks into two separate 32-bit blocks and applies encryption independently.
The encryption process of the plaintext proceeds in 3 phase
• The 64-bit plaintext passes through an initial permutation that rearranges the bits to produce the permuted input.
• This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.
• The left and right halves of the last output are swapped to produce the pre-output.
• Finally, the pre-output is passed through a permutation [IP^{-1}] which is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

•The right-hand portion of the Figure shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function.
• Then, for each of the sixteen rounds, a subkey (Ki) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits

Triple-DES Encryption and Decryption:



Applies DES algorithm thrice to each block

2. Blowfish

Blowfish is an encryption technique designed by Bruce Schneier in 1993 as an alternative to DES Encryption Technique. It's a block cipher algorithm. The size of the input is bits and the length of the key is variable.

Properties:

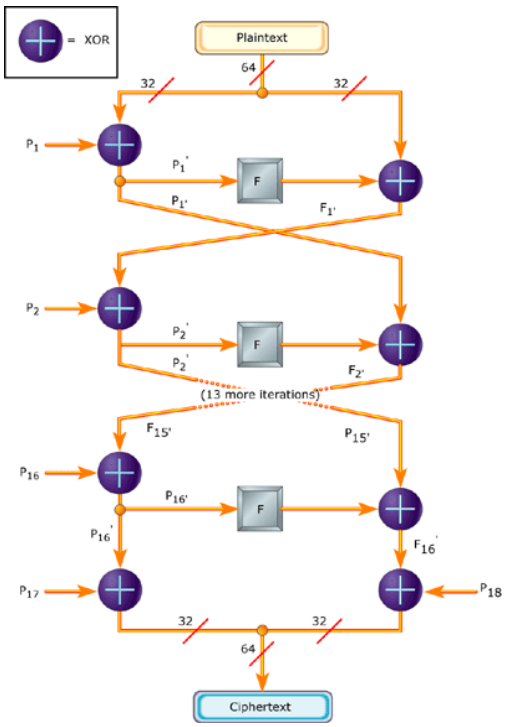
- Fast
- Takes less memory
- Simple to understand and implement
- More secured because of variable length key

The blowfish algorithm has two steps:

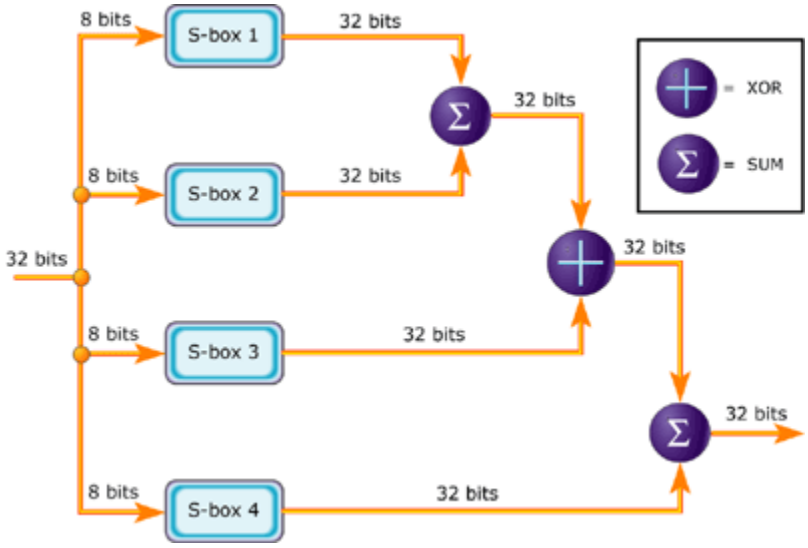
1. Key generation
    - Keys are stored in an array
      K1, K2,K3,...,Kn[1<=n<=14]
      The length of each block is 32 bits
    - Initialize an array(P)
      P1,P2,P3,...,P18
      The length of each word is 32 bits

- Initialise S-boxes(4)
  S1 => S0,S1,...,S255
  S2 => S0,S1,...,S255
  S3 => S0,S1,...,S255
  S4 => S0,S1,...,S255
- Initialise each element of P-array and S-boxes with hexadecimal values
- XOR operations between P array and Key Array
  P1 = P1 XOR K1
  P2 = P2 XOR K2
  .
  .
  .
  P14 = P14 XOR K14
  P15 = P15 XOR K1
  .
  .
  .
  P18 = P18 XOR K4
- Take 64-bit PlainText (Initially all bits are 0)
  Sub key is generated

2. Data encryption



Blowfish encrypts data in 64-bit blocks and has a variable key length between 32 and 448 bits. A graphical representation of the blowfish algorithm



Graphical Representation of Function F

AES(Advanced Encryption System):

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES.

The features of AES are as follows −

- Symmetric key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger and faster than Triple-DES
- Provide full specification and design details
- Software implementable in C and Java

AES relies on the substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling the input data. AES performs operations on bytes of data rather than in bits.
The number of rounds depends on the key length as follows :

- 128-bit key – 10 rounds
- 192-bit key – 12 rounds
- 256-bit key – 14 rounds

A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.

Encryption:
AES considers each block as a 16-byte (4-byte x 4-byte = 128) grid in a column-major arrangement.
Each round comprises 4 steps :

- SubBytes
- ShiftRows
- MixColumns

Decryption :
The stages in the rounds can be easily undone as these stages have an opposite to them which when performed reverts the changes. Each 128 block goes through 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption are as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte
- Add Round Key

Application

- Wireless security
- Mobile apps
- Libraries in many software development languages
- VPN Implementations
- Operating system components such as file systems

Asymmetric Encryption:

4. RSA:

- It was invented by Rivest, Shamir and Adleman in the year 1978 and hence the name RSA algorithm.
- Best known and widely used public-key scheme

There are two broad components when it comes to RSA cryptography, they are:

- Key Generation: Generating the keys to be used for encrypting and decrypting the data to be exchanged.
- Encryption/Decryption Function: The steps that need to be run when scrambling and recovering the data.

Key Generation

- Choose two large prime numbers (p and q)
- Calculate n = p*q and z = (p-1)(q-1)
- Choose a number e where 1 < e < z
- Calculate d = e-1mod(p-1)(q-1)
- You can bundle private key pair as (n,d)
- You can bundle public key pair as (n,e)

Encryption/Decryption Function

- If the plaintext is m, ciphertext = m^e mod n.
- If the ciphertext is c, plaintext = c^d mod n

Advantages of RSA

- No Key Sharing: RSA encryption depends on using the receiver's public key, so you don't have to share any secret key to receive messages from others.
- Proof of Authenticity: Since the key pairs are related to each other, a receiver can't intercept the message since they won't have the correct private key to decrypt the information.
- Faster Encryption: The encryption process is faster than that of the DSA algorithm.
- Data Can't Be Modified: Data will be tamper-proof in transit since meddling with the data will alter the usage of the keys. And the private key won't be able to decrypt the information, hence alerting the receiver of manipulation.

5. ECC:

•majority of public-key crypto use either integer or polynomial arithmetic with very large numbers/polynomials

•imposes a significant load in storing and processing keys and messages

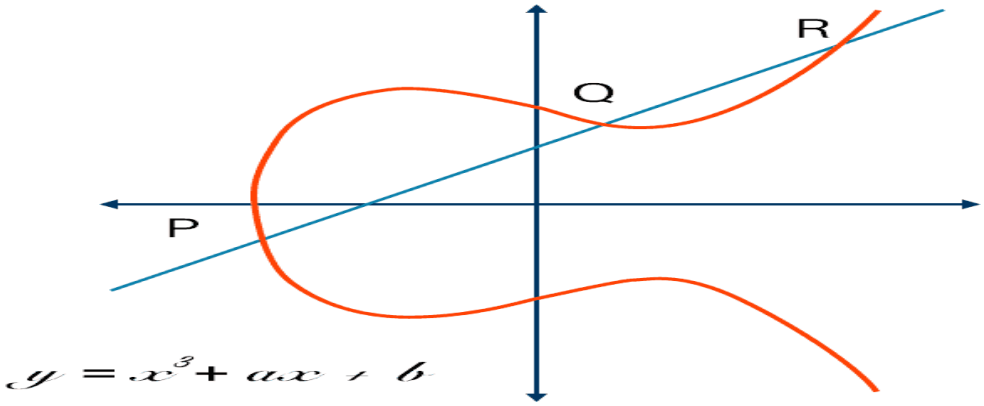
•an alternative is to use elliptic curves

•offers the same security with smaller bit sizes

•Shorter key lengths

–Encryption, Decryption and Signature Verification speed up

–Storage and bandwidth savings



•select base point $G=(x_1,y_1)$ with large order n which both the sender and receiver agree on.

•A & B select private keys $n_A<n$, $n_B<n$

•compute public keys: $P_A=n_A\times G$, $P_B=n_B\times G$

•compute shared key: $K=n_A\times P_B$, $K=n_B\times P_A$

–the same since $K=n_A\times n_B\times G$

•encode any message M as a point on the elliptic curve $P_m=(x,y)$

•each user chooses private key $n_A<n$

•and computes public key $P_A=n_A\times G$

•to encrypt pick random k: $C_m=\{kG, P_m+k P_b\}$,

• decrypt C_m compute:

$$P_m+kP_b-n_B(kG) = P_m+k(n_BG)-n_B(kG) = P_m$$

Applications of ECC:

–Wireless communication devices

–Smart cards

–Web servers that need to handle many encryption sessions

–Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems