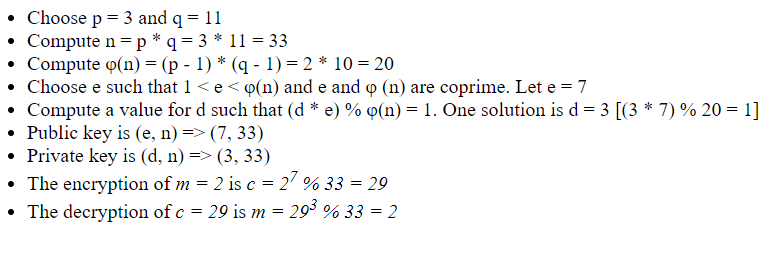
**RSA**

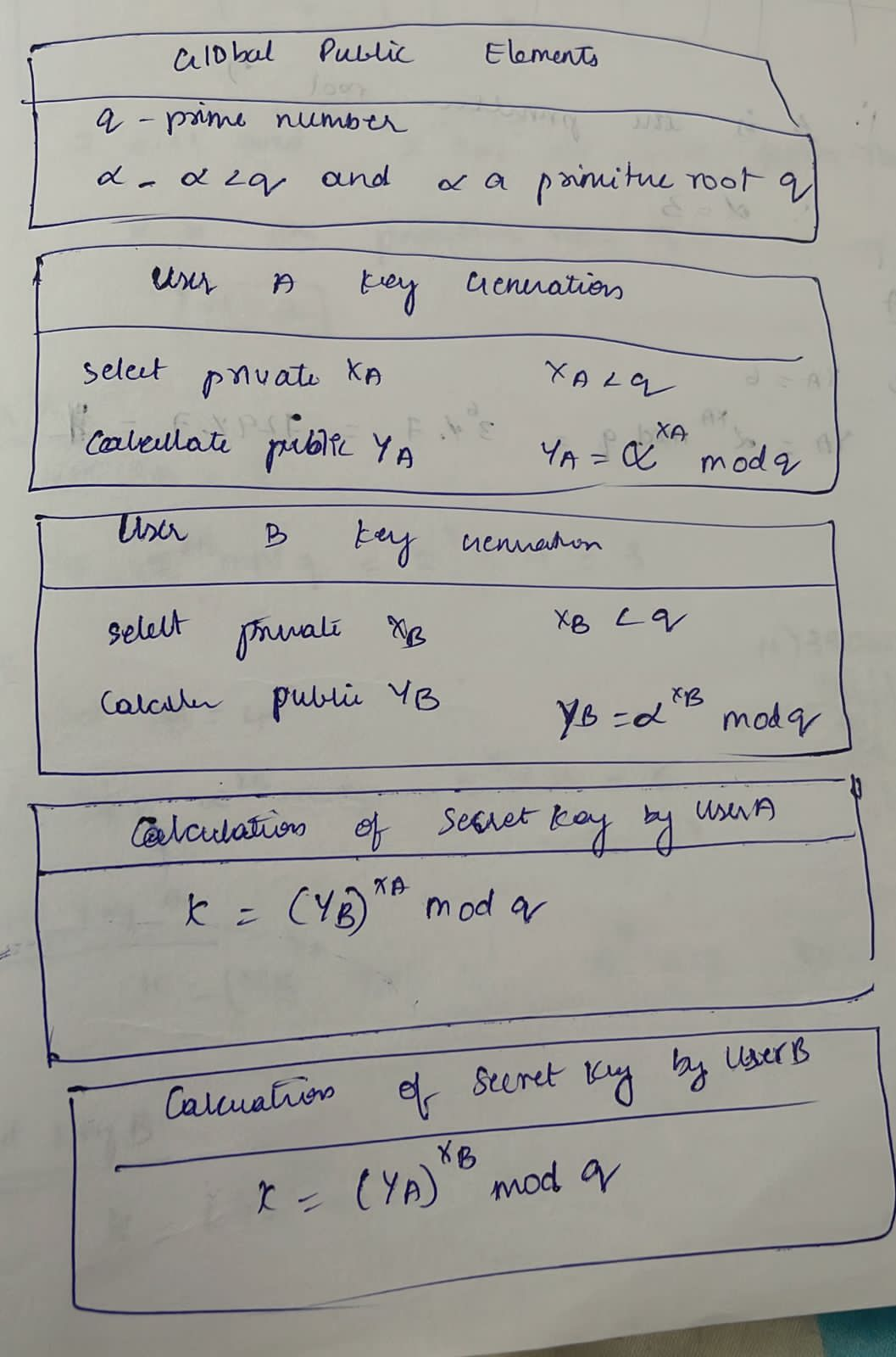
* It was developed by Rivest, Shamir and Adleman
* RSA algorithm is asymmetric cryptography algorithm.
* Asymmetric actually means that it works on two different keys i.e. **Public Key** and **Private Key.** As the name describes that the Public Key is given to everyone and Private key is kept private.

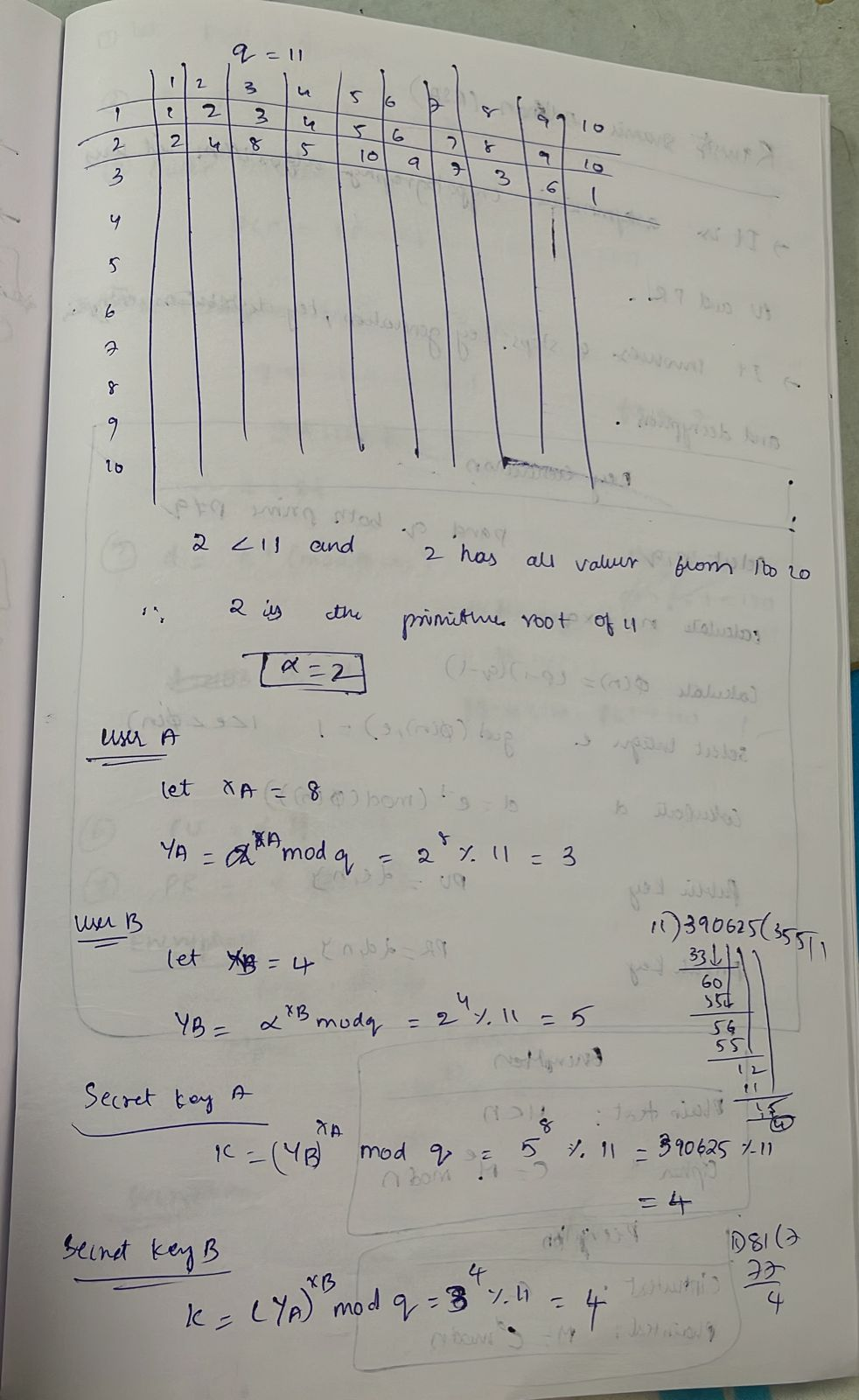
Steps for RSA Algorithm :

1. Select two large prime numbers, p and q. where(p≠q)
2. Calculate n = p x q.
3. Calculate ϕ (n) = (p-1) x (q-1)
4. Choose an integer e, 1<e<ϕ, such that gcd(e,ϕ)=1.
5. Compute a value for d such that (d \* e) % ϕ (n) = 1.
6. Find c = me % n
7. Find d = cd % n



**Diffie-Hellman key Exchange algorithm**

****

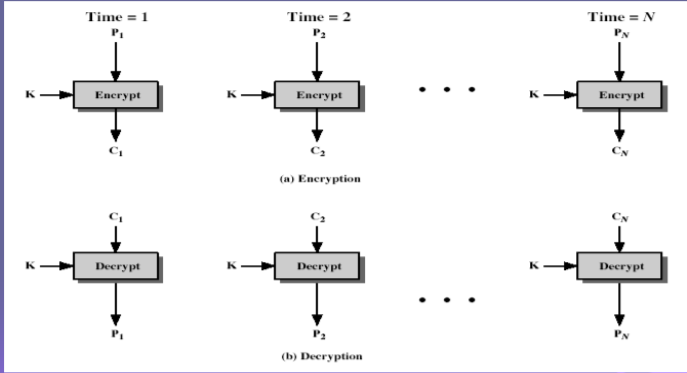
****

**Cipher Block modes of operation**

* Block ciphers encrypt fixed size blocks
* need some way to encrypt/decrypt arbitrary amounts of data in practise

**Electronic Codebook Book (ECB):**

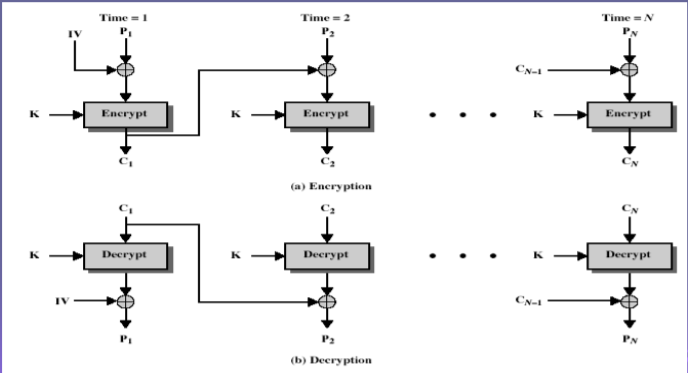
* Electronic code book is the easiest block cipher mode of functioning.
* It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext.
* uses: secure transmission of single values



* **Advantages of using ECB:**
  + Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
  + Simple way of the block cipher.
  + The other main advantage is that ECB can tolerate the loss of blocks without affecting other available blocks.
* **Disadvantages of using ECB:**
  + Identical blocks will have the same ciphers under ECB mode, which may reveal patterns the blocks have; so, ECB doesn’t wholly hide its details. This is a security threat to its users.

**Cipher Block Chaining (CBC)**

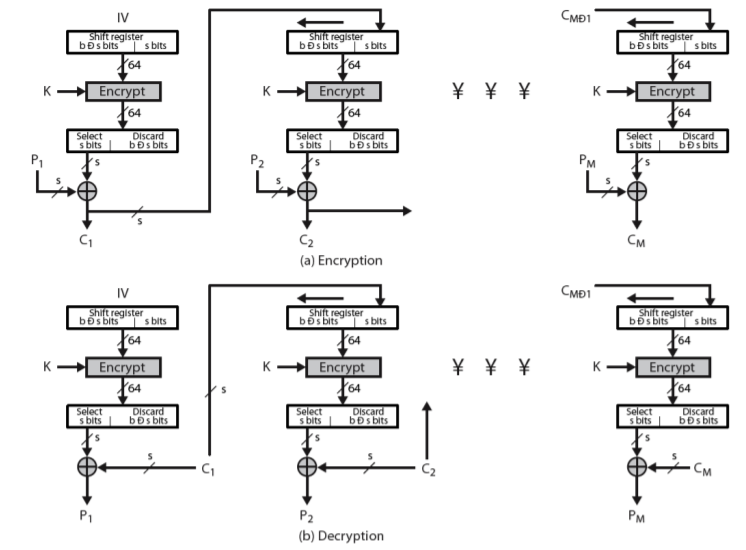
* Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements.
* In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block.



* at end of message must handle a possible last short block
* which is not as large as blocksize of cipher
* pad either with known non-data value
* or pad last block along with count of pad size
  + eg. [ b1 b2 b3 0 0 0 0 5]
  + means have 3 data bytes, then 5 bytes pad+count
* Advantages:
  + CBC works well for input greater than *b* bits.
  + CBC is a good authentication mechanism.
* Disadvantages:
  + Parallel encryption is not possible since every encryption requires a previous cipher.

**Cipher FeedBack (CFB)**

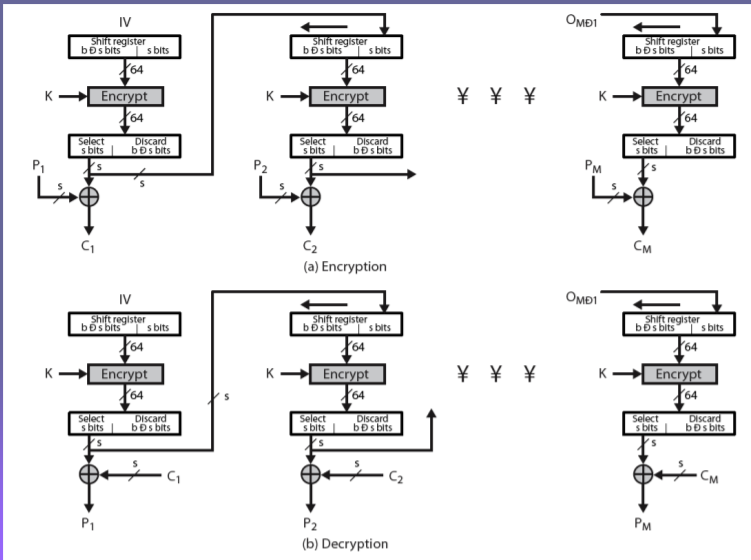
* In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first, an initial vector IV is used for first encryption and output bits are divided as a set of *s* and *b-s* bits the left-hand side *s* bits are selected and are applied an XOR operation with plaintext bits.
* The result is given as input to a shift register and the process continues. The encryption and decryption process for the same is shown below, both of them use encryption algorithms.



* most common stream mode
* errors propogate for several blocks after the error

**Output FeedBack (OFB)**

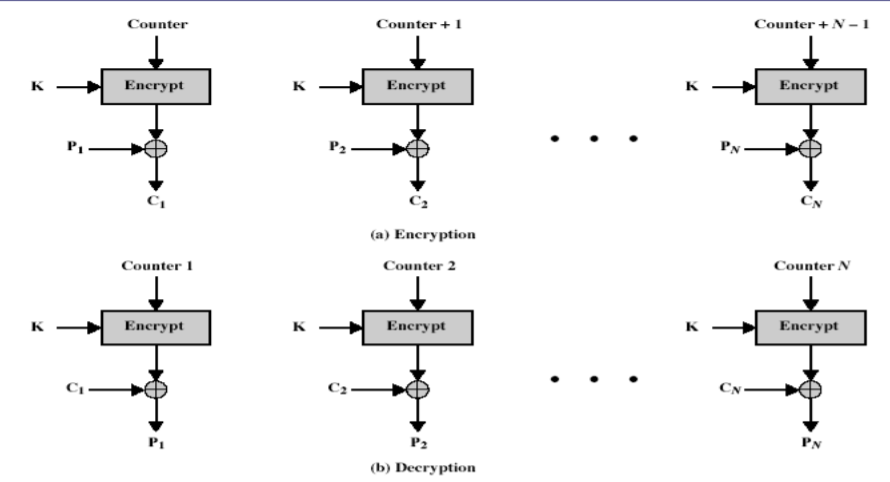
* The output feedback mode follows nearly the same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output.
* In this output feedback mode, all bits of the block are sent instead of sending selected *s* bits.



* sender & receiver must remain in sync
* originally specified with m-bit feedback

**Counter:**

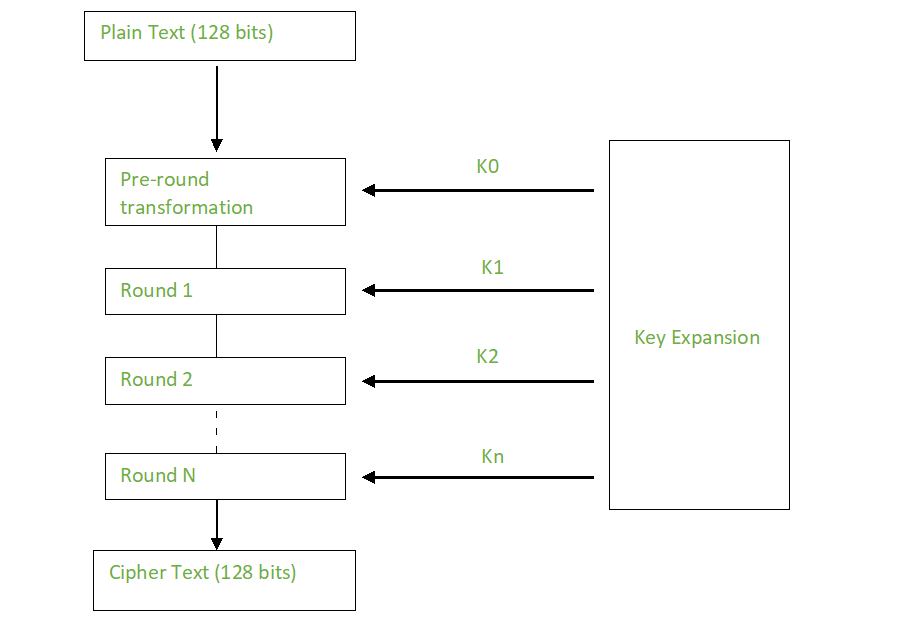
* The Counter Mode or CTR is a simple counter-based block cipher implementation.
* Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block.
* The CTR mode is independent of feedback use and thus can be implemented in parallel.

****

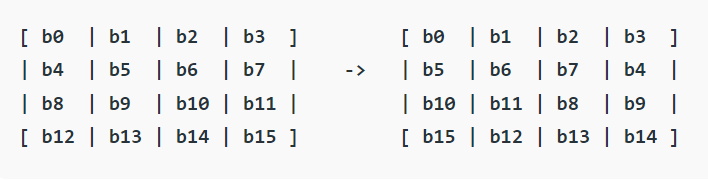
* Advantages and Limitations of CTR
* efficiency
  + can do parallel encryptions in h/w or s/w
  + can pre-process in advance of need

**AES / Rijndael**

* AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.
* It is found at least six time faster than triple DES. A replacement for DES was needed as its key size was too small.
* AES is a block cipher.
* The key size can be 128/192/256 bits.
* Encrypts data in blocks of 128 bits each.
* That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output.
* AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.



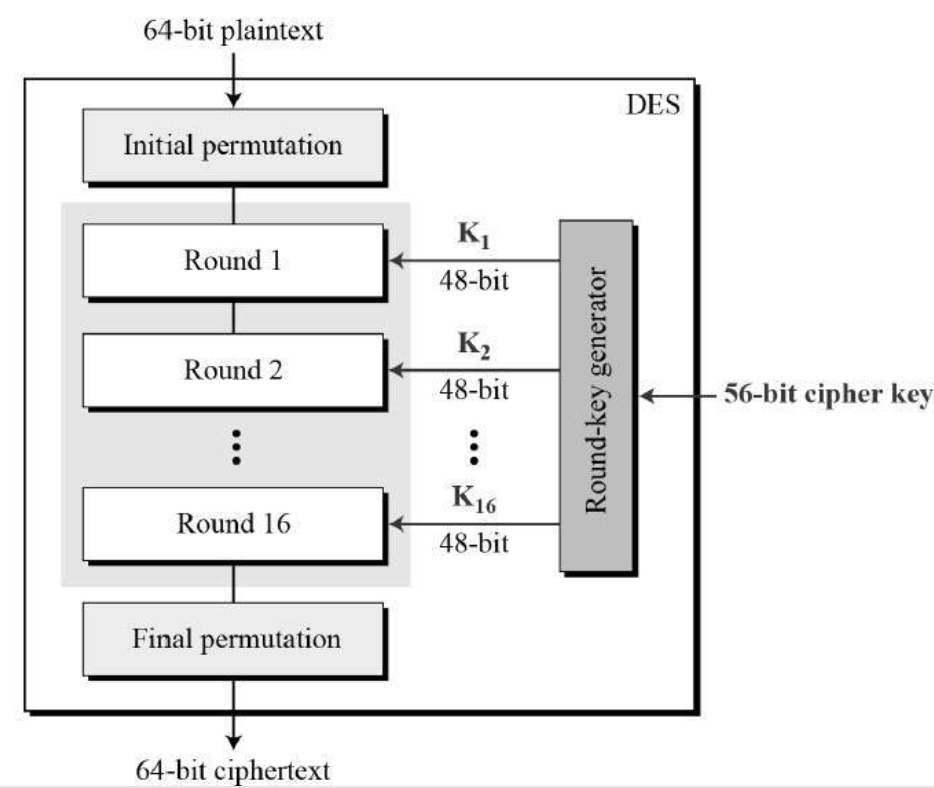
* Encryption Process
* Each round comprises of 4 steps:
  + SubBytes
  + ShiftRows
  + MixColumns
  + Add Round Key
* Byte Substitution (SubBytes)
  + In this step each byte is substituted by another byte. The result is in a matrix of four rows and four columns.
* ShiftRows
  + This step is just as it sounds. Each row is shifted a particular number of times.
    - The first row is not shifted
    - The second row is shifted once to the left.
    - The third row is shifted twice to the left.
    - The fourth row is shifted thrice to the left.



* MixColumns
  + Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.
* AddRoundKey
  + The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.
* **Decryption:**
* The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.
* The stages of each round in decryption is as follows:
  + Add round key
  + Inverse MixColumns
  + ShiftRows
  + Inverse SubByte

**DES**

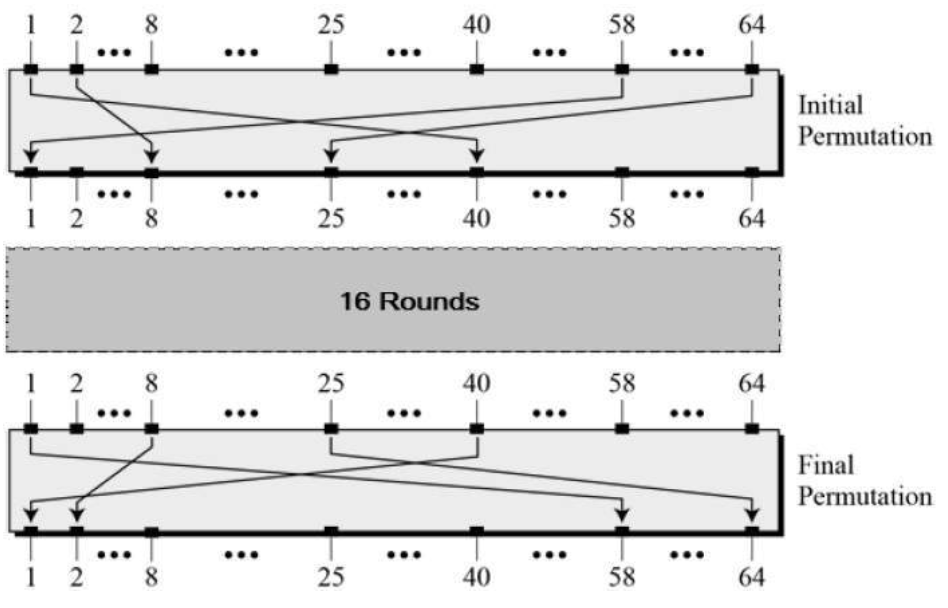
* **Data encryption standard (DES)** has been found vulnerable against very powerful attacks and therefore, the popularity of DES has been found slightly on the decline.
* DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text goes as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits. The basic idea is shown in the figure.



* We have mentioned that DES uses a 56-bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

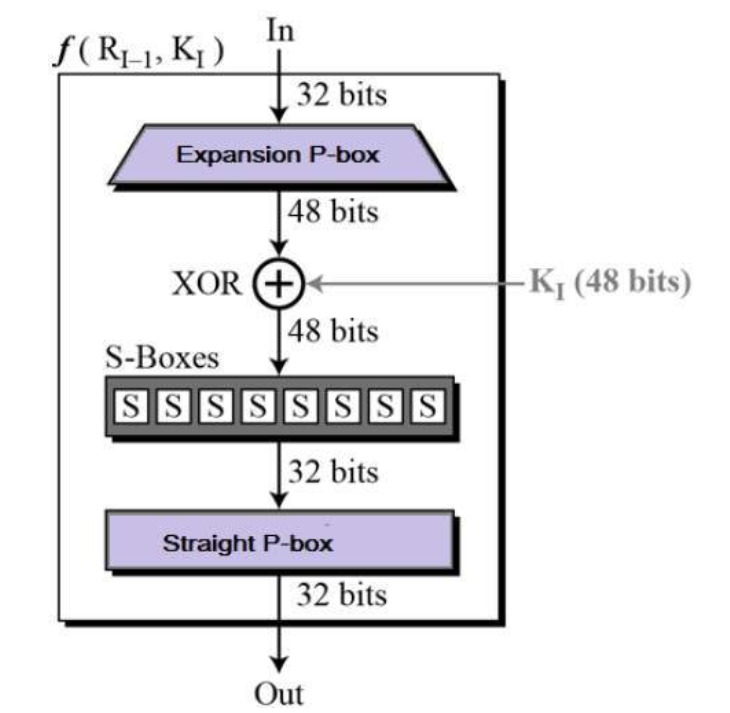
## **Initial and Final Permutation**

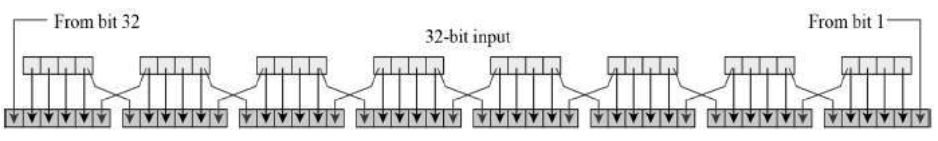
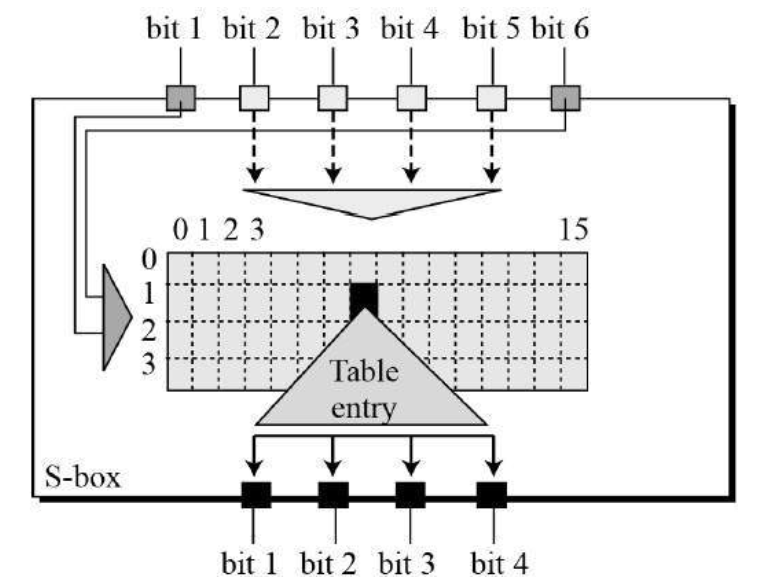
* The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES.



## **Round Function**

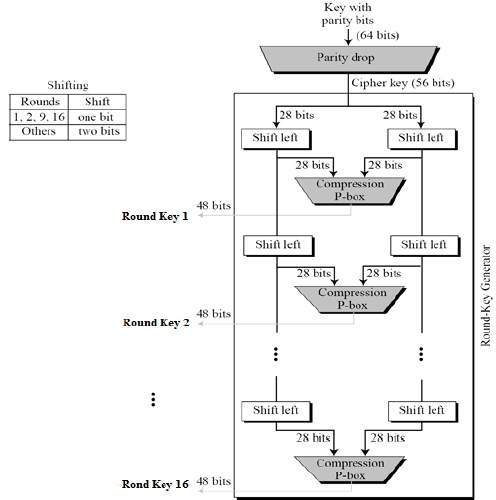
* The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



* **Expansion Permutation Box** − Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration −
* 
* **XOR (Whitener).** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.
* **Substitution Boxes.** − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.
* The S-box rule is illustrated below −
* 
* There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.

## **Key Generation**

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration −



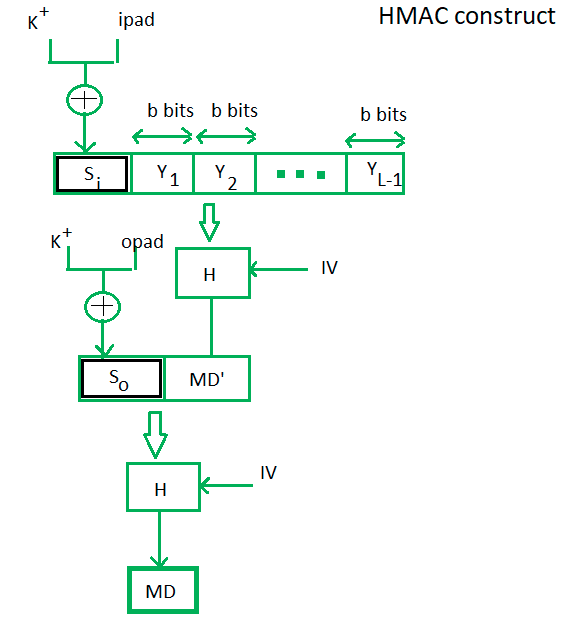
The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

**HMAC Algorithm**

* **HMAC algorithm** stands for Hashed or Hash-based [Message Authentication Code](https://www.geeksforgeeks.org/computer-network-message-authentication-code-works/).
* HMAC is a great resistance towards cryptanalysis attacks as it uses the Hashing concept twice.
* HMAC consists of twin benefits of Hashing and MAC and thus is more secure than any other authentication code.
* It aims at being less affected by collisions than the hash functions.
* HMAC reuses the algorithms like MD5 and SHA-1 and checks to replace the embedded hash functions with more secure hash functions, in case found.
* HMAC tries to handle the Keys in a more simple manner.

**Working Of HMAC algorithm:**

* HMACs provides client and server with a shared private key that is known only to them.
* The client makes a unique hash (HMAC) for every request.
* When the client requests the server, it hashes the requested data with a private key and sends it as a part of the request.
* Both the message and key are hashed in separate steps making it secure.
* When the server receives the request, it makes its own HMAC. Both the HMACS are compared and if both are equal, the client is considered legitimate.



Here, H stands for Hashing function,   
M is the original message   
Si and So are input and output signatures respectively,   
Yi is the ith block in original message M, where I ranges from [1, L)   
L = the count of blocks in M   
K is the secret key used for hashing   
IV is an initial vector (some constant)   
The generation of input signature and output signature *Si* and *So* respectively.

