CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Department of Computer Sc. and Engg

Internal Assessment Test 1 Solution– Nov 2021

Sub: Cryptography (18CS744)

Prepared By : Sagarika Behera

1.a) Explain the symmetric cipher model with a neat diagram.

Ans:

A symmetric encryption scheme has five ingredients:

• Plaintext: This is the original intelligible message or data that is fed into the algorithm as input

• Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.

• Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

• Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

• Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext

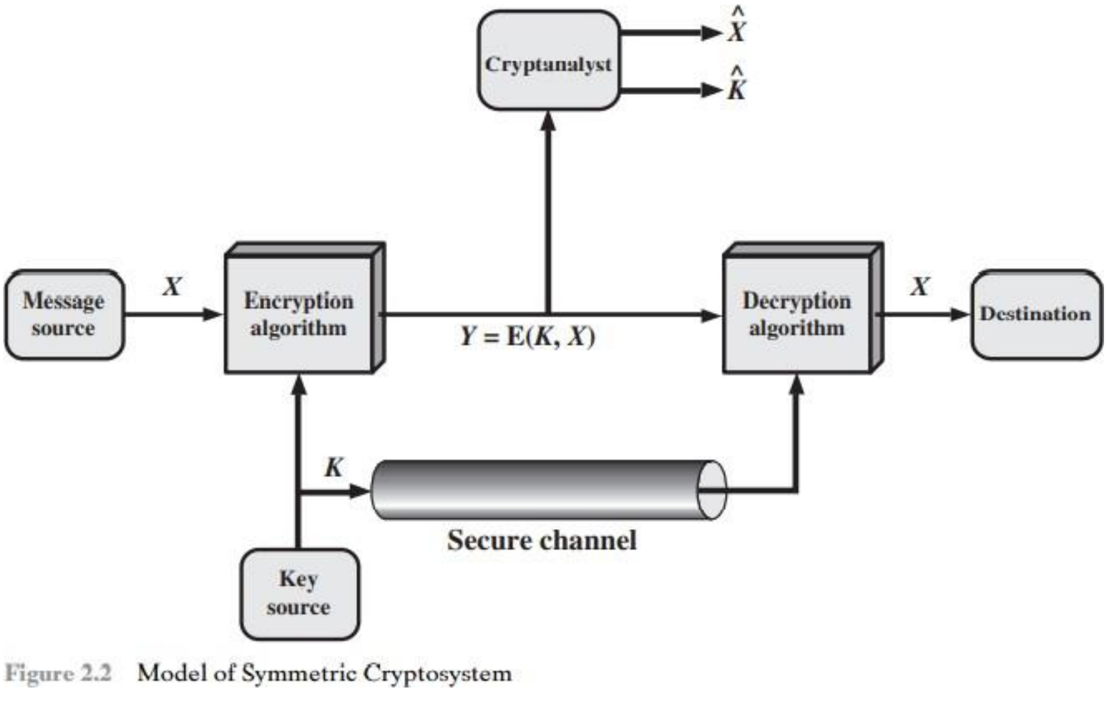


Figure 2.2 Model of Symmetric Cryptosystem

1. A source produces a message in plaintext.

2. A key of the form is generated.

3. If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel.

Alternatively, a third party could generate the key and securely deliver it to both source and destination.

The ciphertext is produced by the encryption algorithm with the message X and the encryption key K as input.
The encryption process is:

Y = E(K, X)

This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X, with the specific function determined by the value of the key K.

The intended receiver with the key is able to invert the transformation:

X = D(K, Y)

An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both. It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms. The opponent may do one of the following:

1. Recover X by generating a plaintext estimate, if the opponent is interested in only this particular message.

2. Recover K by generating an estimate, if the opponent is interested in being able to read future messages.

b) Define Trap-door one way function.

Ans:

A one-way function3 is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible:
Y = f(X) easy
X = f^{-1} (Y) infeasible.
A trap-door one-way function is easy to calculate in one direction and infeasible to calculate in the other direction unless certain additional information is known. With the additional information the inverse can be calculated in polynomial time. We can summarize as follows:
A trap-door one-way function is a family of invertible functions fk, such that
Y = f_k(X) easy, if k and X are known
X = fk^{-1} (Y) easy, if k and Y are known
X = fk^{-1} (Y) infeasible, if Y is known but k is not known
Thus, the development of a practical public-key scheme depends on discovery of a suitable trap-door one-way function.

2 a) Construct a Playfair matrix with the key **“largest”** and encrypt the following message **“meet me at college canteen”.**

l	**a**	**r**	**g**	**e**
s	**t**	**b**	**c**	**d**

f	**h**	**i/j**	**k**	**m**
n	**o**	**p**	**q**	**u**
v	**w**	**x**	**y**	**z**

Plaintext: me et me at co lx le ge ca nt ex en

Ciphertext: ud ad ud th tq rv al el tg os rz lu

b) Discuss about the requirements for public key cryptography

Ans:

Requirements for Public-Key Cryptography:

1. It is computationally easy for a party B to generate a pair (public key PUb, private key PRb).
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext: C = E(PUb, M)
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message: M = D(PRb, C) = D[PRb, E(PUb,M)]
4. It is computationally infeasible for an adversary, knowing the public key, PUb, to determine the private key, PRb
5. It is computationally infeasible for an adversary, knowing the public key, PUb, and a ciphertext, C, to recover the original message, M.

We can add a sixth requirement that, although useful, is not necessary for all public-key applications:

6. The two keys can be applied in either order: M = D[PUb, E(PRb, M)] = D[PRb, E(PUb, M)]

3. **a.** Encrypt the following plain text message using Vigenere Cipher.
Pain Text: **Hide and Protect yourself** Key: **CMRIT**

Ans:

H I D E A N D P R O T E C T Y O U R S E L F
C M R I T C M R I T C M R I T C M R I T C M

Ciphertext:

J U U M T P P G Z H V Q T B R Q G I A X N R

b) Explain Diffie-Hellman key exchange method.

Ans:

Figure below summarizes the Diffie-Hellman key exchange algorithm. For this scheme, there are two publicly known numbers: a prime number q and an integer a that is a primitive root of q. Suppose the users A and B wish to create a shared key.

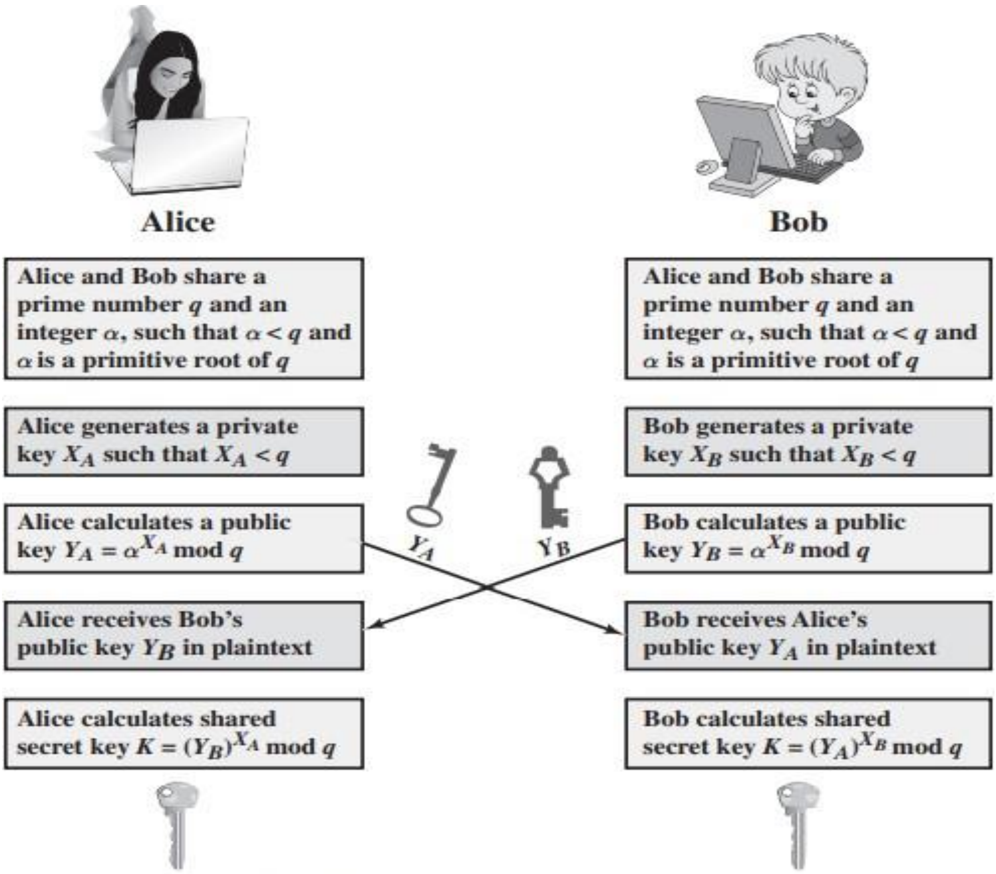
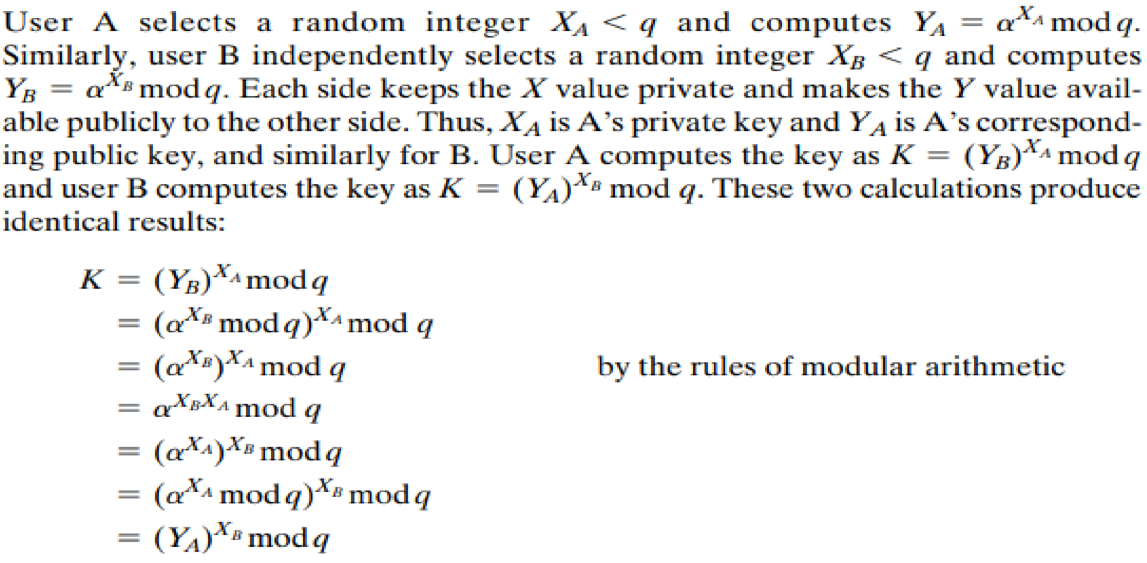

Alice
Bob
Alice and Bob share a prime number q and an integer α, such that $\alpha < q$ and α is a primitive root of q
Alice and Bob share a prime number q and an integer α, such that $\alpha < q$ and α is a primitive root of q
Alice generates a private key X_A such that $X_A < q$
Bob generates a private key X_B such that $X_B < q$
Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$
Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$
Y_A
Y_B
Alice receives Bob's public key Y_B in plaintext
Bob receives Alice's public key Y_A in plaintext
Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$
Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$

Figure 10.1 The Diffie-Hellman Key Exchange



User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \bmod q$. Similarly, user B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \bmod q$. Each side keeps the X value private and makes the Y value available publicly to the other side. Thus, X_A is A's private key and Y_A is A's corresponding public key, and similarly for B. User A computes the key as $K = (Y_B)^{X_A} \bmod q$ and user B computes the key as $K = (Y_A)^{X_B} \bmod q$. These two calculations produce identical results:

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod q \\ &= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= (\alpha^{X_B})^{X_A} \bmod q && \text{by the rules of modular arithmetic} \\ &= \alpha^{X_B X_A} \bmod q \\ &= (\alpha^{X_A})^{X_B} \bmod q \\ &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\ &= (Y_A)^{X_B} \bmod q \end{aligned}$$

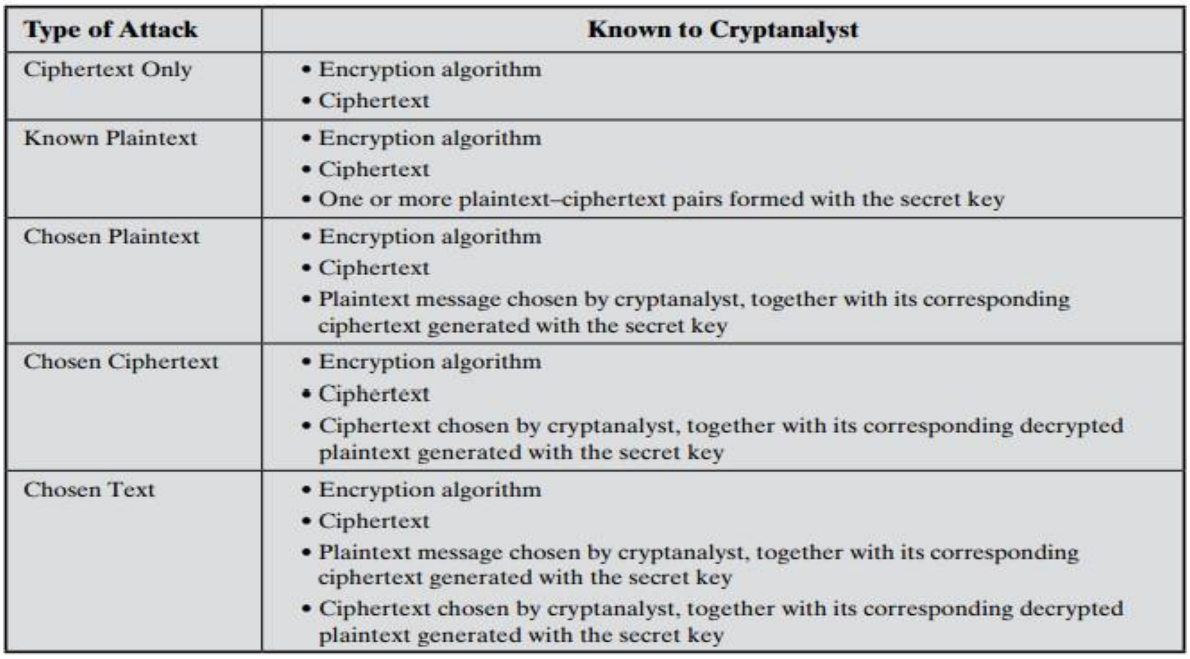
4. What are the different approaches to attack a conventional encryption scheme? Discuss the different types of attacks on encrypted messages.

Ans: Typically, the objective of attacking an encryption system is to recover the key in use rather than simplyto recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:

• Cryptanalysis: Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

• Brute-force attack: The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

Table 2.1 Types of Attacks on Encrypted Messages



Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

In general, we can assume that the opponent does know the algorithm used for encryption. If the key space is very large, the brute-force approach of trying all possible keys, which is one possible attack, becomes impractical. Thus, the opponent must analyse the ciphertext itself, applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed.

Ciphertext-only attack *

The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of information to work with.

Known-plaintext attack *

In many cases, the analyst has more information than ciphertext only:

The analyst may be able to capture one or more plaintext messages and their encryptions.

The analyst may know that certain plaintext patterns will appear in a message.

Probable-word attack *

The probable-word attack is closely related to the known-plaintext attack. If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message.

However, if the opponent is after some very specific information, then parts of the message may be known.

Chosen-plaintext attack *

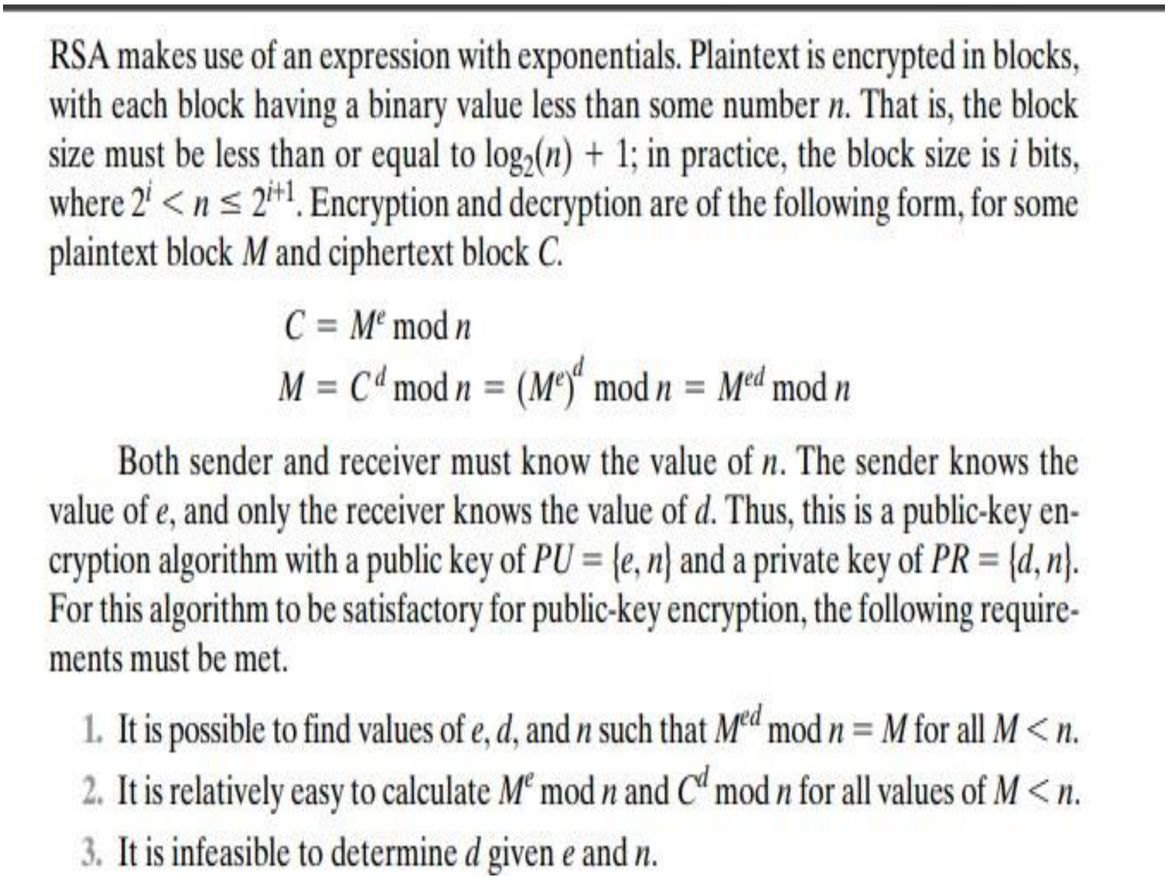
If the analyst is able to get the source system to insert into the system a message chosen by the analyst, then a chosen-plaintext attack is possible. An example of this strategy is differential cryptanalysis, explored in Chapter

The other two types of attack: chosen ciphertext and chosen text, are less commonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack.

Generally, an encryption algorithm is designed to withstand a known-plaintext attack; only weak algorithms fail to withstand a ciphertext-only attack.

5. a) Explain encryption and decryption process of RSA algorithm.

Ans:



RSA makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. That is, the block size must be less than or equal to $\log_2(n) + 1$; in practice, the block size is i bits, where $2^i < n \leq 2^{i+1}$. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C.

$$C = M^e \bmod n$$
$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met.

1. It is possible to find values of e, d, and n such that $M^{ed} \bmod n = M$ for all $M < n$.
2. It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$.
3. It is infeasible to determine d given e and n.

For now, we focus on the first requirement and consider the other questions later. We need to find a relationship of the form

$$M^{ed} \bmod n = M$$

The preceding relationship holds if e and d are multiplicative inverses modulo $\phi(n)$, where $\phi(n)$ is the Euler totient function. It is shown in Chapter 8 that for p, q prime, $\phi(pq) = (p - 1)(q - 1)$. The relationship between e and d can be expressed as

$$ed \bmod \phi(n) = 1 \tag{9.1}$$

This is equivalent to saying

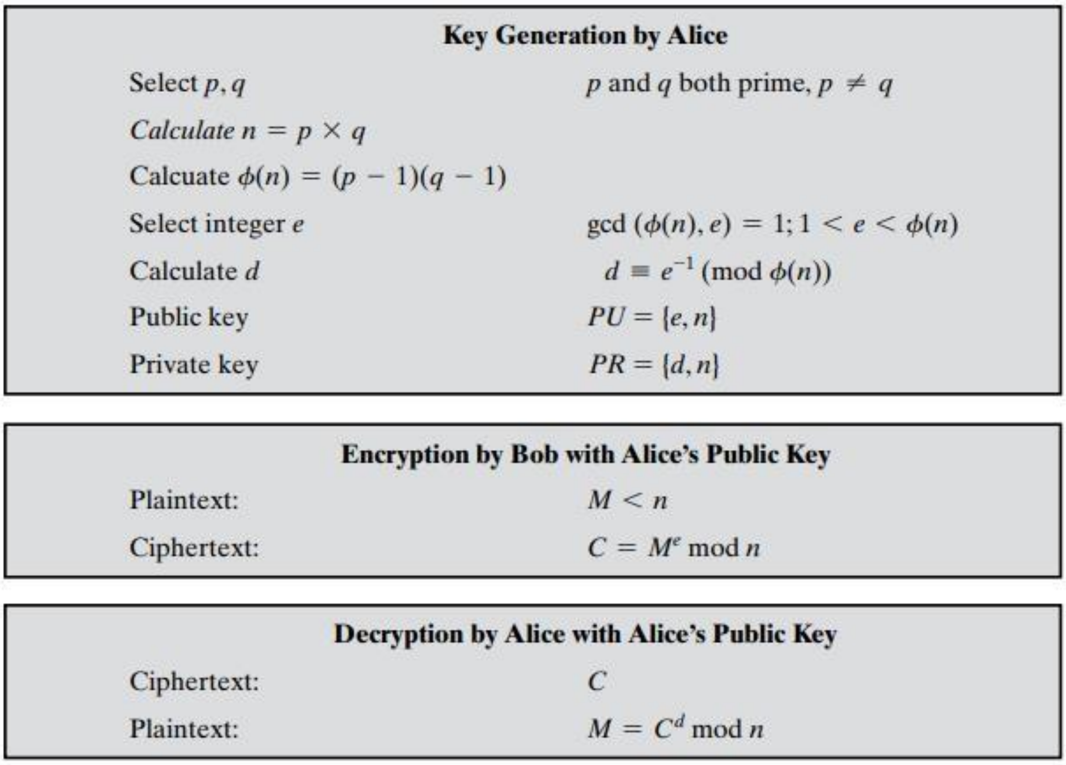
$$ed \equiv 1 \bmod \phi(n)$$
$$d \equiv e^{-1} \bmod \phi(n)$$

That is, e and d are multiplicative inverses mod $\phi(n)$. Note that, according to the rules of modular arithmetic, this is true only if d (and therefore e) is relatively prime to $\phi(n)$. Equivalently, $\gcd(\phi(n), d) = 1$. See Appendix R for a proof that Equation (9.1) satisfies the requirement for RSA.

We are now ready to state the RSA scheme. The ingredients are the following:

p, q, two prime numbers	(private, chosen)
$n = pq$	(public, calculated)
e, with $\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$	(public, chosen)
$d \equiv e^{-1} \pmod{\phi(n)}$	(private, calculated)

The private key consists of $\{d, n\}$ and the public key consists of $\{e, n\}$. Suppose that user A has published its public key and that user B wishes to send the message M to A. Then B calculates $C = M^e \bmod n$ and transmits C. On receipt of this ciphertext, user A decrypts by calculating $M = C^d \bmod n$.



Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} (\bmod\ \phi(n))$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Figure 9.5 The RSA Algorithm

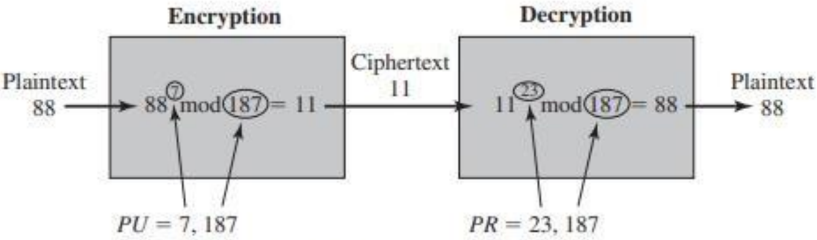


Figure 9.6 Example of RSA Algorithm

5.b) Perform encryption and decryption using RSA algorithm. Where two prime numbers are p=11, q=13, public key e=11 and plain text M=7.

Ans:

n = p*q = 143

Ø (n) =(p-1)*(q-1) = 120

d= e^{-1}mod Ø (n) = 11

Ciphertext C = M^{e} mod n = 106
Plaintext M = C^{d} mod n = 7

6. **What is stream cipher and block cipher? Explain DES scheme with a neat block diagram.**

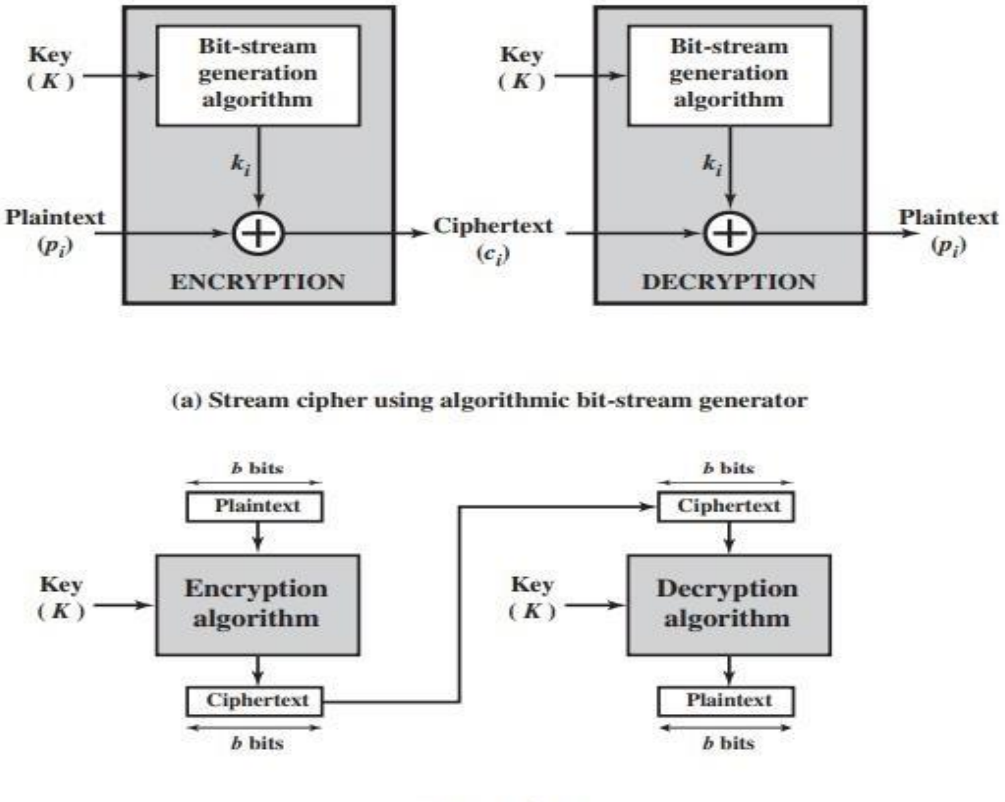
Ans:

Steam Cipher: A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.

Examples of classical stream ciphers are the auto keyed Vigenère cipher and the Vernam cipher.

Block Cipher: A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used.

As with a stream cipher, the two users share a symmetric encryption key. Using some of the modes of operation, a block cipher can be used to achieve the same effect as a stream cipher.



(a) Stream cipher using algorithmic bit-stream generator

(b) Block cipher

Figure 3.1 Stream Cipher and Block Cipher

Data Encryption Standard (DES) was issued in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST). The algorithm itself is referred to as the Data Encryption Algorithm (DEA).7 For DEA, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

The overall scheme for DES encryption is illustrated in the figure below:

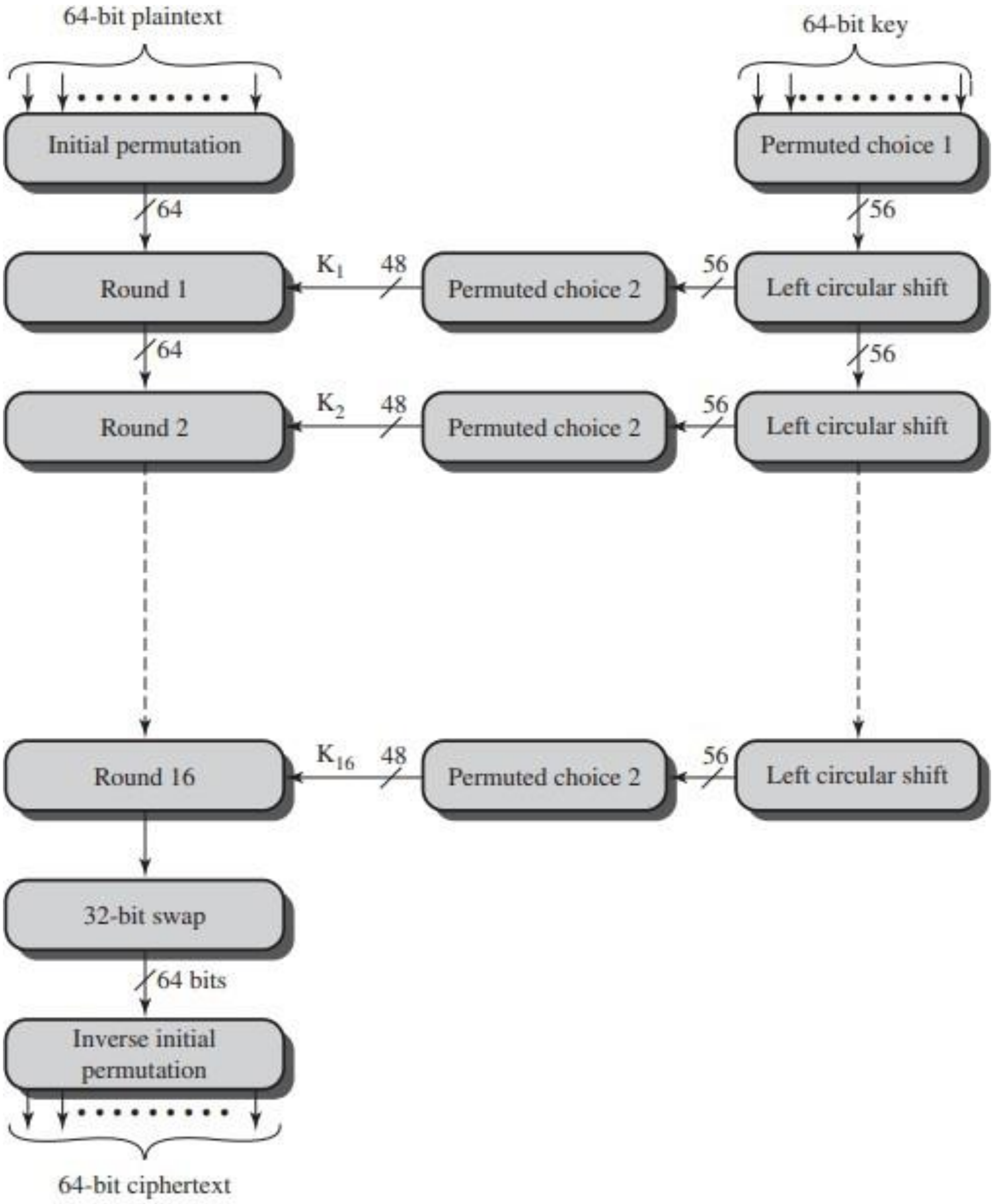


Figure 3.5 General Depiction of DES Encryption Algorithm

1. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

2. Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases.

3. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input.

4. This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.

5. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key.

6. The left and right halves of the output are swapped to produce the pre output. Finally, the pre output is passed through a permutation [IP -] that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

7. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher, as shown in Figure.

8. The right-hand portion of 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. T

9. The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

10. As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of the subkeys is reversed. Additionally, the initial and final permutations are reversed.

7. Using Hill Cipher technique encrypt and decrypt the plain text "crypto" using the key K = $\begin{pmatrix}\mathbf{7} & \mathbf{8}\\ \mathbf{11} & \mathbf{11}\end{pmatrix}$

Ans:

Plain Text: C R Y P T O
2 17 24 15 19 14

$\begin{pmatrix}\mathbf{7} & \mathbf{8}\\ \mathbf{11} & \mathbf{11}\end{pmatrix}\begin{pmatrix}\mathbf{2}\\ \mathbf{17}\end{pmatrix}$ **mod 26 =** $\begin{pmatrix}\mathbf{20}\\ \mathbf{1}\end{pmatrix}$ = $\begin{pmatrix}\boldsymbol{U}\\ \boldsymbol{B}\end{pmatrix}$

So ciphertext for crypto is " ubcnlz"

| K | = 77-88= -11 mod 26 = 15 mod 26

Co-factor of A is $\begin{pmatrix}\mathbf{11} & \mathbf{-11}\\ \mathbf{-8} & \mathbf{7}\end{pmatrix}$

Adj A is $\begin{pmatrix}11 & -8\\ -11 & 7\end{pmatrix}$

Inverse of K is

K^{-1} = 1/15 $\begin{pmatrix}11 & -8\\ -11 & 7\end{pmatrix}$ mod 26 = $\begin{pmatrix}77 & -56\\ -77 & 49\end{pmatrix}$bmod 26 = $\begin{pmatrix}25 & -4\\ -25 & 23\end{pmatrix}$ mod 26

Plain text P = CK^{-1}