

Terna Engineering College  
Computer Engineering Department

Class: TE

Sem.: VI

Course: System Security Lab

PART A

(PART A : TO BE REFERRED BY STUDENTS)

Experiment No.01

A.1 Aim:


Design and Implementation of a product cipher using Substitution and Transposition ciphers

A.2 Prerequisite:

1. Basic Knowledge of Cryptography.
2. Knowledge of Substitution Cipher techniques.

A.3 Outcome:

After successful completion of this experiment students will be able to apply the knowledge of symmetric cryptography to implement simple ciphers. A.4 Theory:

 **Cryptography** is art of achieving security by encoding message to make it non readable.

There are two types of cryptographic algorithms:

☐ Substitution and ☐

Transposition.

Product cipher is combination of both these types to achieve better effect of security.

An original message is known as the plaintext, while the coded message is called the cipher text.

The process of converting from plaintext to cipher text is known as enciphering or encryption; restoring the plaintext from the cipher text is deciphering or decryption.

The many schemes used for encryption constitute the area of study known as cryptography. Such a scheme is known as a cryptographic system or a cipher.

Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of cryptanalysis.

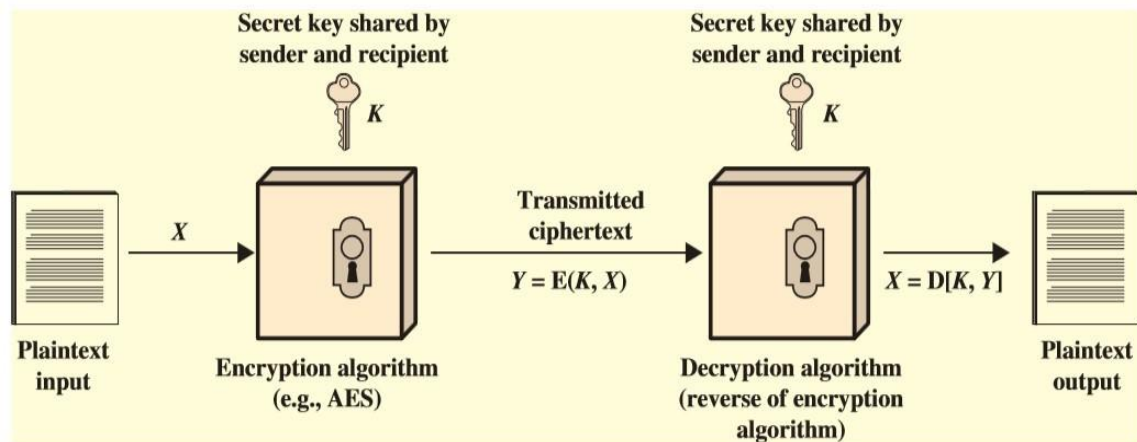
Cryptanalysis is what the layperson calls “breaking the code.” The areas of cryptography and cryptanalysis together are called cryptology.

- Cryptography is art of achieving security by encoding message to make it non readable. There are two types of cryptographic algorithms: Substitution and Transposition. Product cipher is combination of both these types to achieve better effect of security.
- Substitution Cipher: Additive or Shift or Caesar Cipher algorithm is cryptographic algorithm invented by Caesar. It is substitution based algorithm.

## Symmetric Cipher Model

A symmetric encryption scheme has five ingredients (Figure 2.1):

- 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 2.1** Simplified Model of Symmetric Encryption

There are two requirements for secure use of conventional encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
1. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

We assume that it is impractical to decrypt a message on the basis of the ciphertext plus knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret.

This feature of symmetric encryption is what makes it feasible for widespread use. The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms.

These chips are widely available and incorporated into a number of products. With the use of symmetric encryption, the principal security problem is maintaining the secrecy of the key.

Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 2.2. A source produces a message in plaintext,  $X = [X_1, X_2, X_3, \dots, X_m]$ .

The  $M$  elements of  $X$  are letters in some finite alphabet. Traditionally, the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet  $\{0, 1\}$  is typically used.

For encryption, a key of the form  $K=[K_1, K_2, K_3, \dots, K_j]$  is generated. 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.

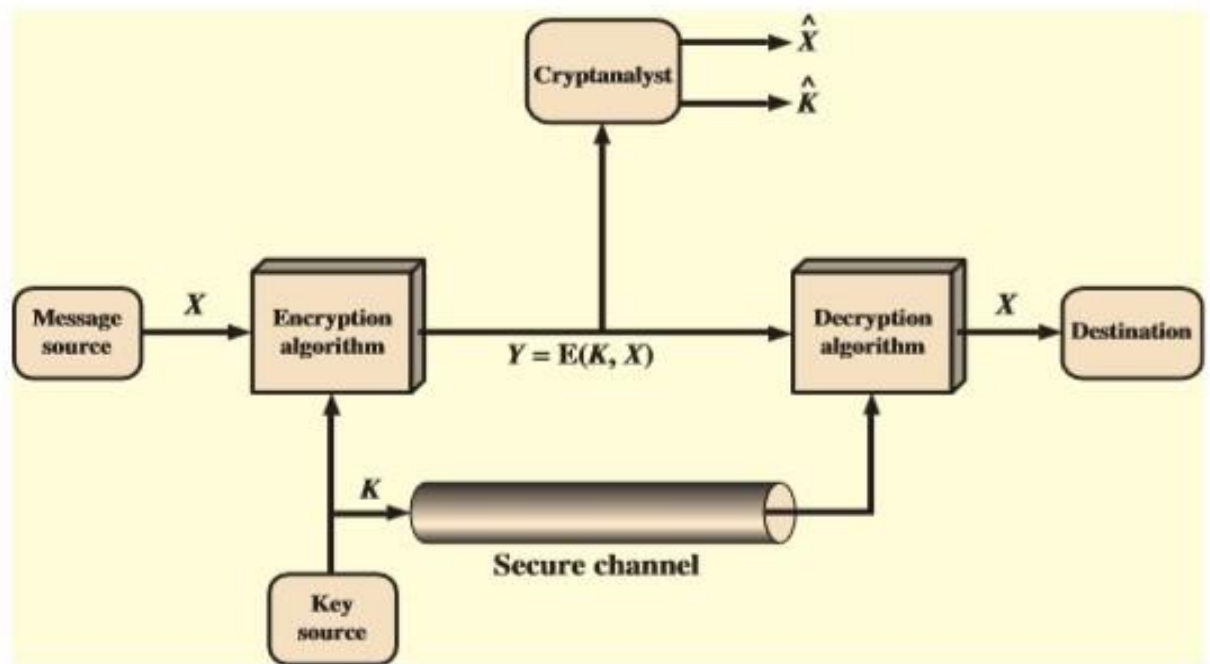


Figure 2.2 Model of Symmetric Cryptosystem

With the message and the encryption key  $K$  as input, the encryption algorithm forms the ciphertext  $Y=[Y_1, Y_2, Y_3, \dots, Y_n]$ . We can write this as;

$$Y=E(K,X)$$

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, in possession of 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  $X$  and  $K$ . It is assumed that the opponent knows the encryption ( $E$ ) and

decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of the effort is to recover  $X$  by generating a plaintext estimate  $\hat{X}$ . Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover  $K$  by generating an estimate  $\hat{K}$ .

### A.5 EXAMPLE:

1. Plaintext: CryPto(Sender's input)  
Key: 5(agreement key, here input)  
Cipher text:  $(C+5)(r+5)(y+5)(P+5)(t+5)(o+5)$   
: HwdUyt(output)
2. Cipher Text: XdkcZm(Receiver's Input)  
Key: 5(agreement key, here input)  
Plaintext: CiphEr (output)

## A.6 ALGORITHM:

@Encryption....

- A - Substitution Cipher
1. Take Value of Plaintext.
  2. Take Value of Key.
  3. Run the loop n times here n will be the length of the text.
  4. Convert string into characters and that same time add with Key.
  5. After Addition mod 26 or 125 can be used depending where you want to limit that entry of plain text.
  6. Then save and convert the calculated part (character) into a string or string Buffer.
  7. Now the Final output will be the actual Cipher Text.

## B - Transposition Cipher

1. Input of this Cipher will be the output from the Substitution Cipher.
2. Convert into matrix with the help of the key (key is different these time).
3. And will the help of the matrix, just lay down the content of the matrix or save the content into a simple string.
4. This String will be the Cipher Text.

@Decryption.....

Now out steps will be in the reverse order.

Hence,

#### A - Transposition Cipher-

1. Input here will be the cipher text.
2. Take Value of Key.
3. Insert this Cipher Text into the Transposition Cipher whole algorithm remains same.  
It just the value of k will change ( $k = \text{cipher text}/k$ ).
4. Collect the characters and built it into a string.
5. That String will act as text that will be later interested into Substitution for the final decryption.

#### B - Substitution Cipher

1. After taking the input.
2. Use the same key
3. Follow the Same producer as encryption, here the only difference is that instead of addition we will perform subtraction with the help of the key.
4. The result will be the Plain Text. And the code is finally decrypted.

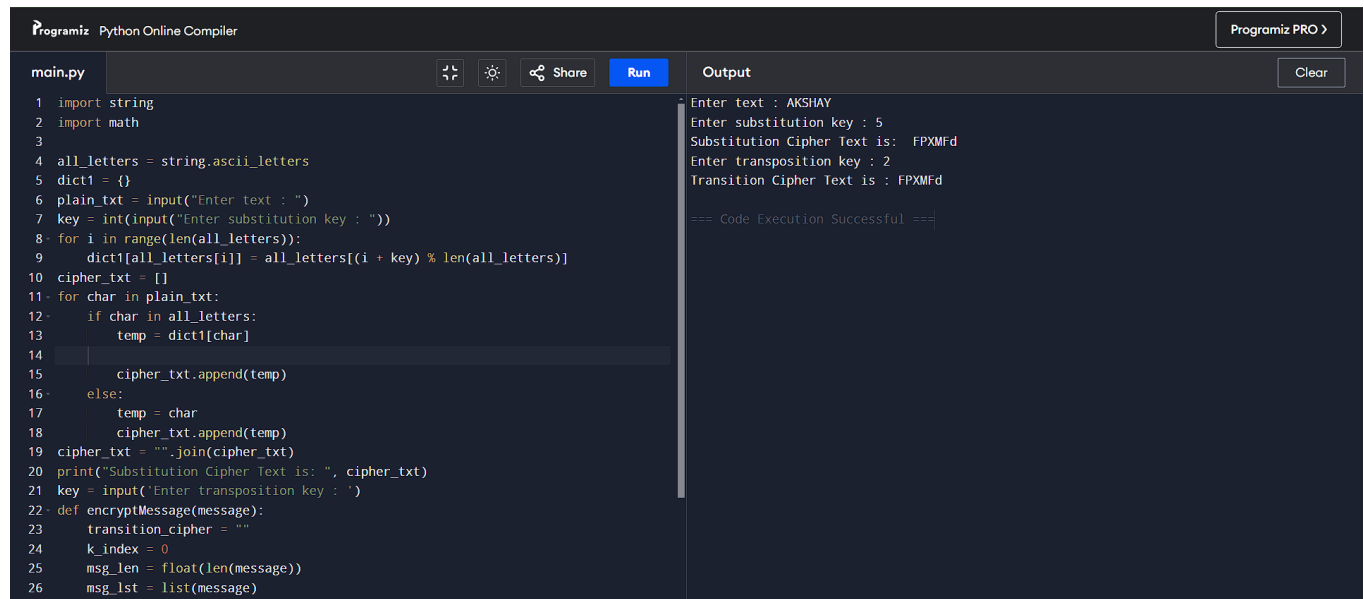
## PART B

### (PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No. 31	Name: Diwate Akshay Arun
Class : TE	Batch : B2
Date of Experiment:	Date of Submission
Grade :	

### B.1 Output of Substitution and Transposition Cipher:



The screenshot shows a Python Online Compiler interface with a dark theme. The code editor on the left contains a script named 'main.py' that implements a Substitution Cipher and a Transposition Cipher. The script takes user input for a plain text, a substitution key, and a transposition key, and outputs the corresponding cipher texts. The output window on the right shows the execution results for the input 'AKSHAY', substitution key '5', and transposition key '2'. The substitution cipher text is 'FPXMFd' and the transposition cipher text is also 'FPXMFd'. The execution was successful.

```
1 import string
2 import math
3
4 all_letters = string.ascii_letters
5 dict1 = {}
6 plain_txt = input("Enter text : ")
7 key = int(input("Enter substitution key : "))
8 for i in range(len(all_letters)):
9     dict1[all_letters[i]] = all_letters[(i + key) % len(all_letters)]
10 cipher_txt = []
11 for char in plain_txt:
12     if char in all_letters:
13         temp = dict1[char]
14
15         cipher_txt.append(temp)
16     else:
17         temp = char
18         cipher_txt.append(temp)
19 cipher_txt = "".join(cipher_txt)
20 print("Substitution Cipher Text is: ", cipher_txt)
21 key = int(input("Enter transposition key : "))
22 def encryptMessage(message):
23     transition_cipher = ""
24     k_index = 0
25     msg_len = float(len(message))
26     msg_lst = list(message)
```

Output:

```
Enter text : AKSHAY
Enter substitution key : 5
Substitution Cipher Text is: FPXMFd
Enter transposition key : 2
Transition Cipher Text is : FPXMFd

=== Code Execution Successful ===
```

### B.2. Source Code of Substitution and Transposition ciphers:

```
import string
import math

all_letters = string.ascii_letters
dict1 = {}
plain_txt = input("Enter text : ")
key = int(input("Enter substitution key : "))
for i in range(len(all_letters)):
```

```

    dict1[all_letters[i]] = all_letters[(i + key) % len(all_letters)]
cipher_txt = []
for char in plain_txt:
    if char in all_letters:
        temp = dict1[char]

        cipher_txt.append(temp)
    else:
        temp = char
        cipher_txt.append(temp)
cipher_txt = "".join(cipher_txt)
print("Substitution Cipher Text is: ", cipher_txt)
key = input('Enter transposition key : ')
def encryptMessage(message):
    transition_cipher = ""
    k_index = 0
    msg_len = float(len(message))
    msg_lst = list(message)
    key_lst = sorted(list(key))
    col = len(key)
    row = int(math.ceil(msg_len / col))
    fill_null = int((row * col) - msg_len)
    msg_lst.extend('_' * fill_null)
    matrix = [msg_lst[i: i + col] for i in range(0, len(msg_lst), col)]
    for _ in range(col):
        curr_idx = key.index(key_lst[k_index])
        transition_cipher += ".join([row[curr_idx] for row in matrix])
        k_index += 1
    return transition_cipher

msg = cipher_txt
cipher = encryptMessage(msg)
print("Transition Cipher Text is : {}".format(cipher))

```

### B.3 Question of Curiosity: (At least 3 questions should be handwritten)

1. What are the Ciphers?
  
2. What are the different types of Cipher?



Name:- Akshay Divak

TE-B2 31

CSS Expt 1

Page No.

Date

1) What are the ciphers?

Ans Ciphers are made methods used to change a message so that only certain people can understand it. They work by scrambling the original message (called plaintext) into a secret form (called ciphertext) using a set of rules or keys. The person who has the key can then unscramble it back into the original message. Ciphers are used to keep information private and secure.

2) What are the different types of cipher?

Ans There are several types of cipher, but here are the main

1. Substitution cipher: Each letter in the cipher is replaced with another letter or symbol.  
(Example Caesar cipher)

2. Transposition cipher: The letters in the message are rearranged or shuffled around, but the letters themselves stay the same.

3. Stream cipher: Encrypt the message one bit or a character at a time usually using a key. It's often used for real-time communication.

4. Block cipher: Encrypt the message in fixed-size blocks of data, rather than one character at a time.

5. Affine cipher: A type of substitution cipher where each letter is mapped to its numeric equivalent modified by a mathematical formula.

3. What is Substitution Cipher? Explain with an example.

characters is substituted by some other character from the same set depending on a key. For example with a shift of 1, A would be replaced by B, B would become C, and so on.

Examples:

Plain Text: I am studying Data

Encryption Key: 4

Output: M eq wxyhCmrk Hexe IrgvCtxmsr

4. What is transposition Cipher? Explain with an example.

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher text.

1. The message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order.

2. Width of the rows and the permutation of the columns are usually defined by a keyword.

3. For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be “3 1 2 4”.

4. Any spare spaces are filled with nulls or left blank or placed by a character

5. Finally, the message is read off in columns, in the order specified by the keyword

**Encryption**

**Given text** = Geeks for Geeks  
**Keyword** = HACK      **Length of Keyword** = 4 (no of rows)      **Order of Alphabets in HACK** = 3 1 2 4

H	A	C	K
3	1	2	4
G	e	e	k
s	_	f	o
r	_	G	e
e	k	s	_

Print Characters of column 1,2,3,4  
**Encrypted Text** = e kefgsGreko\_e\_

5. Explain Affine Cipher.

The Affine cipher is a type of monoalphabetic substitution cipher, wherein each letter in an alphabet is mapped to its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter.

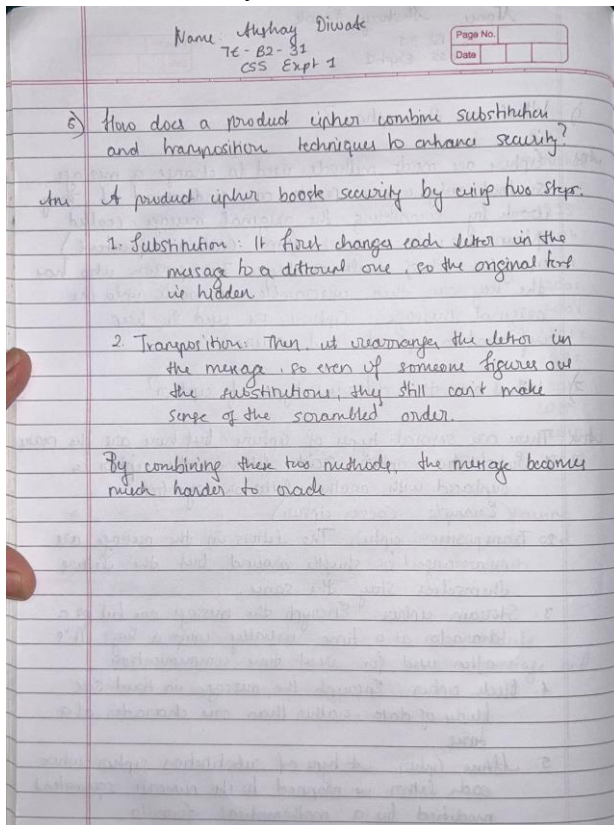
The formula used means that each letter encrypts to one other letter, and back again, meaning the cipher is essentially a standard substitution cipher with a rule governing which letter goes to which. The whole process relies on working modulo m (the length of the alphabet used).

In the affine cipher, the letters of an alphabet of size  $m$  are first mapped to the integers in the range  $0 \dots m-1$ .

The 'key' for the Affine cipher consists of 2 numbers, we'll call them  $a$  and  $b$ . The following discussion assumes the use of a 26 character alphabet ( $m = 26$ ).  $a$  should be chosen to be relatively prime to  $m$  (i.e.  $a$  should have no factors in common with  $m$ ).

It uses modular arithmetic to transform the integer that each plaintext letter corresponds to into another integer that correspond to a ciphertext letter.

6. How does a product cipher combine substitution and transposition techniques to enhance security?



#### B.4 Conclusion:

In conclusion, the design and implementation of a product cipher using substitution and transposition techniques provides a secure method of encryption. By combining these two techniques, it enhances the overall security of the message, making it more difficult to break. Substitution hides the message content, while transposition shuffles the order of the characters, adding an extra layer of protection. This combination ensures that even if one technique is compromised, the other still provides a level of security.

