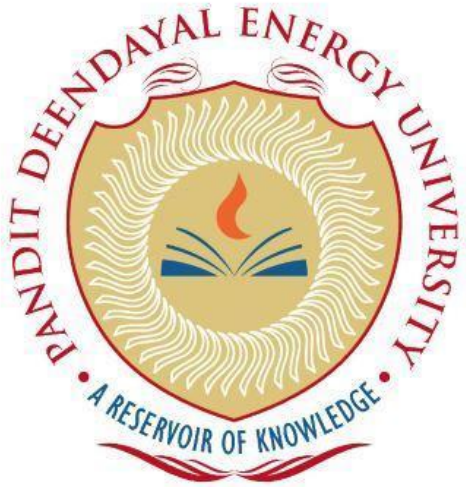
PANDIT DEENDAYAL ENERGY UNIVERSITY

SCHOOL OF TECHNOLOGY



Course: Information Security Lab

Course Code: 20CP304P

LAB MANUAL

Submitted To:

Submitted By:
NAME:
ROLL NO:

PANDIT DEENDAYAL ENERGY UNIVERSITY

Raisan, Gandhinagar – 380 007, Gujarat, India

Computer Science Engineering Department

Certificate

This is to certify that

Mr./Ms. _____ Roll no._____

Exam No. _____ of 5th Semester Degree course in Computer Science and Engineering has satisfactorily completed his/her term work in Information Security Lab (20CP304P) subject during the semester from_____ to _____ at School of Technology, PDEU.

Date of Submission:
Signature:

Faculty In-charge **Head of Department**

INDEX

S. No.	List of experiments	Date	Sign
1	Study and Implement a program for Caesar Cipher		
2	Study and Implement a program for 5x5 Playfair Cipher to encrypt and decrypt the message.		
3	Study and Implement a program for Rail Fence Cipher		
4	Study and implement a program for columnar Transposition Cipher		
5	Study and implement a program for Vigenère Cipher		
6	Study and Implement a program for n-gram Hill Cipher		
7	Study and Use of RSA algorithm (encryption and decryption)		
8	Study and implement a program of the Digital Signature with RSA algorithm (Reverse RSA)		
9	Study and Use of Diffie-Hellman Key Exchange		
10	Design cipher with detailed explanation. Sample as follows. a) Write a program to encrypt the plaintext with the given key. E.g. plaintext GRONSFELD with the key 1234. Add 1 to G to get H (the letter 1 rank after G is H in the alphabet), then add 2 to C or E (the letter 2 ranks after C is E), and so on. Use smallest letter from plaintext as filler. b) Encrypt the input words PLAINTEXT= RAG BABY to obtain CIPHERTEXT = SCJ DDFD		
11	Mini project (Need for Post Quantum Cryptography)		

For each experiment, students need to perform their variation of the standard algorithm mentioned as the AIM of the experiment and perform the ***Security Analysis,*** *where the student must perform a security analysis of the algorithm mentioned above. Analysis may include a) Cryptanalysis, b)Performance analysis*

Experiment 1

AIM: Study and Implement a program for Caesar Cipher

Introduction : In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code, or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

Example : The transformation can be represented by aligning two alphabets; the cipher alphabet is the plain alphabet rotated left or right by some number of positions. For instance, here is a Caesar cipher using a left rotation of three places, equivalent to a right shift of 23 (the shift parameter is used as the key):

Plain	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

When encrypting, a person looks up each letter of the message in the "plain" line and writes down the corresponding letter in the "cipher" line.

Plaintext: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD

Deciphering is done in reverse, with a right shift of 3.

The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A → 0, B → 1, ..., Z → 25.

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 2

AIM: Study and Implement a program for 5x5 Playfair Cipher to encrypt and decrypt the message.

Introduction : The Playfair cipher or Playfair square or Wheatstone–Playfair cipher is a manual symmetric encryption technique and was the first literal digram substitution cipher. The scheme was invented in 1854 by Charles Wheatstone, but bears the name of Lord Playfair for promoting its use.

The technique encrypts pairs of letters (*bigrams* or *digrams*), instead of single letters as in the simple substitution cipher and rather more complex Vigenère cipher systems then in use. The Playfair cipher is thus significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it. The frequency analysis of bigrams is possible, but considerably more difficult.

Example : For the encryption process let us consider the following example:

Key: monarchy
Plaintext: instruments

The Playfair Cipher Encryption Algorithm:
The Algorithm consists of 2 steps:

1. **Generate the key Square(5×5):**
   - The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.
   - The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
2. **Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter. **For example:**

PlainText: "instruments"
After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz

Plain Text: "instrumentsz"
Encrypted Text: gatlmzclrqtx

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 3

AIM: Study and Implement a program for Rail Fence Cipher with columnar transposition

Introduction : The rail fence cipher (also called a zigzag cipher) is a classical type of transposition cipher. It derives its name from the manner in which encryption is performed, in analogy to a fence built with horizontal rails.

Example : In the rail fence cipher, the plaintext is written downwards diagonally on successive "rails" of an imaginary fence, then moving up when the bottom rail is reached, down again when the top rail is reached, and so on until the whole plaintext is written out. The ciphertext is then read off in rows.

For example, to encrypt the message 'WE ARE DISCOVERED. RUN AT ONCE.' with 3 "rails", write the text as:

W . . . E . . . C . . . R . . . U . . . O . . .
. E . R . D . S . O . E . E . R . N . T . N . E
. . A . . . I . . . V . . . D . . . A . . . C .

(Note that spaces and punctuation are omitted.) Then read off the text horizontally to get the ciphertext:

WECRUO ERDSOEERNTNE AIVDAC

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 4

AIM: Study and implement a program for columnar Transposition Cipher

Introduction : The columnar transposition cipher is a fairly simple, easy to implement cipher. It is a transposition cipher that follows a simple rule for mixing up the characters in the plaintext to form the ciphertext. Although weak on its own, it can be combined with other ciphers, such as a substitution cipher, the combination of which can be more difficult to break than either cipher on it's own.

Example : The key for the columnar transposition cipher is a keyword e.g. GERMAN. The row length that is used is the same as the length of the keyword. To encrypt a piece of text, e.g.

```
defend the east wall of the castle
```

we write it out in a special way in a number of rows (the keyword here is GERMAN):

```
G E R M A N
d e f e n d
t h e e a s
t w a l l o
f t h e c a
s t l e x x
```

In the above example, the plaintext has been padded so that it neatly fits in a rectangle. This is known as a regular columnar transposition. An irregular columnar transposition leaves these characters blank, though this makes decryption slightly more difficult. The columns are now reordered such that the letters in the key word are ordered alphabetically.

```
A E G M N R
n e d e d f
a h t e s e
l w t l o a
c t f e a h
x t s e x l
```

The ciphertext is read off along the columns:

```
nalcxehwttdttfseeleedsoaxfeahl
```

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 5

AIM: Study and implement a program for Vigenère Cipher

Introduction : The Vigenère cipher (French pronunciation: [viʒnɛːʁ]) is a method of encrypting alphabetic text where each letter of the plaintext is encoded with a different Caesar cipher, whose increment is determined by the corresponding letter of another text, the key. The Vigenère cipher is therefore a special case of a polyalphabetic substitution.

Example : If the plaintext is attacking tonight and the key is OCULORHINOLARINGOLOGY, then

- the first letter a of the plaintext is shifted by 14 positions in the alphabet (because the first letter O of the key is the 14th letter of the alphabet, counting from zero), yielding o;
- the second letter t is shifted by 2 (because the second letter C of the key means 2) yielding v;
- the third letter t is shifted by 20 (U) yielding n, with wrap-around;

and so on; yielding the message ovnlqbpvt hznzouz. If the recipient of the message knows the key, they can recover the plaintext by reversing this process.

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 6

AIM: Study and Implement a program for n-gram Hill Cipher

Introduction : Hill cipher is a polygraphic substitution cipher based on linear algebra.Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.
The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

Example : Input : Plaintext: ACT
Key: GYBNQKURP
Output : Ciphertext: POH

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 7

AIM: Study and Use of RSA algorithm (encryption and decryption)

Introduction : RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem, one of the oldest widely used for secure data transmission. The initialism "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977. An equivalent system was developed secretly in 1973 at Government Communications Headquarters (GCHQ), the British signals intelligence agency, by the English mathematician Clifford Cocks. That system was declassified in 1997.
In a public-key cryptosystem, the encryption key is public and distinct from the decryption key, which is kept secret (private). An RSA user creates and publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers are kept secret. Messages can be encrypted by anyone, via the public key, but can only be decrypted by someone who knows the private key.
The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem". Breaking RSA encryption is known as the RSA problem. Whether it is as difficult as the factoring problem is an open question. There are no published methods to defeat the system if a large enough key is used.

Example : First, let's assume you calculated your keys as follows:

1. p=17 and q =7. Notice 17 and 7 are both prime numbers
2. n= 17 x 7 = 119
3. f(n) = (17-1)(7-1)=96
4. e=11, notice that gcd(96,11)=1 and 1<11<96
5. d=35

The keys are:

- private key: {35,119}
- public key: {11,119}

Now, you published somehow your public key and I want to send you a message only you can read. The message I want to send you is M=21. Notice that you can always find a numeric representation for any message. At the end of the day, all data in a computer is represented as numbers.

$$C = M^e \bmod n = 21^{11} \bmod 119 = 98$$

When you receive the encrypted message C=45, you use your private key to decrypt it.

$$M = C^d \bmod n = 98^{35} \bmod 119 = 21$$

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 8

AIM: Study and implement a program of the Digital Signature with RSA algorithm (Reverse RSA)

Introduction : Assume that there is a sender (A) and a receiver (B). A wants to send a message (M) to B along with the digital signature (DS) calculated over the message.
Step-1 : Sender A uses SHA-1 Message Digest Algorithm to calculate the message digest (MD1) over the original message M.
Step-2 : A now encrypts the message digest with its private key. The output of this process is called Digital Signature (DS) of A.
Step-3 : Now sender A sends the digital signature (DS) along with the original message (M) to B.
Step-4 : When B receives the Original Message(M) and the Digital Signature(DS) from A, it first uses the same message-digest algorithm as was used by A and calculates its own Message Digest (MD2) for M.
Step-5 : Now B uses A's public key to decrypt the digital signature because it was encrypted by A's private key. The result of this process is the original Message Digest (MD1) which was calculated by A.
Step-6 : If MD1==MD2, the following facts are established as follows.

- B accepts the original message M as the correct, unaltered message from A.
- It also ensures that the message came from A and not someone posing as A.

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 9

AIM: Study and Use of Diffie-Hellman Key Exchange

Introduction : The Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters.

- For the sake of simplicity and practical implementation of the algorithm, we will consider only 4 variables, one prime P and G (a primitive root of P) and two private values a and b.
- P and G are both publicly available numbers. Users (say Alice and Bob) pick private values a and b and they generate a key and exchange it publicly. The opposite person receives the key and that generates a secret key, after which they have the same secret key to encrypt.

Step-by-Step explanation is as follows:

Alice	**Bob**
Public Keys available = P, G	Public Keys available = P, G
Private Key Selected = a	Private Key Selected = b
Key generated = $x=G^a mod P$	Key generated = $y=G^b mod P$
Exchange of generated keys takes place	
Key received = y	key received = x
Generated Secret Key = $k_a=y^a mod P$	Generated Secret Key = $k_b=x^b mod P$
Algebraically, it can be shown that $k_a=k_b$	
Users now have a symmetric secret key to encrypt	

Example :

Step 1: Alice and Bob get public numbers P = 23, G = 9
Step 2: Alice selected a private key a = 4 and
Bob selected a private key b = 3

Step 3: Alice and Bob compute public values
Alice: x =(9^4 mod 23) = (6561 mod 23) = 6
Bob: y = (9^3 mod 23) = (729 mod 23) = 16
Step 4: Alice and Bob exchange public numbers
Step 5: Alice receives public key y =16 and
Bob receives public key x = 6
Step 6: Alice and Bob compute symmetric keys
Alice: k_a = y^a mod p = 65536 mod 23 = 9
Bob: k_b = x^b mod p = 216 mod 23 = 9
Step 7: 9 is the shared secret.

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 10

A) AIM: Write a program to encrypt the plaintext with the given key.

E.g. plaintext GRONSFELD with the key 1234. Add 1 to G to get H (the letter 1 rank after G is H in the alphabet), then add 2 to C or E (the letter 2 ranks after C is E), and so on. Use smallest letter from plaintext as filler. In the given example last three letters are filler.

Plain letter	G	R	O	N	S	F	E	L	D	D	D	D
Key (repeated)	1	2	3	4	1	2	3	4	1	2	3	4
Cipher Letter	H	T	R	R	T	H	H	P	E	F	G	H

The encrypted message is HTRRTHHPEFGH.

Input : Plaintext (in capital letters only), key (in numbers only)
Output : ciphertext (in capital letters only)

Test Case 1

```
Output
/tmp/8FZ9WXCQfZ.o
Enter the text to encrypt: GRONSFELD
min letter=D
Enter the key: 1234
Revised Plaintext after padding = GRONSFELDDDD
ciphertext =HTRRTHHPEFGH
```

Test Case 2

```
Output
/tmp/8FZ9WXCQfZ.o
Enter the text to encrypt: SAMPLETEXT
min letter=A
Enter the key: 2145
Revised Plaintext after padding = SAMPLETEXTAA
ciphertext =UBQUNFXJZUEF
```

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

B) AIM: Encrypt the input words PLAINTEXT= RAG BABY to obtain CIPHERTEXT = SCJ DDFD

Plain letter	R	A	G		B	A	B	Y
word i	1	1	1		2	2	2	2
letter j	0	1	2		0	1	2	3
i+j	1	2	3		2	3	4	5
Cipher letter	S	C	J		D	D	F	D

Input : Plaintext (in capital letters only, upto nine words only)
Output : ciphertext (in capital letters only)
Note : Students can use string.h if required.

Test case 1	Test Case 2
Output /tmp/8FZ9WXCQfZ.o Enter the plaintext: RAG BABY Plaintext: RAG BABY Key: 123 2345 Ciphertext: SCJ DDFD	Output /tmp/8FZ9WXCQfZ.o Enter the plaintext: THIS IS SAMPLE TEXT Plaintext: THIS IS SAMPLE TEXT Key: 1234 23 345678 4567 Ciphertext: UJLW KV VERVSM XJDA

Output : ciphertext (in capital letters only)

Security Analysis: The student must perform a security analysis of the algorithm mentioned above. Analysis includes a) Cryptanalysis, b)Design analysis, c)Performance analysis

Sign of Faculty **Date of performance**

Experiment 11

A) AIM: Mini Project To understand the need for Post-Quantum Cryptography.

To understand the need for post-quantum cryptography, analyse how quantum computing threatens traditional cryptographic schemes.

- Explain the threat posed by quantum computers to RSA, ECC, and other traditional cryptosystems.
- Understand the basics of post-quantum cryptographic schemes.
- Evaluate security properties and draw comparisons with classical cryptography.

Quantum Threats:

- **Shor's Algorithm:** Breaks RSA, ECC efficiently.
- **Grover's Algorithm:** Weakens symmetric key strength (AES-128 → AES-64-bit equivalent).

NIST PQC Standardization Effort:

- Leading candidates: CRYSTALS-Kyber (encryption), CRYSTALS-Dilithium (signature), Falcon, etc.

Sign of Faculty **Date of performance**