JINDAL VIDYA MANDIR, VIDYANAGAR

2024-25

Computer Science Project :

Introduction:

" Cipher-Based Encryption and Decryption System for Secure Communication "

Submitted By: P.Nitin.

Class/Sec: XI - B

Index:

Contents:

	Pg.No
Certificate	
Acknowledgement	
Introduction	
Project Report	
Working Description	
Source Code	
Output	
Conclusion & Bibliography	

CENTRAL BOARD OF SECONDARY EDUCATION

JINDAL VIDYA MANDIR, VIDYANAGAR

This is to certify that the project titled "**Cipher based Encryption and Decryption System for Secure Communication**" was successfully carried out by "**Nitin.P"** of **Class XI- B** during the Academic year **2024-2025**

Mrs.Ishwari Sharma Principal	External Examiner	**Mrs.Soubhagya** Dept. Of Computer Science

Acknowledgement

I would like to express my sincere gratitude to my Computer Science teacher Soubhagya Mam, for their invaluable guidance, encouragement, and support throughout the development of this project, "**Cipher-Based Encryption and Decryption System for Secure Communication".**

I am also thankful to my school, **Jindal Vidya Mandir, Vidyanagar** for providing the resources and platform to explore and enhance my programming skills. Lastly, I extend my heartfelt thanks to my family and friends for their encouragement and feedback, which motivated me to complete this project successfully.

Introduction to the Project:

In the digital age, **secure communication** is essential to **protect sensitive information from unauthorized access.** Messages shared through online platforms are often at risk of being intercepted, leading to potential privacy breaches.

This project, **Cipher-Based Encryption and Decryption System for Secure Communication,** provides a reliable solution by **encrypting and decrypting messages using a unique passkey**. A **separate key generator is used to create a randomized passkey**, ensuring that each communication session uses a distinct and secure key. The **sender and receiver must share this passkey to encode and decode the messages,** making the system both effective and private.

Developed in Python, this project introduces **key-based encryption using a substitution cipher.** It highlights the importance of secure data exchange and gives us hands-on experience with **fundamental cryptographic concepts. By incorporating a dedicated key generator,** the system ensures uniqueness and reduces the chances of predictability, making it **a practical tool for private messaging and secure communication.**

This project not only serves as an educational demonstration of encryption but also lays the groundwork for more **advanced secure communication systems** in real-world applications.

Project Report:

The objective of this project is to create a Python-based system that enables secure communication through message encryption and decryption. The system utilizes a separate key generator to produce unique randomized passkeys, ensuring privacy and security for the sender and receiver.

1. **Separate Key Generator:**

   Generates a unique, randomized passkey for each session.

   Ensures security by providing an unpredictable substitution cipher.

2. **Encryption**:

   Converts a plaintext message into an encrypted format using the passkey.

   Prevents unauthorized access to the message.

3. Decryption:

Reverts the encrypted message back into plaintext using the same passkey.

Ensures that only authorized parties can decode the message.

4.**Interactive Options:**

Users can choose to encrypt or decrypt messages.

Flexibility to switch or continue actions within the program.

4. Technology Used

Programming Language: Python

5. Working

Step 1: Key Generation

A separate program generates a unique randomized key, which is a shuffled version of all valid characters (letters, digits, punctuation, and space). This key is shared between the sender and receiver.

Step 2: Encryption

The user selects the encryption option.

The plaintext message is converted into ciphertext by mapping each character to its corresponding character in the randomized key.

Step 3: Decryption

The user selects the decryption option.

The ciphertext is converted back into plaintext using the same passkey.

Code Implementation

The project comprises two programs:

1. Key Generator: To generate a unique randomized passkey.

```
import random
import string
chars = " " + string.punctuation + string.digits +
string.ascii_letters
chars = list(chars)

key = chars.copy()
random.shuffle(key)

print("Generated Key: ", key)
```