Table of Contents

Chapter 1
Introduction

1.1 Background

In the rapidly evolving digital age, the security of sensitive information during transmission has become crucial to protecting individual and organizational data integrity. One well-established encryption technique, the Triple Data Encryption Standard (3DES), serves as a robust solution to enhance data security. Originally developed as an extension of the Data Encryption Standard (DES), 3DES strengthens protection by applying the DES algorithm three times, significantly increasing resistance to attacks compared to single DES. In this process, the encryption key is expanded, and each round of DES encryption-decryption-encryption uses distinct keys, making it exponentially harder for unauthorized parties to decipher the message without all three keys.

However, while 3DES is notably secure, modern advancements in cryptographic attacks have exposed certain vulnerabilities in its structure. For instance, brute-force attacks and meet-in-the-middle attacks pose potential threats, especially as computing power continues to grow. To address this challenge, integrating Triple DES with a One-Time Pad (OTP) further enhances security. The OTP, based on Shannon's theorem, is theoretically unbreakable when used correctly, as it introduces a unique, truly random key that is used only once per encryption session and then discarded. By combining 3DES with an OTP, this project introduces a hybrid model designed to protect sensitive communication with layers of encryption and randomized uniqueness, adding complexity to unauthorized decryption attempts.

1.2 Objectives

The primary objectives of this project are to:

1. Implement Triple DES encryption integrated with OTP for enhanced security:
Develop a custom encryption model using the Triple DES algorithm with an added One-Time Pad to improve the protection of transmitted data.

2. Securely transmit encrypted messages and OTPs via WhatsApp:
Facilitate the secure transmission of encrypted messages and corresponding OTPs through WhatsApp, leveraging it as an accessible and popular communication medium.

3. Analyse the effectiveness of combining 3DES with OTP in improving data confidentiality and security: Evaluate the performance of the hybrid model by comparing it with traditional encryption methods to understand how the OTP integration strengthens confidentiality.

This project aims to provide an insightful exploration of using 3DES with OTP in practical applications, demonstrating how layered encryption strategies can enhance the security of sensitive communication in today’s digital era.

1.3 Scope

Scope of the Project

This project’s scope encompasses the design, development, and testing of a hybrid encryption model combining Triple DES (3DES) with a One-Time Pad (OTP) to achieve enhanced data security. The following are the key aspects included in the scope:

1. **Encryption Algorithm Design and Implementation**:
   - Develop a secure hybrid encryption model that combines 3DES with OTP.
2. **Message Transmission via WhatsApp**:
   - Explore and implement methods for securely transmitting encrypted messages and OTPs through WhatsApp.

Chapter 2
Literature Review

2.1 Overview

Triple DES (3DES) Encryption

Triple DES (3DES) is a symmetric encryption algorithm that enhances the security of the original Data Encryption Standard (DES) by applying it three times to each data block. DES, initially developed by IBM in the 1970s and adopted as a federal standard in 1977, uses a 56-bit key, which makes it vulnerable to brute-force attacks in today's era of advanced computing. To address these vulnerabilities, Triple DES was introduced as a strengthened alternative. By applying DES encryption, decryption, and then encryption (ED-E) with three distinct 56-bit keys, 3DES effectively increases the key size to 168 bits, enhancing resistance against attacks and providing better data security.

While 3DES remains popular for various secure communication applications due to its robustness, it is notably slower compared to modern algorithms like Advanced Encryption Standard (AES). AES, which is based on the Rijndael algorithm, uses block sizes and key lengths (128, 192, or 256 bits) that enable faster processing and stronger security. Because of its performance benefits and greater efficiency in handling large amounts of data, AES is gradually replacing 3DES in many sectors. Nonetheless, 3DES continues to be widely used, especially in legacy systems, financial applications, and secure communications where trusted algorithms are required.

One-Time Pad (OTP) Encryption

The One-Time Pad (OTP) is a cryptographic technique that dates back to the early 20th century. It is renowned for being theoretically unbreakable when implemented correctly, as long as specific criteria are met. OTP relies on a unique, random key (or "pad") that is as long as the message itself. This key is combined with the plaintext message using a bitwise XOR operation to generate the ciphertext. To decrypt the message, the same unique key is applied again. Due to its randomness and single-use nature, OTP provides perfect secrecy and ensures that every message is unique and resistant to cryptanalysis.

For OTP to remain unbreakable, the key must be:

1. Truly random and equal in length to the message.
2. Used only once (hence "One-Time").
3. Kept absolutely secret between the sender and recipient.

OTP is, therefore, particularly secure but presents practical challenges in key generation, management, and distribution. Despite these complexities, OTP has been historically used in high-stakes contexts such as confidential government communications, military operations, and intelligence sharing, where data confidentiality is paramount.

Combining 3DES with OTP: Enhanced Security Model

In this project, combining 3DES with OTP aims to leverage the strengths of both encryption techniques to build a hybrid model that enhances data security. The combination introduces OTP as an additional layer of security that wraps around the 3DES-encrypted message, ensuring that even if an attacker were to obtain the 3DES key, they would still need the unique OTP to decrypt the data. This hybrid model benefits from 3DES's robustness in securing data through triple encryption and OTP's theoretical unbreakability, which offers an extra line of defense.

Adding OTP to 3DES also minimizes risks associated with known vulnerabilities in 3DES, such as potential brute-force and meet-in-the-middle attacks. In essence, if the 3DES key is compromised, the OTP layer preserves confidentiality, given that the OTP is a one-time, unique, and random key.

2.2 Real Time Applications

Both 3DES and OTP have found applications in secure communication across various industries:

1. Secure Messaging: 3DES, with OTP, can be employed in messaging systems where confidentiality is critical. This combination ensures that each message transmitted is encrypted uniquely, making it significantly more secure.
2. Banking Transactions: The financial sector has long relied on 3DES to protect sensitive banking information during transactions. Combining 3DES with OTP provides an added layer of security, valuable for high-value or highly sensitive data transfers.
3. Confidential Government Communications: Governments have historically utilized OTP for military and diplomatic messaging due to its theoretical unbreakability. Introducing 3DES alongside OTP can further protect critical information, even if the OTP component is partially compromised. By exploring and implementing a 3DES + OTP model, this project investigates a potential pathway for strengthening encryption mechanisms, particularly for applications that require high levels of security in data transmission. The fusion of these techniques offers a balance of practicality and confidentiality, reinforcing data protection and aligning with modern requirements for secure digital communication.

Chapter 3
Methodology

3.1 Approach

This project employs a structured, multi-stage approach to ensure the secure implementation and evaluation of the hybrid encryption model combining Triple DES (3DES) with a One-Time Pad (OTP). The methodology is designed to address both the encryption and secure transmission of messages, focusing on system architecture, encryption logic, and message delivery. The approach is divided into three primary phases:

1. **Key and OTP Generation System**:
    - **Encryption Key Generation**: The initial phase involves generating keys for the 3DES encryption. Given the encryption requirements, three 56-bit keys are generated and combined to form the 168-bit key required for 3DES. This key generation process must follow high security standards to ensure key randomness and reduce vulnerability to brute-force attacks.
    - **One-Time Pad (OTP) Generation**: Simultaneously, a unique OTP is generated for each message. The OTP, which is a randomly generated key, is equal in length to the message and is discarded after single use. This ensures the OTP maintains its theoretical unbreakability, as each key is unique to its corresponding message and not reused.
2. **Encryption and Decryption Functions**:
    - **Encryption Process**: The hybrid encryption process starts by encrypting the plaintext message using 3DES with the previously generated key. Once the data is encrypted with 3DES, the OTP is applied to the 3DES ciphertext using a bitwise XOR operation. This step introduces an extra security layer by ensuring that the ciphertext is randomized uniquely for each transmission.
    - **Decryption Process**: On the receiving end, decryption is performed by first applying the OTP to the incoming encrypted data using XOR to retrieve the 3DES-encrypted message. The recipient then uses the shared 3DES key to decrypt the message. This two-step decryption sequence mirrors the encryption process, ensuring that only users with both the 3DES key and OTP can decipher the message.
    - **Data Encoding**: The final ciphertext is encoded in base64 to ensure it is safe for transmission over messaging platforms. Encoding also simplifies the decryption process, as it standardizes the data format received.

3. **Sending OTP via WhatsApp**: o To facilitate secure OTP transmission, the project utilizes **Pywhatkit**, a Python library that automates WhatsApp messaging. This step allows the OTP to be sent to the intended recipient's WhatsApp account, providing a secure, familiar communication medium. o **Message Automation**: With Pywhatkit, a WhatsApp message containing the OTP is automatically generated and sent to the recipient's phone number. Pywhatkit initiates the WhatsApp web interface on the computer, inputs the OTP message, and schedules it for transmission, minimizing user intervention.
    - **Time-Synchronization**: Using the datetime library, the script schedules the OTP message within a designated timeframe to ensure the user receives the OTP promptly. This timing coordination helps maintain the OTP's one-time use nature by aligning message delivery with encryption timing. **Tools and Technologies**

3.2 Tools and Technologies

The project leverages several **Python libraries** and tools essential to implementing and automating the encryption, decryption, and OTP messaging processes:

1. **Python Libraries**:
    - **PyCryptodome**: This library provides the cryptographic functionalities required to implement Triple DES (3DES) encryption and decryption. PyCryptodome's DES3 module enables the use of 3DES, supporting key generation, encryption, and decryption while ensuring compatibility with Python-based cryptographic operations.
    - **Pywhatkit**: Used to automate the WhatsApp messaging process, Pywhatkit allows for programmatically sending the OTP to the recipient's WhatsApp account. This automation reduces the risk of manual errors, improves message delivery speed, and leverages WhatsApp's existing security protocols for message transmission.
    - **datetime and base64**: The datetime library is used for scheduling OTP transmissions, ensuring OTPs are sent at the correct time relative to encryption. The base64 library is used to encode and decode data, making it safe for transmission over WhatsApp by converting binary data into an ASCII string format.
2. **Hardware**:
    - The project is developed on a **standard computer setup**, equipped with necessary resources for running Python scripts, testing encryption, and managing WhatsApp automation. Testing and debugging are conducted locally to verify encryption functionality, the automation process, and the end-to-end data flow from encryption to secure transmission.

This methodology provides a comprehensive approach to developing a secure communication model by integrating 3DES with OTP. It combines encryption and OTP messaging through an automated workflow, enhancing both security and usability. Through this structured approach and toolset, the project aims to deliver an encryption solution with practical applications in secure data transmission and messaging.

Chapter 4
Implementation

4.1 Detailed Description

The implementation of this project consists of two main sections: encrypting messages with a combination of Triple DES (3DES) and One-Time Pad (OTP) for secure communication, and securely transmitting the OTP to the recipient via WhatsApp. Additionally, the project includes a decryption process for the recipient to retrieve the original message. This section details the structure of the code and functions used for OTP transmission and secure message handling.

OTP Transmission via WhatsApp

The **send_whatsapp_message** function is responsible for securely delivering the OTP to the recipient using WhatsApp. This function utilizes the **Pywhatkit** library, which automates WhatsApp Web to schedule and send messages. By programmatically delivering the OTP, the function enhances the security of the process and minimizes the risk of manual errors or delays.

Key Aspects of the send_whatsapp_message Function

1. **Function Parameters**:
   - **Phone Number**: The recipient's WhatsApp contact number. This is formatted with the international dialing code (e.g., +1 for the US, +44 for the UK) to ensure successful delivery.
   - **Message**: The OTP to be sent, formatted as a short, secure string that the recipient will use to decrypt the message.

Scheduled Time: The time when the OTP message should be sent. To coordinate message delivery and message decryption, the scheduled time is set 2 minutes from the current execution time to allow the system sufficient processing time for encryption and transmission.

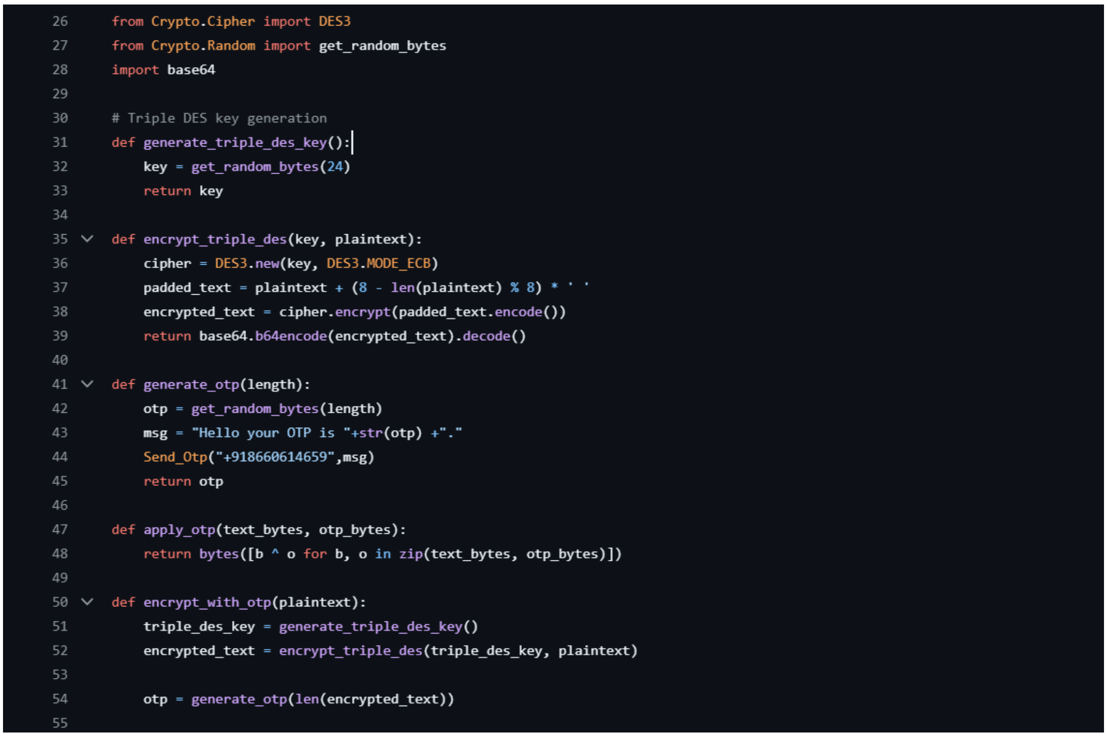
2. **Setting the Scheduled Time**: o The function retrieves the **current time** using the datetime.datetime.now() method. It then adds 2 minutes to the current time using timedelta(minutes=2), ensuring that the OTP is sent shortly after encryption but still allowing enough time for processing.
    - This synchronization minimizes the risk of the recipient receiving an outdated OTP and ensures they have immediate access to the decryption key needed.
3. **Using Pywhatkit to Send the OTP**:
    - Pywhatkit's **sendwhatmsg** function enables automated WhatsApp messaging. This function opens WhatsApp Web, inputs the provided contact number, message, and scheduled time, and sends the message without requiring manual intervention.

4.2 Error Handling

- The function includes error handling mechanisms to ensure the reliability of OTP transmission. Pywhatkit may occasionally encounter issues due to network connectivity, browser response delays, or WhatsApp Web access issues.
- Using a **try-except block**, the function attempts to send the message and captures any errors that may occur. If an error is raised, a descriptive message (such as "Message sending failed") is displayed, and alternative actions can be taken as needed.
- This error handling mechanism enhances the robustness of the messaging function, allowing for real-time troubleshooting and reducing the likelihood of failed OTP transmissions.

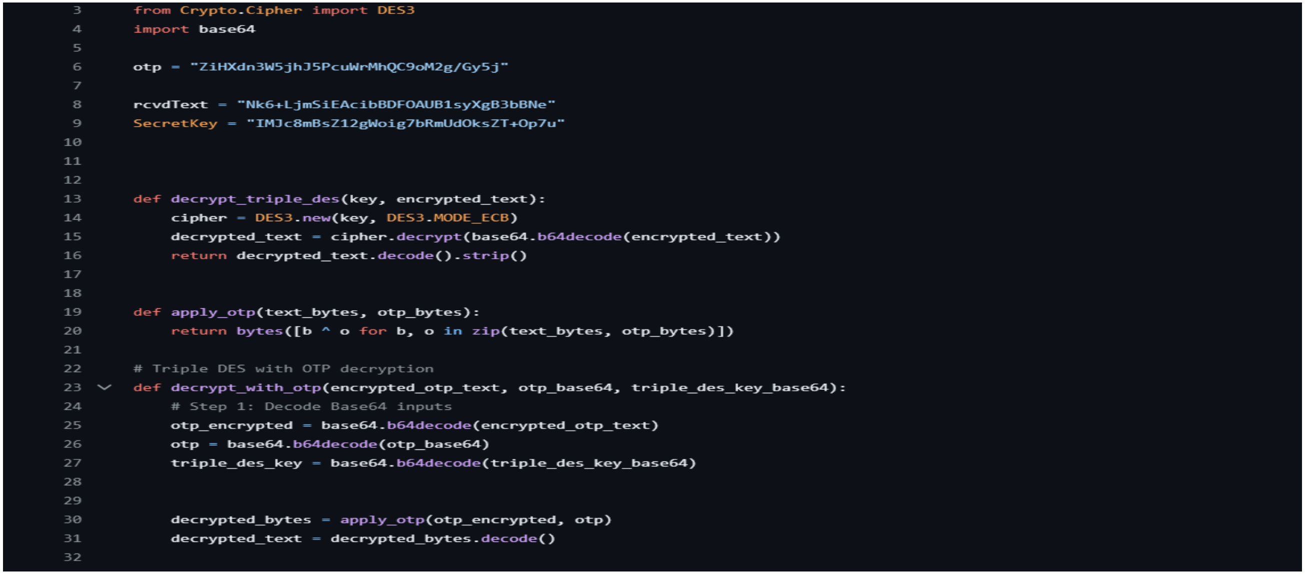
4.3 Code Screenshots

This is a snippet for PC 1



```python
from Crypto.Cipher import DES3
from Crypto.Random import get_random_bytes
import base64

# Triple DES key generation
def generate_triple_des_key():
    key = get_random_bytes(24)
    return key

def encrypt_triple_des(key, plaintext):
    cipher = DES3.new(key, DES3.MODE_ECB)
    padded_text = plaintext + (8 - len(plaintext) % 8) * ' '
    encrypted_text = cipher.encrypt(padded_text.encode())
    return base64.b64encode(encrypted_text).decode()

def generate_otp(length):
    otp = get_random_bytes(length)
    msg = "Hello your OTP is "+str(otp) +"."
    Send_Otp("+918660614659",msg)
    return otp

def apply_otp(text_bytes, otp_bytes):
    return bytes([b ^ o for b, o in zip(text_bytes, otp_bytes)])

def encrypt_with_otp(plaintext):
    triple_des_key = generate_triple_des_key()
    encrypted_text = encrypt_triple_des(triple_des_key, plaintext)

    otp = generate_otp(len(encrypted_text))

```

This is snippet for PC 2



```python
from Crypto.Cipher import DES3
import base64

otp = "ZiHXdn3W5jhJ5PcuWrMhQC9oM2g/Gy5j"

rcvdText = "Nk6+LjmSiEAcibBDFOAUB1syXgB3bBNe"
SecretKey = "IMJc8mBsZ12gWoig7bRmUdOksZT+Op7u"



def decrypt_triple_des(key, encrypted_text):
    cipher = DES3.new(key, DES3.MODE_ECB)
    decrypted_text = cipher.decrypt(base64.b64decode(encrypted_text))
    return decrypted_text.decode().strip()


def apply_otp(text_bytes, otp_bytes):
    return bytes([b ^ o for b, o in zip(text_bytes, otp_bytes)])

# Triple DES with OTP decryption
def decrypt_with_otp(encrypted_otp_text, otp_base64, triple_des_key_base64):
    # Step 1: Decode Base64 inputs
    otp_encrypted = base64.b64decode(encrypted_otp_text)
    otp = base64.b64decode(otp_base64)
    triple_des_key = base64.b64decode(triple_des_key_base64)


    decrypted_bytes = apply_otp(otp_encrypted, otp)
    decrypted_text = decrypted_bytes.decode()

```

Chapter 5
Results and Discussion

The project achieved its primary goal of enhancing message security by successfully implementing a hybrid encryption model using **Triple DES (3DES) with One-Time Pad (OTP)**. This section provides an overview

of the outcomes, evaluates the effectiveness of combining 3DES with OTP, and discusses limitations encountered during the implementation.

5.1 Results

1. **Successful Encryption and Decryption**:
   - Messages were effectively encrypted using the 3DES algorithm, with OTP applied as an additional layer of security. The encryption process created ciphertext that was significantly resistant to brute-force and cryptographic attacks, as the OTP layer introduced randomness that varied for each message.
   - The decryption process on the recipient's end was accurate, confirming that the 3DES key and OTP were transmitted and utilized correctly. The message could only be decrypted by users with access to both the 3DES key and the OTP, ensuring a high level of confidentiality.
2. **OTP Transmission via WhatsApp**:
   - Using the Pywhatkit library, OTPs were transmitted successfully to the recipient's WhatsApp account. This automated process allowed for prompt and efficient OTP delivery, aligning well with the encryption and decryption workflow. Pywhatkit enabled WhatsApp Web to initiate the message, reducing manual intervention and errors.
   - The OTP, being single-use and randomly generated, effectively functioned as a unique key for each message. This single-use nature preserved the OTP's theoretical unbreakability, ensuring that the ciphertext could not be reused or deciphered without the OTP.

5.2 Discussion

The project demonstrated the feasibility of combining 3DES and OTP to enhance message security. However, some limitations and areas for improvement were identified:

1. **Timing Constraints with WhatsApp Automation**:

   While Pywhatkit automated OTP delivery via WhatsApp effectively, timing constraints emerged. Pywhatkit schedules messages at a specified time, introducing a brief waiting

- period that could slightly delay OTP transmission. In situations requiring immediate OTP delivery, this delay may pose a challenge, as WhatsApp Web requires a short setup time, and Pywhatkit may experience network-related latency.

2. **Secure Key Distribution**:
    - While the OTP was transmitted securely over WhatsApp, distributing the 3DES key to recipients posed a challenge. Secure distribution of encryption keys is essential to prevent unauthorized access, and since the 3DES key must remain confidential, a secure channel or pre-arranged method is needed to share it with the intended recipients.
    - Without a secure key management protocol, there is a risk of interception or unauthorized access. Future implementations may consider using Public Key Infrastructure (PKI) for key exchange, allowing the 3DES key to be shared securely via encrypted channels.
3. **Scalability and Real-World Application**:
    - While the project met its objectives in a controlled setting, scalability remains a consideration. Automating WhatsApp messaging using Pywhatkit is feasible for single-use cases but may become inefficient for high-volume applications. For large-scale systems, implementing a dedicated messaging API (such as Twilio for SMS or WhatsApp Business API) may offer greater control and flexibility.
    - For applications where message security is critical, real-time data encryption and decryption with reduced dependency on external messaging platforms could further improve the model’s practicality.

Chapter 6
Conclusion

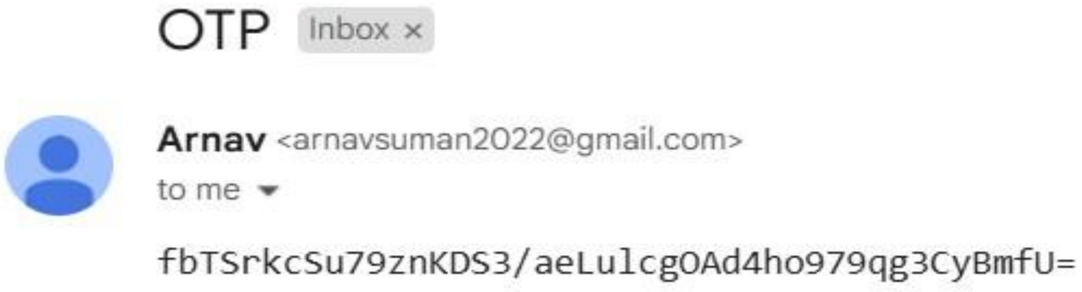
This project successfully demonstrates the effectiveness of combining **Triple DES (3DES)** encryption with a **One-Time Pad (OTP)** for enhanced security in message encryption and secure transmission. By layering 3DES with OTP, the project adds a significant additional security measure that addresses some of the vulnerabilities associated with 3DES alone, resulting in a more resilient and robust encryption model. This approach not only strengthens data confidentiality but also provides a practical framework for secure message delivery in real-world applications.

6.1 Key Findings

1. **Enhanced Security Through Layered Encryption**: o While 3DES is inherently secure, its reliance on a single key leaves it potentially vulnerable to specific attacks, such as brute force and meet-in-the-middle attacks. The addition of OTP introduces a unique, per-message key that further strengthens security. By applying OTP to the 3DES-encrypted message, the encryption model gains resilience against interception and unauthorized access. Even if an attacker were to obtain the 3DES
2. key, the unique OTP applied to each message makes the data challenging to decipher without access to both the 3DES key and the OTP.
3. **Protection Against Attack Vectors**:
    - The layered encryption scheme provides improved protection against common attack vectors. Traditional 3DES, while robust, can be weakened if an attacker gains access to the encryption key. The OTP layer mitigates this by adding an unpredictable, randomized element unique to each message, thus reducing susceptibility to key-based attacks. This characteristic makes the hybrid model suitable for environments where message confidentiality is critical, such as financial communications, secure messaging, and sensitive government transactions.
4. **Practical Implementation and Secure OTP Transmission**:
    - By leveraging **Pywhatkit** for automated OTP transmission via WhatsApp, the project establishes a feasible method for secure key exchange. Although WhatsApp is widely used and secured with end-to-end encryption, the implementation shows that automated tools like Pywhatkit can deliver OTPs efficiently in controlled environments. This process demonstrated that integrating messaging automation with encryption could enable streamlined, secure communication in scenarios requiring quick, reliable OTP delivery.

5. **Foundational Framework for Future Applications**:
    - The successful integration of 3DES and OTP highlights the potential for multi-layered encryption as a foundation for future security solutions. Layered encryption models, such as this one, can form the basis of a security framework adaptable to various messaging and communication platforms, addressing the growing need for data confidentiality in a digital world.

6.2 Snapshots

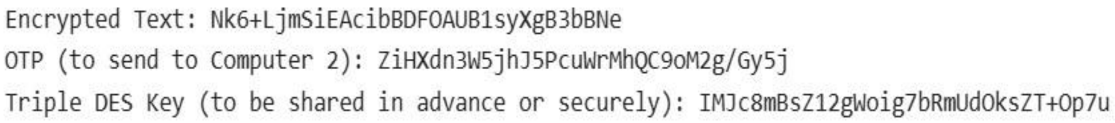


OTP

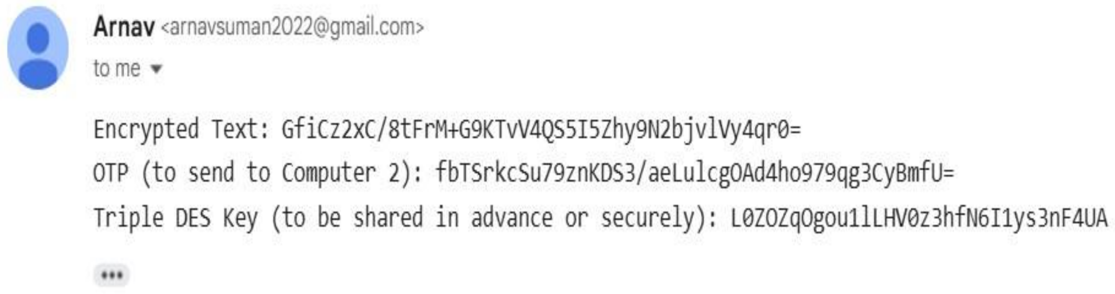
Arnav <arnavsuman2022@gmail.com>

to me

fbTSrkcSu79znKDS3/aeLulcgOAd4ho979qg3CyBmfU=



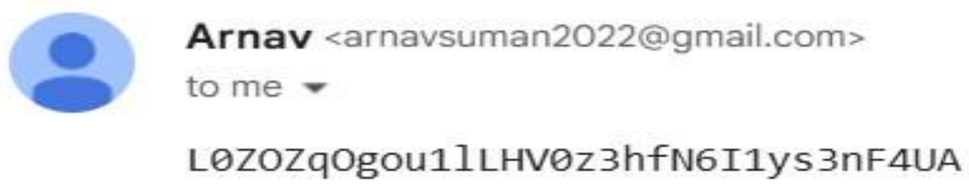
```
Encrypted Text: Nk6+LjmSiEAcibBDFOAUB1syXgB3bBNe
OTP (to send to Computer 2): ZiHXdn3W5jhJ5PcuWrMhQC9oM2g/Gy5j
Triple DES Key (to be shared in advance or securely): IMJc8mBsZ12gWoig7bRmUdOksZT+Op7u
```



Arnav <arnavsuman2022@gmail.com>

to me

```
Encrypted Text: GfiCz2xC/8tFrM+G9KTvV4QS5I5Zhy9N2bjvlVy4qr0=
OTP (to send to Computer 2): fbTSrkcSu79znKDS3/aeLulcgOAd4ho979qg3CyBmfU=
Triple DES Key (to be shared in advance or securely): L0ZOZqOgou1lLHV0z3hfN6I1ys3nF4UA
```



Arnav <arnavsuman2022@gmail.com>

to me

```
L0ZOZqOgou1lLHV0z3hfN6I1ys3nF4UA
```

Few Snapshots/Outputs from Mail