An

Industry Oriented Mini Project Report

on

Hybrid Encryption Technique To Fortify Caesar Cipher

A report submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

By

Poddutoori Neetha Reddy

(20EG105435)

Madasu Surya Teja

(20EG105423)

Under the guidance of

Mr. B. Ravinder Reddy

Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENKATAPUR– 500088
GHATKESAR
TELANGANA
2023-24

CERTIFICATE

This is to certify that the Report/dissertation entitled **"HYBRID ENCRYPTION TECHNIQUE TO FORTIFY CAESAR CIPHER"** that is being submitted by **Ms. Poddutoori Neetha Reddy** bearing the hall ticket number **20EG105435**, **Madasu Surya Teja** bearing the hall ticket number **20EG105423,** in partial fulfillment for the award of B.Tech in Computer Science And Engineering to the Anurag University is a record of bonafide work carried out by him / her under our guidance and supervision.

The results embodied in this Report have not been submitted to any other university or Institute for the award of any degree or diploma.

Signature of The Supervisor
B. Ravinder Reddy
(Assistant Professor)

Dean, Department of CSE

External Examiner 1

External Examiner 2

DECLARATION

I Here declare that the Report entitled "**HYBRID ENCRYPTION TECHNIQUE TO FORTIFY CAESAR CIPHER**" submitted for the award of Bachelor of technology Degree is our own work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

Poddutoori Neetha Reddy
(20EG105435)

Madasu Surya Teja
(20EG105423)

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **B. Ravinder Reddy, Assistant Professor, Department of CSE** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Department of CSE, Anurag University. We also express my deep sense of gratitude to **Dr. V V S S S Balaram**, Academic Coordinator. **Dr .Pallam Ravi**, Project Coordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage of our project work.

We would like to express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for his encouragement and timely support in our B.Tech program.

Poddutoori Neetha Reddy

(20eg105435)

Madasu Surya Teja

(20eg105423)

ABSTRACT

In today's digital world, data security is of supreme importance. Encryption plays a vital role in protecting confidential information from unauthorized access.This paper introduces a novel hybrid encryption algorithm that combines the strengths of RSA (Rivest-Shamir-Adleman) and the Caesar cipher to enhance data security and avoid intrusions. Although the Caesar cipher is well known for its simplicity and efficiency it is vulnerable to frequency analysis attack, dictionary attack and brute force attack. To prevail over these vulnerabilities, we are proposing a hybrid encryption algorithm in which the data is initially encrypted using the Caesar cipher. Following, the Caesar cipher's encrypted plaintext is encrypted using RSA, applying its asymmetric-key encryption capabilities. This two-step encryption or double encryption process creates a balanced approach that is both efficient and secure. We evaluate the effectiveness of our proposed algorithm and demonstrate its resistance to the attacks. The results demonstrate that the hybrid algorithm enhances the security of the encryption process and prevents the known attacks. This paper describes the use of hybrid encryption techniques to provide security for confidential data.

Keywords:- Encryption, RSA, Caesar Cipher, Frequency analysis attack, Data security

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

In an era defined by the ubiquitous presence of digital information, the security and privacy of data have become paramount concerns. The field of cryptography, an ancient art turned modern science, plays a central role in securing sensitive information and enabling secure communication over digital networks. By leveraging mathematical and computational techniques, cryptography seeks to protect data from unauthorized access, ensuring confidentiality, integrity, and authenticity.

At its core, cryptography involves the transformation of plaintext into ciphertext, making it unreadable to anyone without the appropriate decryption key. This process provides a shield against eavesdropping, data tampering, and unauthorized access. Cryptographic algorithms utilize complex mathematical operations and logical structures, making it extremely challenging for unauthorized entities to decipher encrypted data without the corresponding decryption key.

As the digital landscape continues to expand and evolve, the continued development of robust cryptographic techniques, along with the adoption of best practices in key management, becomes imperative. By embracing innovative cryptographic solutions and fostering a culture of security awareness, society can build a resilient digital infrastructure, ensuring the confidentiality, integrity, and authenticity of information in the digital age.

Cryptography algorithms can be broadly categorized into two main types: symmetric key algorithms and asymmetric key algorithms. Each type serves different purposes and has unique characteristics that make it suitable for various cryptographic applications. A symmetric algorithm, also known as a symmetric key algorithm or secret key algorithm, is a type of encryption algorithm where the same key is used for both encryption and decryption of the data. This key is kept secret and must be known only to the parties that need to communicate securely. These algorithms encrypt fixed-size blocks of data and are commonly used for data encryption and decryption. Examples include the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES).

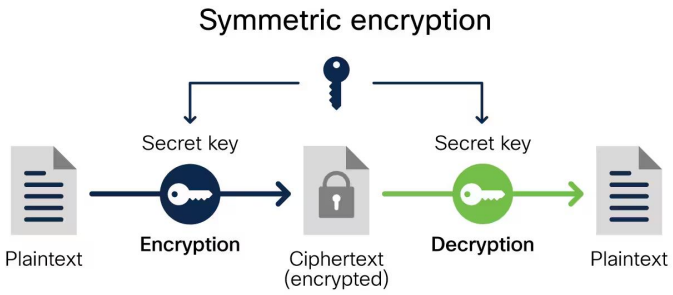

Figure 1.1 Symmetric Encryption

Asymmetric algorithms, also known as public-key cryptography, are a type of cryptographic algorithm that uses two different but mathematically related keys - a public key and a private key. These keys are used for encryption and decryption, digital signatures, and key exchange. Asymmetric algorithms offer a secure method for communication and data protection, particularly in scenarios where key exchange and secure communication are required. Examples include Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA), RSA (Rivest-Shamir-Adleman).

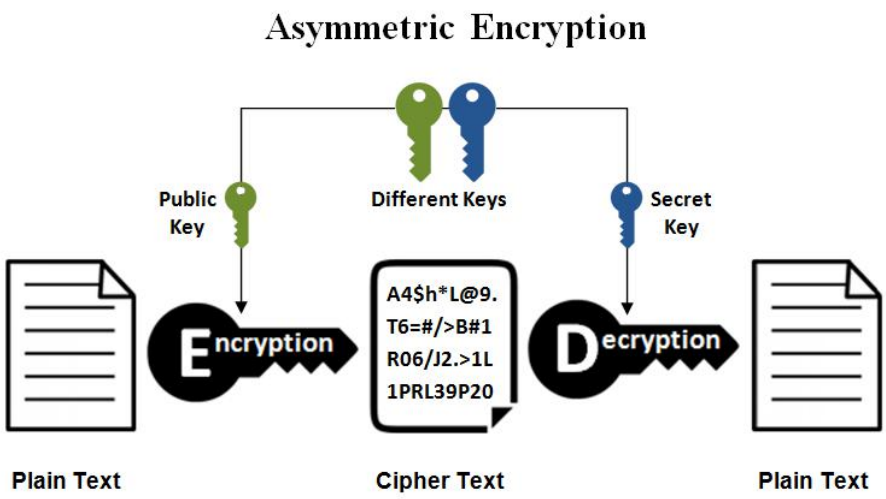

Figure 1.2. Asymmetric Encryption

A substitution cipher is a method of encryption where each letter in the plaintext is replaced by a different letter or symbol based on a predetermined key. It is one of the simplest forms of encryption and was used extensively in ancient times. Among the

earliest and simplest encryption techniques is the Caesar cipher. The Caesar cipher is a substitution cipher that played a crucial role in the protection of sensitive information during ancient times. Although the Caesar cipher is notably simple, it served as an effective means of concealing information in its time.

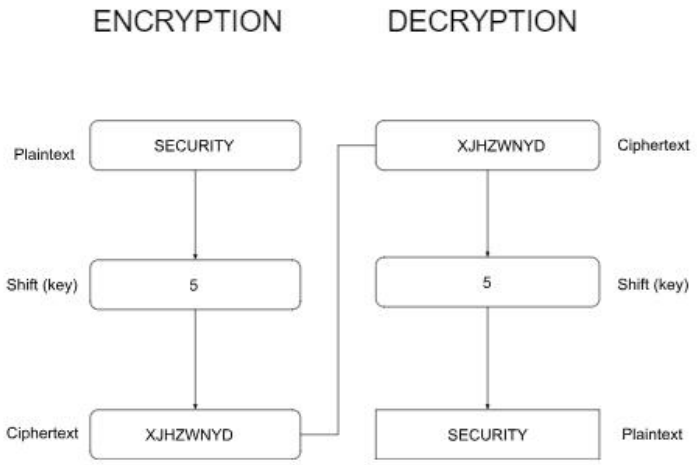

Figure 1.3. Existing Method

However, its security is quite limited by modern standards and its susceptibility to brute-force attacks and frequency analysis, makes it unsuitable for securing sensitive data in today's cyber-threat landscape.

In this work, we proposed a hybrid model combining caesar cipher and rsa. This hybrid approach strikes a balance between efficiency and robustness, making it suitable for diverse applications, from secure communication to data storage and transmission. This method of hybrid encryption provides confidentiality and security by preventing all the vulnerabilities of the caesar cipher.

1.1 CAESAR CIPHER

The Caesar cipher, named after Julius Caesar, is one of the oldest and simplest encryption methods in the history of cryptography. It operates by shifting each letter in the plaintext by a fixed number of positions down or up the alphabet.While it is remarkably easy to understand and implement, the Caesar cipher offers limited security, primarily due to its vulnerability to brute-force attacks, frequency analysis and dictionary attack. Encryption in the Caesar cipher involves using a shift or key.

Let's consider an example to understand the process. We have shift as 5 and we want to encrypt the plaintext " SECURE".

Plaintext: S E C U R E
Shift: 5

We would shift each letter in the plaintext 5 positions down the alphabetic order.

S -> X
E -> J
C -> H
U -> Z
R -> W
E -> J

So, the encrypted ciphertext for "SECURE" with a Caesar cipher shift of 5 is "XJHZWJ."

Decryption of a Caesar cipher is the process of reversing the encryption, reverse the shift by moving each letter in the ciphertext backward in the alphabetic order by the same number of positions as the shift value to obtain the original plaintext.

1.2 RIVEST-SHAMIR-ADLEMAN (RSA)

It is a widely used asymmetric encryption algorithm.It was developed by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977 and remains one of the most robust and secure encryption methods available. RSA relies on the mathematical properties of large prime numbers to achieve its security.

RSA algorithm:-

1. Generate two large prime numbers, typically denoted as 'p' and 'q.' These prime numbers should be randomly chosen and kept secret.
2. Calculate 'n' by multiplying 'p' and 'q.' ($n = p * q$)
3. Compute Euler's totient function (φ) of 'n,' which is the number of positive integers less than 'n' that are coprime (having no common factors other than 1) to 'n.' In RSA, $\varphi(n) = (p - 1)(q - 1)$.

4. Select a public exponent 'e' that is coprime to φ(n) (i.e., gcd(e, φ(n)) = 1).
5. Calculate the private exponent 'd' such that (d * e) % φ(n) = 1.

Public key is (n, e)

Private key is (n,d)

Encryption - $c = m^e \pmod n$

Decryption - $m = c^d \pmod n$

2. LITERATURE REVIEW

The Caesar cipher is considered highly insecure for practical use in modern contexts. It is highly vulnerable to cryptographic attacks due to its simplicity. Therefore modern encryption algorithms, such as those used in symmetric key cryptography and asymmetric key cryptography (e.gRSA) use complex mathematical operations and large key spaces, making them resistant to brute-force attacks and other decryption techniques. **Shreyank N Gowda [1]** has proposed a method that combines Diffie-Hellman key exchange and modular arithmetic with the Caesar Cipher to create a more secure encryption algorithm. This approach aims to enhance security without sacrificing the speed of execution, making it a potentially valuable contribution to the field of cryptography. This proposed enhancement includes the implementation and testing of the modified algorithm to assess its effectiveness and security against known cryptographic attacks. **Fahad Naim Nife [2]** created a technique that aims to prevent the occurrence of repetitive terms in a message before encryption. By doing so, the original message is obfuscated in a way that makes it nearly impossible for anyone to retrieve or predict the original content from the encrypted message. The challenge of secure key distribution has been overcome by extracting the key from the encrypted text using a specific method. The algorithm's unique feature lies in its variable key length, which adjusts based on the length of the text, thereby enhancing the overall security of the encryption process.

Manepalli Dharani Pujitha et al [3] aimed at significantly enhancing the security of the Caesar cipher,in this project they have combined the affine cipher and Caesar cipher to explore the advantages of the affine cipher, which help compensate for the shortcomings of the Caesar cipher. This combination results in a more complex and challenging code to solve, significantly increasing the overall security of the encryption. **Pooja Singh et al[4]** has created an enhanced security of Caesar cipher using divide and conquer method, in this approach the ciphertext which is produced can be easily read and understood, ensuring that intended recipients are not suspicious of the encrypted messages. This method employs a key (key) based on the line turnaround results (root), which users can utilize as required. These modifications result in a single substitution, the method is relatively easy to solve for cryptanalysts.

Benni Purnama et al [5] has developed that the ciphertext generated by this method can be read properly, ensuring that specific parties will not find encrypted messages suspicious. It's important to highlight that this method utilizes a single substitution technique, specifically the Monoalphabetic Substitution. In this technique, each letter in the plaintext is consistently replaced by another letter in the ciphertext. While this approach makes the encryption process straightforward, the text implies that cryptanalysts can potentially decipher the ciphertext. However, it's important to note that the method's use of a single substitution technique makes it vulnerable to cryptanalysis, although the readability of the ciphertext adds an element of security by reducing suspicion from third parties.

A. Rajan, D.Balakumaran et al [6] came up with innovative ideas on Caesar cipher in response to advancements in computational capabilities and the need for increased security. The text implies that by rearranging the alphabet, the cipher text is encrypted again. Additionally, a delta formation technique is employed. These combined modifications create an unconditionally secure cipher. By enhancing the Caesar cipher through alphabet rearrangement, additional encryption, and delta formation, the method ensures high security without the burden of excessive computational requirements.

Rajalaxmi Mishra et al [7] developed a hybrid encryption using Caesar cipher and Euler Totient Function. In this method the text is encrypted using a key derived from the specific plain-text being encrypted. This approach eliminates character repetitions, as indicated by frequency analysis. However, the secure distribution of keys remains a challenge, prompting the need for future research in developing secure schemes for key sharing. **Enas Ismael Imran et al [8]** developed the enhancement of the classic Caesar cipher, a simple and commonly used encryption technique. In this method they have used different keys and suggested the incorporation of additional algorithms to further strengthen the security of the encryption method. To explore diverse key types and the integration of multiple algorithms as potential avenues for future research and also aiming to bolster the security measures applied to the Caesar cipher.

Rohit Singh et al [9] has developed a project which highlights the vulnerability of wirelessly transmitted data to third-party interception. The technique is the Caesar cipher,

which helps in advancement of two enhanced versions: the Delta formation Caesar cipher and XOR Caesar cipher. These advancements involve substitutions in the encryption key, reinforcing the security of the caesar cipher and making it more resistant to unauthorized data recovery attempts. **Priya Verma et al[10]** created this based upon ciphertext containing symbols, digits, upper and lower case characters thereby enhancing the cipher's performance. In this technique the key size is increased to 82, significantly expanding the number of possible combinations. As a result, a brute force attacker would need to try 82! combinations to decrypt the message, making it extremely challenging. It helps in extending the lifespan of the cryptography algorithm compared to existing methods, making it considerably more secure.

G.Gomathi Jawahar et al [11] wrote a study based on Caesar cipher encryption and decryption. It states that the security of ciphertext, the encrypted output, depends on both the strength of the encryption algorithm and the key used. In this method proper key management, secure storage, and transmission of ciphertext are highlighted as essential practices to maintain confidentiality and protect against unauthorized access or disclosure.

Sharad Kumar Verma et al[12] created a enciphering mechanism using Caesar cipher where characters are shifted by a fixed number of positions, this new algorithm focuses on space locations within the plaintext and the encryption key is not fixed but changes whenever a space occurs, depending on the space's position in the given text. Decryption is only possible with knowledge of integer values representing space locations in the plaintext, the encryption key, and the counter value. It is mainly prone to frequency analysis.

Table 2.1. Comparison of Existing Methods

S. no	Author (s)	Method	Advantages	Disadvantages
1	Benna Purnama, Hetty Rohayani.AH	Caesar Cipher With Legible Ciphertext	Increases the complexity of the encryption process.	Vulnerable to Dictionary attack, Frequency analysis attack
2	A.Rajan, D.Balakumaran	Advance Caesar Cipher By Randomization And Delta Formation	Security against brute force attack and differential attack	Vulnerable to Frequency analysis attack
3	Fahad Naim Nife	Caesar Cipher Along With Rail Fence To Encrypt.	Secure against differential attack	Vulnerable to Frequency analysis attack, Brute Force attack
4	Shreyank K Gowda	Caesar Cipher Algorithm With Diffie-Hellman	Secure key sharing	Vulnerable to Frequency analysis attack and Dictionary attack
5	Priya verma, Gurjot Singh Gaba, Rajan Miglani	Diversified Caesar Cipher	Secure against brute force attack	Vulnerable to Frequency analysis attack and Dictionary attack

3. HYBRID ENCRYPTION TECHNIQUE TO FORTIFY CAESAR CIPHER

This hybrid encryption algorithm provides a remarkable enhancement in data security by effectively relieving the vulnerabilities of caesar cipher to attacks like frequency analysis attacks and dictionary attacks. This method includes two elements to increase encryption security:-

i. Randomized shift

Instead of using a fixed shift value, a random number generator is employed to generate a new shift value for each encryption operation. This concept of randomizing the shift value in a Caesar cipher adds an element of unpredictability and security to the encryption process, making it more resilient to attacks.

ii. Rivest-Shamir-Adleman

The caesar cipher is susceptible to dictionary attack and frequency analysis attack due to its simplicity. To overcome these attacks we are using RSA encryption, the ciphertext that is generated after the caesar cipher encryption is encrypted using the RSA algorithm. This dual encryption not only adds complexity to the cipher but also makes it difficult for an attacker to deduce the original plaintext.

3.1 PROPOSED SCHEME - ENCRYPTION PHASE

The random package is used to generate random shift values, eliminating the need for user intervention in selecting the shift value. These randomly generated
shifts serve as the basis for encrypting the plaintext using caesar cipher followed by encryption using RSA with public key.

Proposed Encryption Algorithm

Step 1:- Input the plaintext message

Step 2:- Iterate through each letter in the message and shift each letter by the shift value.

Step 3:- Ciphertext1 is generated by caesar cipher.

Step 4:- Double encryption is done by using RSA by taking ciphertext1 as input.

Step 5:- Encrypt ciphertext using public key (n,e) and c = m^e (mod n) to obtain the ciphertext2.

Table 3.1.1. Encryption using proposed method

1.	Plaintext	NETWORK SECURITY
2.	Shift	4
3.	Caesar cipher Encryption	RIXOS VOWING MXC
4.	RSA public key	(3233, 17)
5.	RSA Encryption (final ciphertext)	516 847 3094 46 1332 321 1963 516 558 847 1886 3171 321 75 3094 2220

3.2 PROPOSED SCHEME - DECRYPTION PHASE

In the decryption process, the receiver decrypts the ciphertext using a private key and then uses the random shift value that was used initially in the encryption process to obtain the plaintext.

Proposed Decryption Algorithm

Step 1:- Input the ciphertext message

Step 2:- Decrypt ciphertext using private key (n,d) and m = c^d (mod n) to obtain caesar cipher ciphertext.

Step 3:- Consider the decrypted ciphertext of rsa as input for caesar cipher decryption.

Step 4:- Iterate through each letter in the message and shift back each letter by the shift value to obtain the plaintext.

Table 3.2.1. Decryption using proposed method

1.	Ciphertext	516 847 3094 46 1332 321 1963 516 558 847 1886 3171 321 75 3094 2220
2.	RSA (private key)	(3233, 2753)
3.	RSA Decryption	RIXOS VOWING MXC
4.	Shift	4
5.	Caesar cipher Decryption(plaintext)	NETWORK SECURITY

4. IMPLEMENTATION

Program file is hybrid technique.py

4.1. Functionalities

- **egcd:** It implements the extended Euclidean algorithm to find the greatest common divisor (gcd) of two integers along with the coefficients that satisfy Bézout's identity.
- **gcd:** Calculates the greatest common divisor of two integers using the Euclidean algorithm.
- **modinv:** Calculates the modular multiplicative inverse of an integer. It uses the extended Euclidean algorithm in combination with the egcd function.
- **generate_keys:** It selects two prime numbers and uses few mathematical calculations for generation of the public and private keys for the RSA algorithm.
- **encrypt_caesar:** Encrypts the plaintext using caesar cipher using the randomized shift value.
- **encrypt_rsa:** Encrypts the caesar encrypted text with rsa encryption using public key.
- **decrypt_rsa:** Decrypt the ciphertext using the rsa algorithm with the private key.
- **decrypt_caesar:** Decrypts the rsa decrypted text with caesar cipher using the randomized shift value.

4.2. Attributes: -

- **File:** open the plaintext file in read mode.
- **Plaintext**: Copies the plaintext in bytes and converts text into uppercase text.
- **Shift:** Stores the random integer used as the shift value in the Caesar cipher.
- **Ciphertext1:** Stores the ciphertext generated from the Caesar cipher encryption.
- **Ciphertext2:** Stores the ciphertext generated from the RSA encryption.
- **Pu:** Stores the public key for RSA encryption.
- **Pr:** Stores the private key for RSA decryption.

- **Start:** Marks the start time of the encryption and decryption processes.
- **End:** Marks the end time of the encryption and decryption processes.

4.3. Sample Code

 # key generation for rsa

```
def generate_keys():
    p = 61
    q = 53
    n = p * q
    phi = (p-1)*(q-1)
    e = random.randrange(1, phi)
# Use Euclid's Algorithm to verify that e and phi(n) are coprime
    g = gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = gcd(e, phi)
    d = modinv(e, phi)
    return (e, n), (d, n)
```

 # caesar cipher encryption

```
def encrypt_caesar(plaintext, shift):
    ct = ""
    for char in plaintext:
```

```
        if (char.isupper()):
            ct += chr((ord(char) + shift-65) % 26 + 65)
        else:
            ct += chr((ord(char) + shift - 97) % 26 + 97)
    return ct
# rsa encryption
def encrypt_rsa(plaintext, pu):
    e, n = pu
    lst = []
    for char in plaintext:
        m = ord(char)
        c = pow(m, e) % n
        lst.append(str(c))
    ct = " ".join(lst)
    return ct
# rsa decryption
def decrypt_rsa(ciphertext, pr):
    d, n = pr
    lst = ciphertext.split(' ')
    pt = ''
    for char in lst:
        c = int(char)
```

```
        m = pow(c, d) % n
        pt += chr(m)
    return pt
# caesar cipher decryption
def decrypt_caesar(ciphertext, shift):
    pt = ""
    for char in ciphertext:
        if (char.isupper()):
            pt += chr((ord(char) - shift-65) % 26 + 65)
        else:
            pt += chr((ord(char) - shift - 97) % 26 + 97)
    return pt
```

ENVIRONMENT SCREENSHOTS

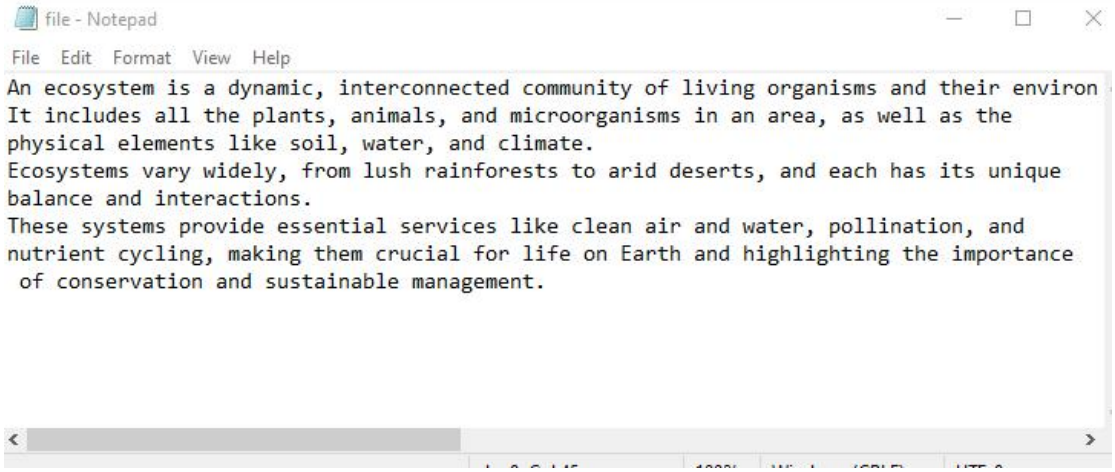

Figure 4.1. Plaintext file

Figure 4.2. Output

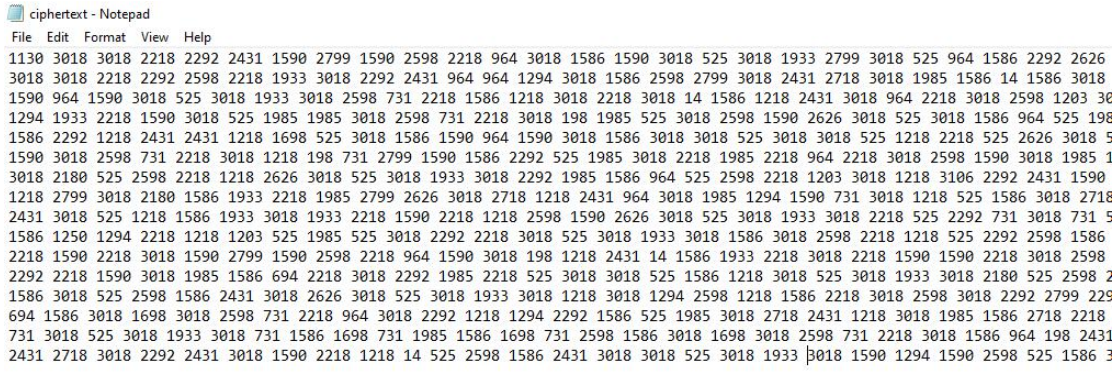

Figure 4.3. Ciphertext file

5. EXPERIMENTAL SETUP

Used **python IDLE** to develop this hybrid encryption technique

5.1. PACKAGES

5.1.1. Random package

The random package in Python is a versatile and essential module for generating random numbers, making decisions based on probability, and creating randomized elements in various applications. It provides functions for generating random integers, floating-point numbers, and sequences, as well as for shuffling lists, selecting random elements, and simulating random events. With its use, Python programmers can implement randomness in games, simulations, statistical analysis, and cryptographic applications, among others, contributing to the unpredictability and variety needed for a wide range of programming tasks. Used to generate random numbers for round key generation.

5.1.2. Time package

The "time" package in Python is a built-in module that provides various functions and methods for working with time-related operations. It allows you to measure time intervals, work with time in both human-readable and machine-readable formats, and perform tasks like creating delays in your programs. The "time" module can be used to access the current time, calculate execution times, format time as strings, and manipulate time in different ways. It's a valuable tool for tasks such as benchmarking code performance, scheduling actions based on time, and handling date and time information in various applications, making it an essential component for developers working on time-sensitive or time-dependent projects in Python. It has been used to determine encryption and decryption time.

Figure. 5.1. Python IDLE

5.2. PARAMETERS

5.2.1. Frequency Analysis Attack

A frequency analysis attack is a classical cryptanalysis technique used to break substitution ciphers, including the Caesar cipher and other similar methods. This attack relies on the fact that in any language, certain letters and characters appear more frequently than others. By analyzing the frequency of these letters in ciphertext, an attacker can make educated guesses about the substitutions and ultimately decrypt the message.

5.2.2. Ciphertext only Attack

A ciphertext-only attack is a type of cryptographic attack in which the attacker has access to one or more pieces of ciphertext (encrypted text) but does not have access to the corresponding plaintext (original message) or any other information about the

encryption process. The goal of a ciphertext-only attack is to deduce or recover the plaintext or the encryption key used to create the ciphertext.

5.2.3. Dictionary Attack

A dictionary attack is a type of cyberattack that involves systematically attempting to guess words or encryption keys by trying a large number of potential combinations, often using a list of commonly used words or known words from a dictionary.

6. DISCUSSION OF RESULT

The proposed method provides a remarkable security to confidential data, which is obtained mainly through encryption with RSA. This hybrid method is capable of resisting well-known cryptographic attacks like frequency analysis attack, dictionary attack and many more.

6.1. FREQUENCY ANALYSIS ATTACK

In the below images, it is evident that the existing encryption method is prone to frequency analysis attacks. The ciphertext can be deduced by the patterns of the letters. However, when we compared the existing method results with the proposed method results, a remarkable difference is apparent. In the proposed method, it can be seen that the ciphertext obtained is a numerical text which makes it challenging to conduct a frequency analysis test.

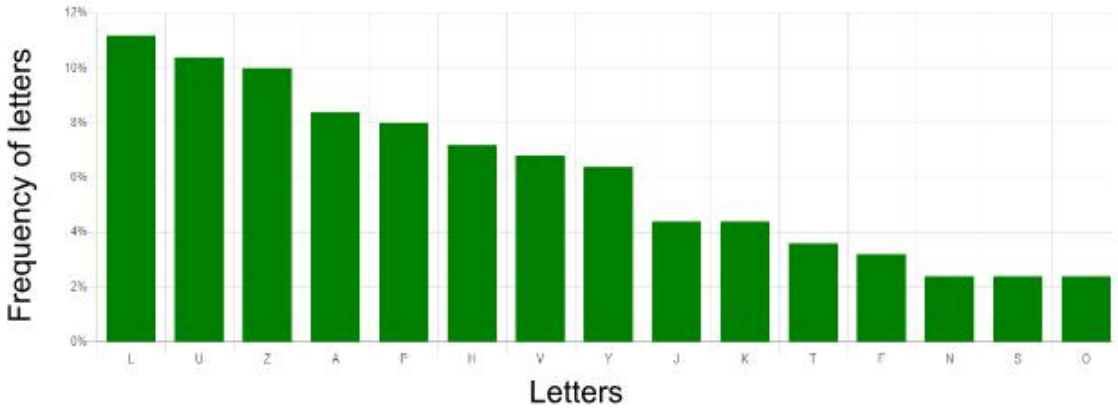

Figure 6.1. Frequency analysis of existing method

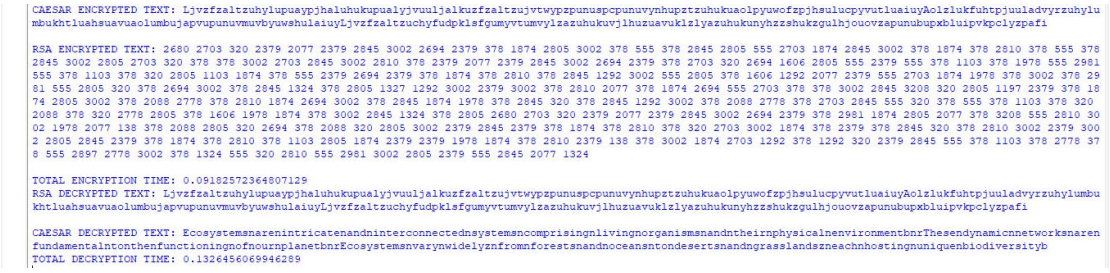

Figure 6.1. Frequency analysis Proposed method

6.2. CIPHERTEXT ONLY ATTACK

The existing method is prone to ciphertext only attack as the attacker has access to the ciphertext and since there are only 25 possible keys (shift values), an attacker can systematically try all possible shifts to deduce the plaintext or the encryption key used to create the ciphertext. But the proposed method overcomes the attack as even if the attacker has access to the ciphertext, the private key of the receiver is required to decipher the ciphertext.

6.3. DICTIONARY ATTACK

The existing method is vulnerable to dictionary attack as an attacker tries to decipher the ciphertext by trying all the known word combinations from a dictionary. In our proposed method the attack has been overridden as the ciphertext derived in our method is a numerical text.

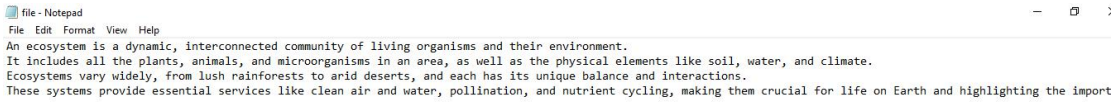

Figure 6.3. Existing Method

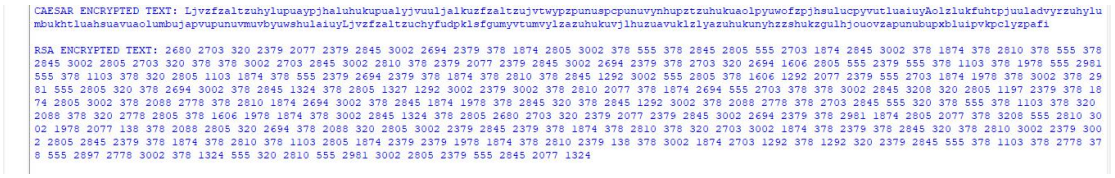

Figure 6.3. Proposed Method

Table 6.1 Comparison of results

Parameters	**Existing Value**	**Improved Value**	**Justification**
Frequency analysis attack	Textual data	Numerical data	The attack has been overcome as the data is converted into numeric format.
Ciphertext Only text	Accurate guess of key length or plaintext	Inaccurate guess of key length or plaintext	The attacker needs the private key of the receiver to decipher
Dictionary attack	Textual data	Numerical data	This attack cannot be performed on numeric data.

7. SUMMARY, CONCLUSION AND RECOMMENDATION

The proposed hybrid encryption technique provides protection against frequency analysis attacks and dictionary attacks, common methods employed by attackers to decipher encrypted data. By incorporating a randomized shift and dual encryption through RSA, the technique makes it significantly challenging for an attacker to determine the original plaintext.Although this approach may result in slightly increased encryption and decryption times, the benefits of the hybrid technique are deemed to be significantly advantageous, outweighing the minor performance trade-offs. By utilizing this approach, the security of sensitive data is bolstered, providing a robust defense mechanism against potential cryptographic attacks.

8. REFERENCES

[1] Shreyank N Gowda, “Innovative enhancement of the Caesar cipher algorithm for cryptography,” “2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)”, Bareilly, India, 2020, DOI: 10.1109/ICACCAF.2016.7749010

[2] Fahad Naim Nife “A New Modified Cesar Cipher Cryptographic Method Along With Rail Fence to Encrypt Message”, “International Journal of Advanced Research (2015)”, vol. 3, 2015, pp. 331-335

[3] Manepalli Dharani Pujitha “Security Enhancement Using Caesar Cipher”.

[4] Pooja Singh and Pintu Sen “Enhancing Security Of Caesar Cipher Using Divide And Conquer Approach.”

[5] Benni Purnamaa, Hetty Rohayani.AH “A New Modified Caesar Cryptography Method With LegibleCiphertext From a Message to be Encrypted” International Conference on Computer Science and Computational Intelligence (ICCSCI), Indonesia, 2015, DOI:10.1016/j.procs.2015.07.552

[6] R. Mishra and D. J. K. Mantri, "An Enhancement to Caesar Cipher using Euler Totient Function," “International Journal of Engineering and Advanced Technology (IJEAT)”, vol. 11, no. 3, pp. 3363-3372, February 2022. DOI: 10.35940/ijeat.c3363.0211322.

[7] A. Rajan, D.Balakumaran “Advancement In Caesar Cipher By Randomization And Delta Formation”, International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India, 2015, DOI: 10.1109/ICICES.2014.7033998

[8] Enas Ismael Imran, Farah abdulameer abdulkareem “Enhancement Caesar Cipher for Better Security”, vol. 16, pp. 01-05, 2014.

[9] Singh, Rohit & Kumar, Naveen. (2018). A Review Paper on Cryptography of Modified Caesar Cipher.

[10] R. Miglani, "Diversified Caesar Cipher for Impeccable Security," “International Journal of Security and its Applications”, vol. 11, pp. 33-40, 2017. DOI: 10.14257/ijsia.2017.11.02.04.

[11] G. G. . Jawahar, D. S. . Anto, T. J. . Thomas, Krishnendu, and M. . Jousva, “A Study on Encryption and Decryption using the Caesar Cipher Method", *Int J Intell Syst Appl Eng*, vol. 11, no. 3, pp. 357–360, Jul. 2023.

[12] Sharad Kumar Verma and Dr. B. Ojha “An Innovative Enciphering Scheme Based On Caesar Cipher.”International Journal of Innovative Science, Engineering & Technology, vol. 1, no. 5, July 2014.

[13] G. Swain, Object-Oriented Analysis and Design Through Unified Modeling Language. Laxmi Publication, Ltd., 2010.

[14] G. Booch, D. L. Bryan, and C. G. Peterson, Software engineering with Ada Redwood city, Calif.: Benjamin/Cummings, 1994.

[15] J. M. Zelle, Python Programming: an introduction to computer science. Portland, Oregon: Franklin, Beedle & Associates Inc, 2017.