International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Study on Encryption and Decryption using the Caesar Cipher Method

G. Gomathi Jawahar[1], D. Silvester Anto[2],T. John Thomas[3], Krishnendu[4], Melvin Jousva[5]

Submitted: 27/04/2023 **Revised**: 25/06/2023 **Accepted**: 08/07/2023

Abstract: The methodology of ciphertext in Python depends on the specific encryption algorithm used. The ciphertext is the result of applying an encryption algorithm to plaintext, typically involving various text processing and wrapping operations. Here the ciphertext is the encrypted form of the plaintext and is intended to keep the information confidential and securefrom unauthorized access.

Keywords: *The Caesar Cipher, Encryption, Ciphertext, Python, cryptography.*

1.Introduction:

The Caesar Cipher Method is an ancient and widely used cipher text method that is easy to encrypt and decrypt. It works by shifting the letters of the alphabet over to create an entirely new alphabet. Caesar Ciphers are not the most secure ciphers text method but they are good for small tasks such as passing secret notes or making passwords a little stronger. It is really easy to decrypt the code, but it can be tedious to encrypt one if youdon't have the special alphabet memorized.To make this process easier, we can use the power of computers, more specifically the programming language like Python. Julius Caesar, who employed it in his communications, gave the approach its name. [4]Dharmendra K. 2012, studied the new concept of symmetric encryption algorithm a hybrid approach of caesar cipher and columnar transposition in multi stages. [5]Prachi P. 2013, studied A Poly-alphabetic Approach to Caesar Cipher Algorithm. International Journal of Computer Science and Information Technologies(IJCSIT), Vol. 4 (6) , 2013, 954-959 . [6]Bhardwaj C. 2012, studied Modification of Vigenère Cipher by Random Numbers, Punctuations & Mathematical Symbols. Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 4, Issue 2 (Sep.-Oct. 2012), PP 35-38 Available at: www.iosrjournals.org With this encryption method, each letter in the text must be changed for a certain difference to encrypt our data.It operates by shifting each letter in the alphabet . For example, with a key of 4, the letter 'A' would be replaced by 'E', 'B' would become 'F', 'C' would become 'G', and so on. This shift canbe applied in both the encryption and decryption method of using the same algorithm.The implementation of this method is relatively straightforward. It can be achieved using basic string manipulation and arithmetic operations. The "ord()" and "chr()" functions in Python can be used as corresponding ASCII values, enabling the shifting of letters in the alphabet. The "%" operator can be used to wrap around the alphabet, ensuring that the shifted letters stay within the valid range of lowercase or uppercase letters.The Caesar Cipher has some limitations.

One major weakness is its vulnerability to brute force attacks, where an attacker systematically tries all possible key values until the correct one is found. With only possible key values (since the key has particular range), the Caesar Cipher can be easily cracked through trial and error.The Caesar Cipher does not provide any security against frequency analysis attacks, where an attacker analyzes the frequency of letters in the encrypted text to deduce the plain messageAlternatively, the Caesar Cipher can still be useful in many situations where a basic level of encryption is needed, or as an educational tool to understand the concepts of cryptography. It plays a major role for more complex encryption algorithms, such as the Vigenère Cipher and the modern-day Advanced Encryption Standard (AES).

2.Problem Statement:

The problem statement of ciphertext is its context of cryptography, is that it is an encrypted form of a message that is not easily understandable without proper decryption using the correct decryption algorithm. Ciphertext is typically the result of applying an encryption algorithm to plaintext, which is the original text , unencrypted message. Thepurpose of encryption is to protect the confidentiality (which means keeping information private and protected from unauthorized access or disclosure.) and integrity of information, ensuring that only authorized parties can access and understand the content of the message.The problem with ciphertext is that it appears as a unintelligible sequence of characters, making it difficult or impossible to decipher without the proper decryption algorithm. This poses challenges in communication, as recipients of ciphertext

[1]*Karunya Institute of Technology and Sciences,Coimbatore.*
ORCID ID : 0000-0002-0361-7116
[2,3,4,5]*Karunya Institute of Technology and Sciences,Coimbatore.*

need to have the correct decryption key or algorithm in order to decode and understand the original message. Without the correct key, ciphertext can be effectively unreadable, providing a layer of security to protect sensitive information from unauthorized access.But it presents challenges in securely managing and sharing decryption keys, as the loss of the key can result in the ciphertext being rendered unreadable indefinitely.the strength of ciphertext depends on the encryption algorithm used and the length and randomness of the encryption key. Weak encryption algorithms predictable keys can be susceptible to attacks, such as brute force attacks or cryptographic attacks, which can potentially compromise the confidentiality of the encrypted information.In conclusion, the problem statement of ciphertext is that it presents a challenge in securely communicating and understanding encrypted messages without the proper decryption key.The confidentiality and integrity of the information protected by ciphertext depend on the strength of the encryption algorithm and the secrecy and robustness of the encryption key. Ensuring the proper management and protection of decryption keys is critical in maintaining the security of ciphertext and preventing unauthorized access tosensitive information.

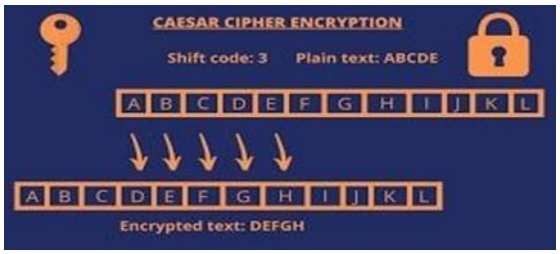
3.Methodology / Architecture

INPUT: The original text and the key (the fixed number of positions to shift) are takenas input from the user

TEXT PROCESSING: The plaintext can also remove any non-alphabetic characters (e.g., spaces) and convert it to uppercase or lowercase, according to the desired case for the ciphertext**ENCRYPTION ALGORITHM**: The encryption algorithm also involves in iterating through each letter in the plaintext and applying the shift operation based on the key value. This can be achieved using basic string manipulation and arithmetic operations in Python.

SHIFTING: The letters in the plaintext are shifted by the key value, either forward orbackward in the alphabet by using the ord() and chr() functions in Python to convert between characters and their ASCII values.

WRAPPING: The shifted letters are wrapped around the alphabet using the "%" operator to ensure that they can stay within the range of uppercase or lowercase letters.



CIPHERTEXT GENERATION: The shifted letters are concatenated (add)to form the ciphertext, which is the encrypted form of the plaintext.

OUTPUT: The ciphertext is displayed or saved to a file, depending on the output format.

PROGRAM INPUT

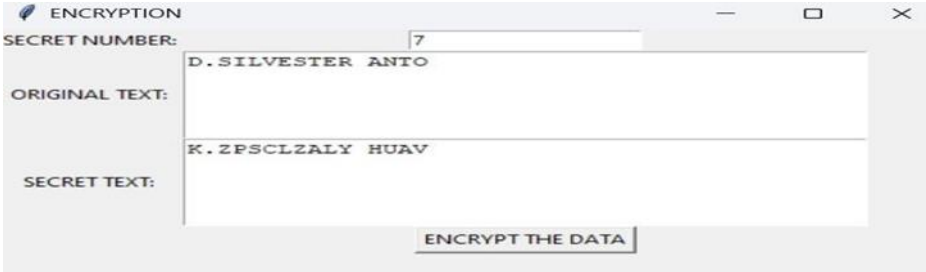
```
    plain_text.grid(row=1, column=1)
    Label(window, text="SECRET TEXT:").grid(row=2, column=0)
    cipher_text = Text(window, height=5, width=45)
    cipher_text.grid(row=2, column=1)
    Label(window, text="SECRET NUMBER:").grid(row=0, column=0)
    key_entry = Entry(window)
    key_entry.grid(row=0, column=1)
    def decrypt_text():
        key = int(key_entry.get())
        text = cipher_text.get("1.0", "end-1c")
        decrypted_text = ""
        for char in text:
            if char.isalpha():
                new_char_code = ord(char) - key
                if char.isupper():
                    if new_char_code > ord('Z'):
                        new_char_code -= 26
                    elif new_char_code < ord('A'):
                        new_char_code += 26
                elif char.islower():
                    if new_char_code > ord('z'):
                        new_char_code -= 26
                    elif new_char_code < ord('a'):
                        new_char_code += 26
                decrypted_text += chr(new_char_code)
            else:
                decrypted_text += char
        plain_text.delete("1.0", "end")
        plain_text.insert("1.0", decrypted_text)
    decrypt_button = Button(window, text="DECRYPT THE DATA", command=decrypt_text)
```

```
decrypt_button = Button(window, text="DECRYPT THE DATA", command=decrypt_text)
decrypt_button.grid(row=3, column=1)
b1 = Button(window, command=encyption, text="ENCRYPTION", width='13',fg="RED")
b2 = Button(window, command=decyption, text="DECRYPTION", width='13',fg="BLUE")
b1.pack()
b2.pack()
window.mainloop()
```

```
        for char in text:
            if char.isalpha():
                new_char_code = ord(char) + key
                if char.isupper():
                    if new_char_code > ord('Z'):
                        new_char_code -= 26
                    elif new_char_code < ord('A'):
                        new_char_code += 26
                elif char.islower():
                    if new_char_code > ord('z'):
                        new_char_code -= 26
                    elif new_char_code < ord('a'):
                        new_char_code += 26
                encrypted_text += chr(new_char_code)
            else:
                encrypted_text += char
        cipher_text.delete("1.0", "end")
        cipher_text.insert("1.0", encrypted_text)
    encrypt_button = Button(window, text="ENCRYPT THE DATA", command=encrypt_text)
    encrypt_button.grid(row=3, column=1)

def decyption():
    window = Tk()
    window.title("DECRYPTION")
    window.geometry('505x250')
    Label(window, text="ORIGINAL TEXT:").grid(row=1, column=0)
    plain_text = Text(window, height=5, width=45)
    plain_text.grid(row=1, column=1)
    Label(window, text="SECRET TEXT:").grid(row=2, column=0)
    cipher_text = Text(window, height=5, width=45)
```

PROGRAM OUTPUT



4. Conclusion

In conclusion, the methodology of ciphertext in Python depends on the specific encryption algorithm used. The ciphertext is the result of applying an encryption algorithm to plaintext, typically involving various text processing and wrapping operations. The ciphertext is the encrypted form of the plaintext and is intended to keep the information confidential and secure from unauthorized access.

The architecture of ciphertext generation in Python involves taking input of plaintext and key,processing the plaintext, applying the encryption algorithm, and generating the ciphertext. The ciphertext is typically

displayed or saved to a file as the output. The security of the ciphertext depends on the strength of the encryption algorithm and the key used.

Confidentiality is a crucial aspect of cryptography, and ciphertext is the result of encryption that ensures the confidentiality of sensitive information. Additionally, There are some features of proper key management, secure storage, and transmission of the ciphertext are essential to maintain confidentiality and protect against unauthorized access or disclosure.

References:

[1] Srikantaswamy S and Phaneendra H. 2012, "Improved Caesar cipher with random number generation technique and multistage encryption". Published by International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.4, December 2012

[2] CSI/FBI 2007, Computer Crime and Security Survey, 9-14-2007. Content Posted by New Media Institute (NMI) Editor. Available at: http://en.wikipedia.org/wiki/Encryption

[3] Rigoberto G. 2009, Using Classical Ciphers in Secondary Mathematics. BSc. Thesis McMurry University Abilene, Texas 2009

[4] Dharmendra K. 2012, New concept of symmetric encryption algorithm a hybrid approach of caesar cipher and columnar transposition in multi stages. Journal of Global Research in Computer Science (JGRCS) Copyright Agreement & Authorship Responsibility Vol 3, No. 1 (2012). Available at:http://www.jgrcs.info/index.php/jgrcs/article/view/295

[5] Prachi P. 2013, A Poly-alphabetic Approach to Caesar Cipher Algorithm. International Journal of Computer Science and Information Technologies(IJCSIT), Vol. 4 (6) , 2013, 954-959

[6] Bhardwaj C. 2012, Modification of Vigenère Cipher by Random Numbers, Punctuations & Mathematical Symbols. Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 4, Issue 2 (Sep.-Oct. 2012), PP 35-38 Available at: www.iosrjournals.org

[7] Ms. Nora Zilam Runera. (2014). Performance Analysis On Knowledge Management System on Project Management. International Journal of New Practices in Management and Engineering, 3(02), 08 - 13. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/28

[8] Kumar, L. R. ., Ashokkumar, C. ., Pandey, P. S. ., Kannaiah, S. K. ., J, B. ., & Hussan, M. I. T. . (2023). Security Enhancement in Surveillance Cloud Using Machine Learning Techniques. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 46–55. https://doi.org/10.17762/ijritcc.v11i3s.6154