श्रद्धावान लभते ज्ञानम्

INSTITUTE OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Project Title:
Decrypting Caesar Cipher

Team Member Name:	Roll No:	Enrollment No:
Debjyoti Ghosh	159	12020002016020
Abhirup Mazumder	161	12020002017003
Priyanjana Saha	166	12020002018002
Srijan Pal	167	12020002018006

Table of Contents

Abstract:

The simplest and oldest form of encryption is the **Caesar Cipher**. The Caesar Cipher technique, also known as a shift cipher or additive cipher, is based on a mono-alphabetic cipher. To communicate with his soldiers, Julius Caesar employed the shift cipher (also known as the additive cipher) method. The shift cipher method is known as the Caesar cipher for this reason. A type of replacement (substitution) encryption, the Caesar Cipher substitutes one letter for each one in plain text.

The project involves the use of **Python** to develop a software that can encrypt and decrypt a given text based on either the encryption procedure or as per as the user’s choice. The system supports **encryption** techniques such as : a standard encryption with a key value of 3, an encryption with a key value ranging from 0 to 25 and an encryption with a key of choice from the user and **decryption** techniques like: a standard decryption with a key value of 3, a decryption table with keys ranging from 0-25, a decryption with a user key and **smart decryption**.

Moving on from the functionalities occuring in the backend, the **GUI** of the system is also made presentable for the users. The frontend has been designed in python using CustomTkinter module that makes the UI user-friendly and easy-to-use.

Overall, the software system is informative, user-friendly and feasible given it is utilising the concept of Object Oriented Programming, GUI designing and Open Source.

Keywords: Caesar Cipher, python, encryption, decryption, GUI, smart decryption

Introduction:

As the importance and the value of exchanged data over the Internet or other media types are increasingly, the search for the best solution to offer the necessary protection against the data thieves' attacks along with providing these services in a timely manner is one of the most active subjects in the security related communities. The primary goal of any system is that the data cannot be modified by any external user or intruder. To avoid such a situation convert data into a non-readable form at sender side and convert that data in readable form again at receiver side. The technique and science of creating non-readable data or cipher so that only authorised persons are only able to read the data is called Cryptograph. It is the art and science of protecting information from undesirable individuals by converting it into a form non-recognizable by its attackers while stored and transmitted. In Cryptography, Caesar cipher is one of the most widely known encryption decryption algorithms. Caesar cipher is a type of substitution type cipher in this kind of cipher each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. The encryption is represented using modular arithmetic.

The main aim of a Caesar Cipher text is to encrypt a given text by shifting the alphabets a few keys(steps) forward or backward. The system which is designed in this project not only encrypts data but also provides several decryptions using which we can decipher the hidden text.

The formula used to encrypt a normal text is:

$$E_n (x) = (x + n) \bmod 26$$

And, the formula used to decrypt the ciphertext is:

$$D_n (x) = (xi - n) \bmod 26$$

The Project intends to satisfy the following objectives:

1. Implement methods of encryption and decryption using Caesar Cipher algorithm
2. Develop a prototype that helps to test the output from the encryption-decryption process
3. Optimising the workflow with minimal use of the computing system
4. Enhance the user experience with responsive UI

The following scopes are established in the project:

1. Caesar cipher algorithm is applied to increase the strength of security.
2. Create encryption and decryption method for the proposed encryption-decryption.

Methodology:

The system has been built on a python environment.
The project can be roughly divided into 7 phases:

1. Standard Encryption [3]
2. Encryption with Random Key [0-25]
3. Encryption with User Key
4. Standard Decryption [3]
5. Decryption Table [0-25]
6. Decryption with User Key
7. SMART Decryption

Standard Encryption [3]:

In case of Standard Encryption, we define the key to be 3. This serves as a user shortcut to encrypt the text quickly without having the need to remember the key. The function used in such a case is:

$E_n(x) = (x + 3) \bmod 26$

Encryption with Random Key [0-25]:

When working with encryption using a random key, we take any random key generated by the system to encrypt our text. This enables the user to encrypt their text with a system generated key and relieve them of the stress of choosing a key. The function used in such a case is:

$E_n(x) = (x + \text{random}(0,25)) \bmod 26$

Encryption with User Key:

User can opt to encrypt a text with a key of their choice. The function used in such a case is:

$E_n(x) = (x + n) \bmod 26$, where n is user choice key

Standard Decryption [3]:

In the case of Standard Decryption, the key defined by us is 3. This is a user shortcut for fast decrypting text without having to remember the key. The function used in such a case is:

$D_n(x) = (xi - 3) \bmod 26$

Decryption Table [0-25]:

This option is particularly useful to display all possible combinations of decryption for an encrypted message. A table of decrypted messages along with respective keys is displayed in a .txt file. The function used in such a case is:

$D_n(x) = (xi - n) \bmod 26$, where n belongs to [0,25]

Decryption with User Key:

The user has the option of decrypting a text with a key of their choice. The function used in such a case is:

$D_n(x) = (x_i - n) \bmod 26$, where n is a user choice key

SMART Decryption:

This option uses a python module called langdetect which uses language models to identify the language in which a message is constructed. The idea is simple. Our application uses the decryption table method to find all the combinations of decrypted message for a particular encrypted form of message. We maintain a certain threshold above which if the confidence for English language comes up to be, we provide the user with the possible key. The python module allow us to derive messages for other languages as well but for now, we have kept it limited to English.

Modules used: tkinter, customtkinter, random, os, langdetect

Functions used:

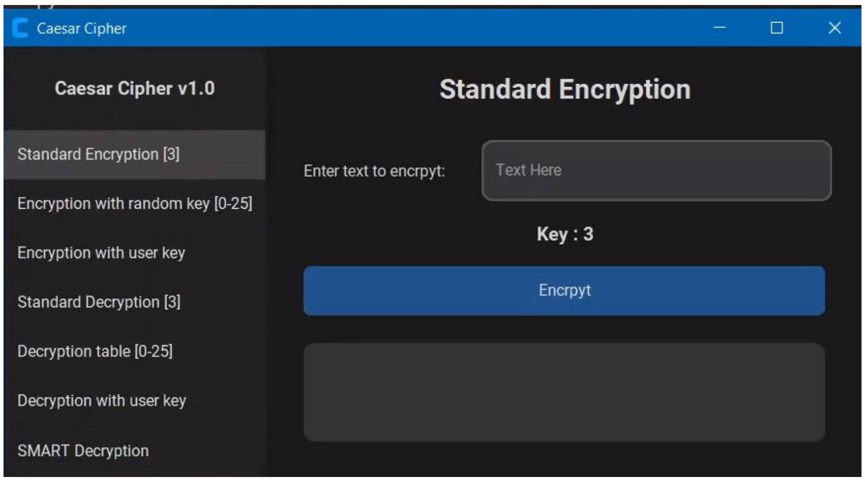
select_frame_by_name(): Used to select frame using GUI options
button_event(): Used to map buttons with event
encryption(): Used to encrypt a parameter message using a parameter key
decryption(): Used to decrypt a parameter message using a parameter key
standardenc(): Used to implement Standard Encryption
randomenc(): Used to implement Random Key Encryption
randomize(): Used to pick a random key
userenc(): Used to implement User Key Encryption
standarddec(): Used to implement Standard Decryption
decrypttable(): Used to implement Decryption Table
openfile(): Used to open the decrypted message files
userdec(): Used to implement User Key Decryption
smartdec(): Used to implement Smart Decryption System

Moreover, the software built is completely dependent on the code and is hence machine independent. That is, the software can run on different operating systems and there won't be any issues with the latency because there is no runtime overhead due to data storage. It is compatible with changes in a personal computer and the interface is responsive.

The project has been pushed to a Github repository:
https://github.com/srijanpal1405/encryptDecryptCaesarCipher.git

Workflow:

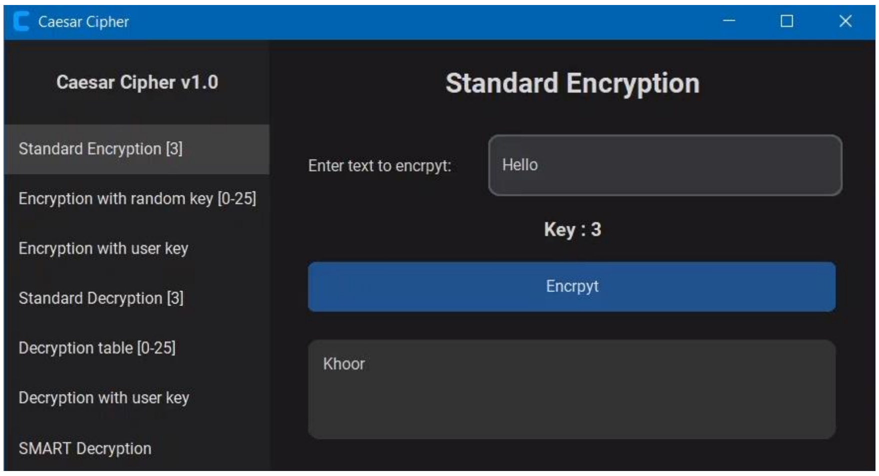
- The user runs the encryptdecrypt.py and the following tab pops up:



Enter text to encrpyt: Text Here

Key : 3

- The user then has an option of using standard encryption, encryption with random key or encryption with user key
  1. If a user uses standard encryption, the text is entered and the encrypt button is pressed to generate the cipher text.

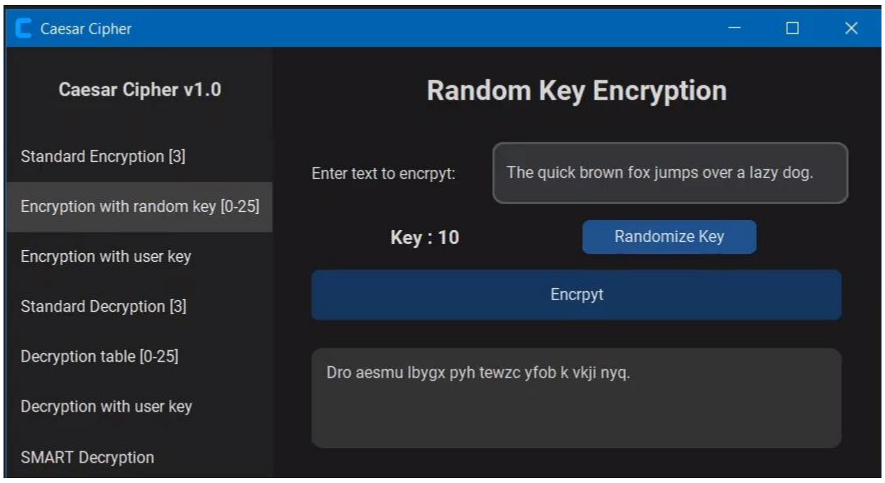


Enter text to encrpyt: Hello

Key : 3

Khoor

  2. If a user uses encryption with a random key, the system will generate a key for the text. After pressing the encrypt button, the text gets encrypted and cipher text shows on the space below:

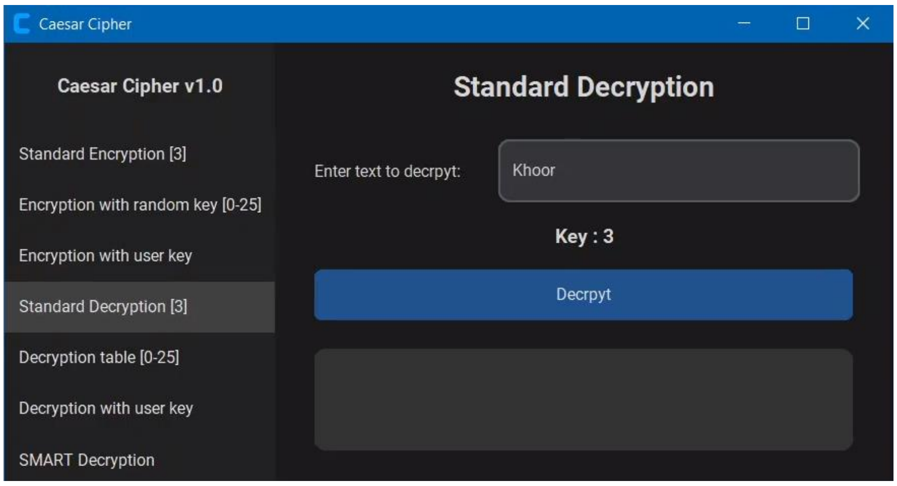


Enter text to encrpyt: The quick brown fox jumps over a lazy dog.

Key : 10

Dro aesmu lbygx pyh tewzc yfob k vkji nyq.

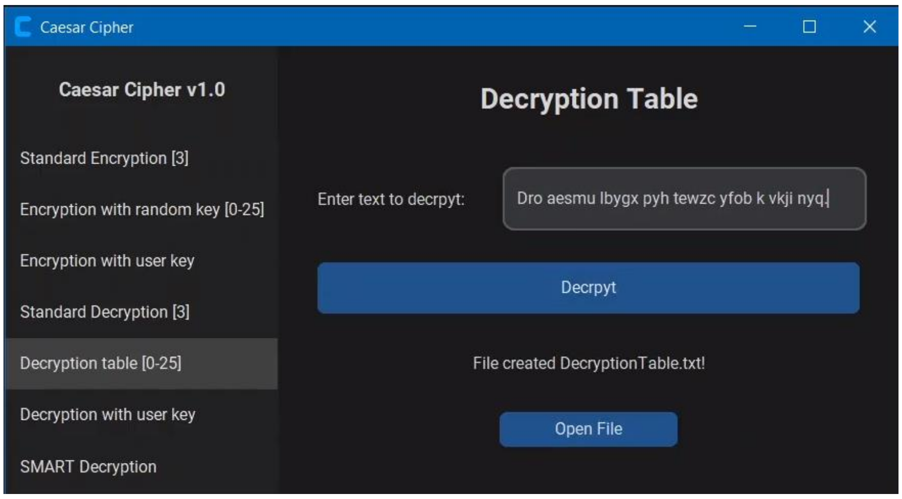
- User then moves to the choice of decryption. Users can decrypt in 4 possible ways which are standard decryption, decryption using a decryption table, decryption using user ways or smart decryption.
    1. If a user chooses standard decryption, the user enters the cipher text and press the decrypt button to generate the normal text.



Enter text to decrpyt: Khoor

Key : 3

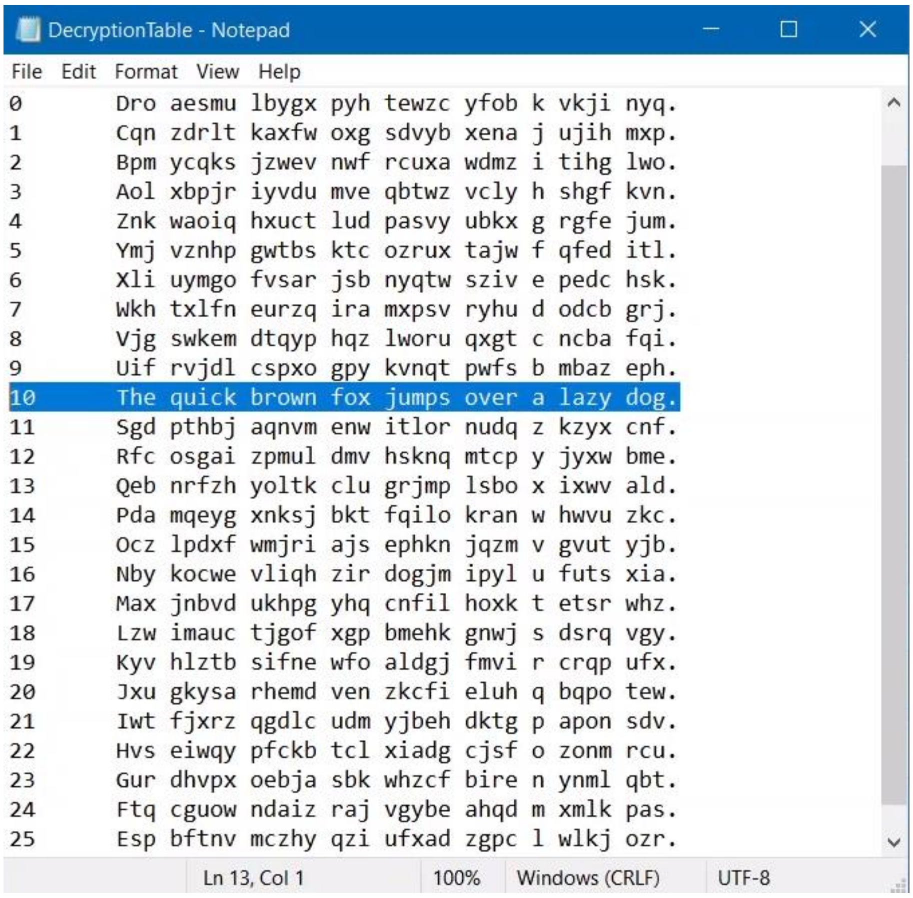
    2. If a user chooses to decrypt using a decryption table, then a decryption file gets generated which contains a number of text in different lines and the original text is in the line number which is equivalent to the key.



Decryption Table

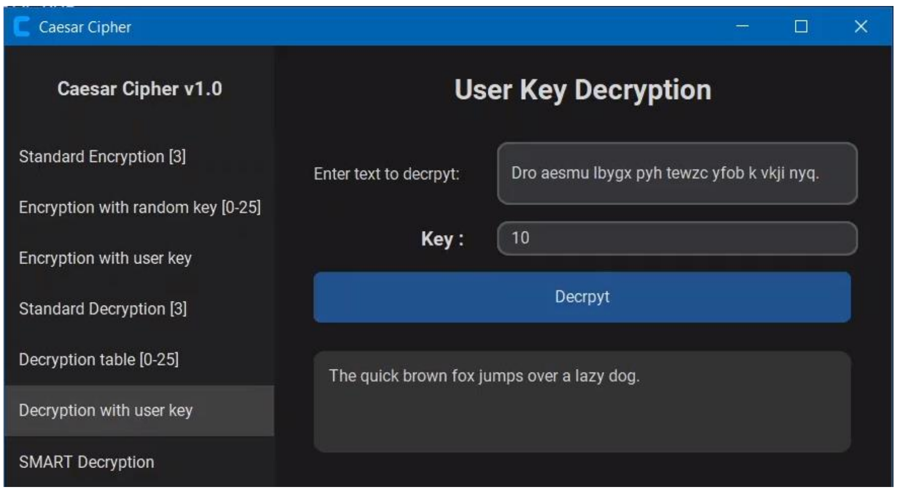
Enter text to decrpyt: Dro aesmu lbygx pyh tewzc yfob k vkji nyq.

File created DecryptionTable.txt!



```
0       Dro aesmu lbygx pyh tewzc yfob k vkji nyq.
1       Cqn zdrlt kaxfw oxg sdvyb xena j ujih mxp.
2       Bpm ycqks jzwev nwf rcuxa wdmz i tihg lwo.
3       Aol xbpjr iyvdu mve qbtwz vcly h shgf kvn.
4       Znk waoiq hxuct lud pasvy ubkx g rgfe jum.
5       Ymj vznhp gwtbs ktc ozrux tajw f qfed itl.
6       Xli uymgo fvsar jsb nyqtw sziv e pedc hsk.
7       Wkh txlfn eurzq ira mxpsv ryhu d odcb grj.
8       Vjg swkem dtqyp hqz lworu qxgt c ncba fqi.
9       Uif rvjdl cspxo gpy kvnqt pwfs b mbaz eph.
10      The quick brown fox jumps over a lazy dog.
11      Sgd pthbj aqnvm enw itlor nudq z kzyx cnf.
12      Rfc osgai zpmul dmv hsknq mtcp y jyxw bme.
13      Qeb nrfzh yoltk clu grjmp lsbo x ixwv ald.
14      Pda mqeyg xnksj bkt fqilo kran w hwvu zkc.
15      Ocz lpdxf wmjri ajs ephkn jqzm v gvut yjb.
16      Nby kocwe vliqh zir dogjm ipyl u futs xia.
17      Max jnbvd ukhpg yhq cnfil hoxk t etsr whz.
18      Lzw imauc tjgof xgp bmehk gnwj s dsrq vgy.
19      Kyv hlztb sifne wfo aldgj fmvi r crqp ufx.
20      Jxu gkysa rhemd ven zkcfi eluh q bqpo tew.
21      Iwt fjxrz qgdlc udm yjbeh dktg p apon sdv.
22      Hvs eiwqy pfckb tcl xiadg cjsf o zonm rcu.
23      Gur dhvpx oebja sbk whzcf bire n ynml qbt.
24      Ftq cguow ndaiz raj vgybe ahqd m xmlk pas.
25      Esp bftnv mczhy qzi ufxad zgpc l wlkj ozr.
```

3. If a user chooses for decryption with user key,the user enters the cipher text as well as the key value and the encrypted text is shown

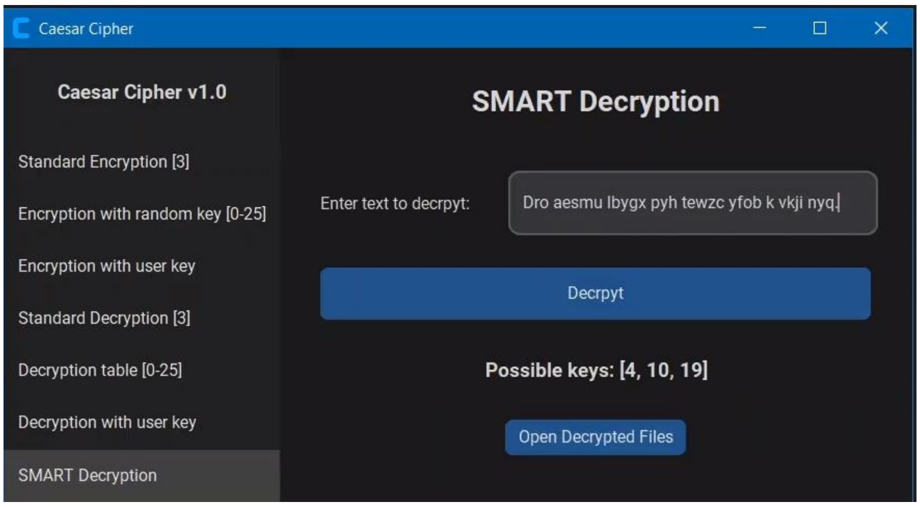


Enter text to decrpyt: Dro aesmu lbygx pyh tewzc yfob k vkji nyq.

Key : 10

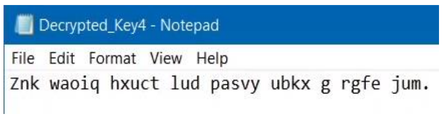
The quick brown fox jumps over a lazy dog.

4. If a user chooses for smart decryption, the algorithm will generate certain files using a machine learning model which have the certainty of the decrypted file to be present

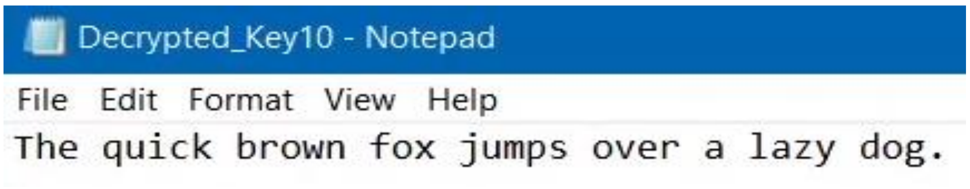


Enter text to decrpyt: Dro aesmu lbygx pyh tewzc yfob k vkji nyq.

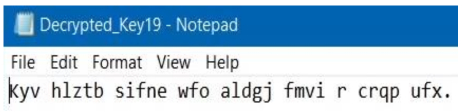
Possible keys: [4, 10, 19]



```
Znk waoiq hxuct lud pasvy ubkx g rgfe jum.
```



```
The quick brown fox jumps over a lazy dog.
```



```
Kyv hlztb sifne wfo aldgj fmvi r crqp ufx.
```

Conclusion:

In this research, we have demonstrated how Caesar cipher, one of the most extensively used and straightforward encryption methods, may be strengthened beyond the capabilities of the basic Caesar cipher algorithm. Based on Caesar cipher algorithms, numerous techniques are employed for security purposes. In the future, we can combine different key types into a single technique, and we can increase the security by including more algorithms.

The key generations are essential in the cipher design. Modified Caesar cipher is presented in this study. A replacement cipher is used. Internet and network usage are expanding quickly. Therefore, there are additional requirements for protecting data exchanged over various networks and using various services. Different encryption techniques are employed to protect the network and data. The Caesar cipher technique is employed in this paper for security. It is distinctive in its own right. If more than one method is used to data, the security provided by this technique can be further improved. This idea will be investigated in more detail in the future work, and a number of algorithms will be used on data to provide a more safe environment for data storage and retrieval.

Acknowledgement:

An endeavour over a long period can be advice and support of many well- wishers. We take this opportunity to express our gratitude and appreciation to all of them. We owe our tributes to **Dr. Indraneel Mukhopadhyay**, Head of the Department, Computer Science and Engineering for his valuable support and guidance during the period of project implementation.

Project meetings were highly informative. We express our sincere thanks for the encouragement, untiring guidance and the confidence they had shown in us. We are immensely indebted for their valuable guidance throughout our project.

References:

The following resources have been referred to:

- https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/
- https://www.khanacademy.org/computing/computer-science/cryptography/crypt/v/caesar-cipher
- https://www.javatpoint.com/caesar-cipher-technique
- https://www.dcode.fr/caesar-cipher
- Cyber Security by Nina Godbole