AN IMPLEMENTATION OF VARIOUS EFFICIENT METHODS OF ENCRYPTION(previously cipher decryption tool with advanced language checker)

A PROJECT REPORT

Submitted by
BATCH - 2

Darshni Gupta (20BCY10017)
Vibhu Kapadia (20BCY10024)
Anamika Mittal (20BCY10084)
Parthivi Sahu (20BCY10188)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING WITH SPECIALISATION IN CYBER SECURITY AND DIGITAL FORENSICS



SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRI KALAN, SEHORE
MADHYA PRADESH - 466114

DEC 2021

BONAFIDE CERTIFICATE

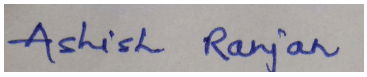
Certified that this project report titled **"AN IMPLEMENTATION OF VARIOUS EFFICIENT METHODS OF ENCRYPTION"** is the bonafide work of "**Darshni Gupta**(20BCY10017), **Vibhu Kapadia**(20BCY10024), **Anamika Mittal**(20BCY10084) and **Parthivi Sahu**(20BCY10188)**"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. R. Rakesh

Program Chair - B.Tech Cyber Security and Digital Forensics,
Assistant Professor,
School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Mr. Ashish Ranjan



Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination was held on 17.12.2021.

ACKNOWLEDGEMENT

First and foremost we would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

We wish to express our heartfelt gratitude to Dr R. Rakesh., Program Chair, B. Tech Cyber Security and Digital Forensics for much of his valuable support and encouragement in carrying out this work.

We would like to thank our internal guide Mr. Ashish Ranjan, for continually guiding us, actively participating in our project and giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

ABSTRACT

In this computer and internet based modern era, people have to deal with private information in thousands of ways. Today's prime concern is to keep secure the information of different channels and mediums related to one's and to communicate securely, and to do so cryptography method is the sole key to it. In one of the algorithms, the use of a random seed is integrated to the Affine cipher. The random seed is used to generate unique keystream values that dynamically change the cipher's additive key for every character encrypted. The modification enables the cipher algorithm to produce ciphertext with no trace of repetitive character despite the advent of repetition of characters in the plaintext. The demand for veracity, privacy, fortification, solitude and procedures required for handling is increased. A new cryptographic Algorithm method for securing the data for Enhancing Data Security that can be used to secure applications on cloud computing has also been implemented. Bio-Inspired Cryptosystems are a modern form of Cryptography where bio- inspired and machine learning techniques are used for the purpose of securing data. An implementation of molecular biology based genetic coding has also been included. An approach for message communication uses 8 bit code values of the alphabet and performs some simple calculations like logical NOT and simple binary division to produce. Thus, this project is a combination of various efficient algorithms.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	
		Page Number
	Abstract	4
1	**CHAPTER-1:**	
	PROJECT DESCRIPTION AND OUTLINE	
	1.1 Introduction	8
	1.2 Motivation for the work	8
	1.3 [About Introduction to the project including techniques]	8
	1.4 Problem Statement	8
	1.5 Objective of the work	8
	1.6 Organization of the project	9
	1.6 Summary	9
2	**CHAPTER-2:**	
	RELATED WORK INVESTIGATION	
	2.1 Introduction	10
	2.2 <Core area of the project>	10
	2.3 Existing Approaches/Methods	
	2.3.1 Approaches/Methods -1	10
	2.3.2 Approaches/Methods -2	10
	2.3.3 Approaches/Methods -3	11
	2.4 <Pros and cons of the stated Approaches/Methods >	11
	2.5 Summary	12

3	**CHAPTER-3:** **REQUIREMENT ARTIFACTS** 3.1 Introduction 3.2 Hardware and Software requirements 3.3 Specific Project requirements 3.3.1 Data requirement 3.3.2 Functions requirement 3.3.3 Look and Feel Requirements 3.4 Summary	 13 13 13 13 13 14
4	**CHAPTER-4:** **DESIGN METHODOLOGY AND ITS NOVELTY** 4.1 Methodology and goal 4.2 Functional modules design and analysis 4.3 User Interface designs	 15 16 17
5	**CHAPTER-5:** **TECHNICAL IMPLEMENTATION & ANALYSIS** 5.1 Technical coding and code solutions 5.2 Prototype submission 5.3 Test and validation 5.4 Performance Analysis(Graphs/Charts)	 18 24 24 29
6	**CHAPTER-6:** **PROJECT OUTCOME AND APPLICABILITY** 6.1 Outline 6.2 Key implementations outlines of the System 6.3 Significant project outcomes	 30 30 30

	6.4 Project applicability on Real-world applications	31
7	**CHAPTER-7:** **CONCLUSIONS AND RECOMMENDATION** 7.1 Outline 7.2 Limitation/Constraints of the System 7.3 Future Enhancements 7.4 Inference	 32 32 33 33
8	**REFERENCES**	34

CHAPTER - 1:

PROJECT DESCRIPTION AND OUTLINE

1.1 **Introduction**

The project is in the form of a menu- driven tool which provides the users an option of using various encryption methods to encrypt their precious data. This project is based on the approaches proposed in several research works with various unique methods to improve upon the space, speed and security of existing ciphers.

1.2 **Motivation for the work**

The motivation behind this project lies in the fact that there are several theoretical approaches suggested by researchers for encryption and decryption of data. These provide secure encryption and information hiding with optimum security and storage. We were asked to read and summarize 40 research papers by our guide which led us to read 43 papers as a whole. Then we were asked to propose something which we can implement from these papers. We needed something which showed our understanding of these papers, thereby creating something new and useful.

1.3 **About Introduction to the project including techniques**

The technique was simply to create a menu- driven program where we have the option to choose between different algorithms implemented.

1.4 **Problem Statement**

At present, there is also a requirement to design encryption algorithms capable of providing optimum security and storage, besides symmetrical and asymmetric ciphers that can provide secure encryption and information hiding.

1.5 **Objective of the work**

The objective of the encryption is to secure or protect data from unauthorized access in terms of viewing or modifying the data. It is to show competent encryption time even on large data sizes when compared with existing cryptosystems. The aim is also to provide an implementation of a new cryptographic algorithm for securing the data to enhance data security that can be used to secure applications on cloud computing. Modification is needed to enable the existing affine cipher algorithm to produce ciphertext with no trace of

repetitive character despite the advent of repetition of characters in the plaintext.

1.6 **Organization of the project**

The tool has mainly 6 algorithms each having a special patch for a vulnerability found in the related existing approach.The included algorithms are as follows :

1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system

1.7 **Summary**

An encryption tool has been created to execute various approaches for optimizing the process of encryption. The advantages of the implemented approaches along with their constraints and limitations have been discussed in detail in this report.

CHAPTER-2:

RELATED WORK INVESTIGATION

2.1 Introduction

A lot of research and investigation was carried out in finalizing the algorithms to be implemented in this project. A summary of all the papers studied has been prepared and is attached below.

https://docs.google.com/document/d/15VIQsOTutZJX8F-hYdaCNfhPzMIc7GOTpjzDjMgTeos/edit?usp=sharing

2.2 Core area of the project

The core focus of the project is elimination of weaknesses of pre-existing ciphers.

2.3 Existing Approaches/Methods

2.3.1 Approaches/Methods -1

The past decades have seen an enormous progress in the development of cryptographic algorithms. The initial developments were in the form of monoalphabetic ciphers, polyalphabetic substitution ciphers, transposition ciphers and block ciphers. These were followed by more advanced encryption algorithms such as AES, DES, RSA and SHA. Currently, developments in the arena of cryptography have been directed towards the principle of confidentiality, integrity and availability. Symmetric and asymmetric ciphers have already been designed to provide secure encryption and information hiding.

2.3.2 Approaches/Methods -2

Existing approaches for affine cipher include the use of modular arithmetic formulas for both encryption and decryption. The encryption formula is given by $A_{j,d} : x \rightarrow y = A_{j,d}(x) = (jx + d) \bmod m$, where m is the range of alphabets, j and d are keys and the values of j and m are co-prime to each

other. The decryption is done using the equation Aj,d (y) ≡ j−1 (y − d) mod m, where j−1 is the inverse modular multiplicative of modulo m. It works by mapping a set of alphabets to a range of integers. Using modular arithmetic, each plaintext character is transformed into an integer and then it is transformed into a cipher-text character. For instance, first, each character is converted to its numerical equivalent according to its alphabetical index, such that A is 0 and Z is 25. Then the affine encryption formula is used to translate plain text into the cipher text.

2.3.3 **Approaches/Methods -3**

The Caesar cipher is an example of a substitution cipher. Each letter of a given plaintext, the information to be encrypted, is substituted with another letter some given number of positions from it in the alphabet. For example, if we had an alphabet comprised of the standard 26 letters in the English alphabet and swapped each letter with the letter three places after it in the alphabet, we would have the following Caesar cipher .

Plaintext:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Cipher text:

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Using this cipher, the text “GOOD MORNING” would encode to “JRRG PRUQLQJ.”

2.4 **Pros and cons of the stated Approaches/Methods**

1. Like any other substitution ciphers, the Affine cipher produces obvious

ciphertext patterns for plaintexts containing identical characters. As soon as a certain repeating character is decrypted, the remaining identical characters can easily be substituted even without computation or cryptanalysis.

2. One of the main problems with the Caesar cipher is that if an individual intercepts the Ciphertext and guesses that the Caesar cipher was used for the encryption, he or she could easily go through the 25 shift values until they come upon a shift that decodes the ciphertext into a meaningful plaintext. On a broader scale, simple substitution ciphers all fall victim to simple analysis. Given any particular alphabetic language, certain letters are used more or less frequently than other letters. In a simple substitution cipher, these frequencies with which each letter occurs in an average sentence or paragraph are maintained. For example, if a substitution cipher encoded “e” to “w,” “w” would occur in the ciphertext with the same frequency as “e” in the original language, allowing for a relatively simple analysis to break the substitution cipher.
3. Symmetric Key Encryption takes less execution time and is always easy to implement in real-life applications, yet their drawback is that both clients need to move their key’s security. This approach highlights improving the standard method of encryption by using the Substitution technique.
4. Each and every algorithm has some drawback be it attacks or weaknesses of the algorithm.

2.5 **Summary**

All the existing approaches have been explained above including their pros and cons.

CHAPTER-3:

REQUIREMENT ARTIFACTS

3.1 Introduction

Our project consists of an extremely lightweight tool which does not require any significant computing power or storage requirements.

3.2 Hardware and Software requirements

Hardware: No external hardware is required. Just a basic computer system is required to implement the tool.
Software: The software requirements are listed down as follows.
About 10KB storage: Since the tool is lightweight, the entire tool can be stored in a space of 10KB.
8GB RAM: The tool can work with lesser amount of RAM, but that would affect the execution time of the program. By using 8GB RAM, it is ensured that all the codes can complete their execution within a run-time of 2 seconds.
Python interpreter: Since, the programs have been coded in python, a Python interpreter is needed to compile the programs and execute them.
Linux Terminal: The tool is in the form of a Linux Command which can execute the program on a given plaintext and encrypt it.

3.3 Specific Project requirements

3.3.1 Data requirement

The encryption was tested using sample plaintext. Data utilized included information about the execution times of various functions and ciphers already available. The data was mostly taken from the research papers.

3.3.2 Functions requirement

Some extra functions have been imported in python as follows:

1. **OS -** To make the python program execute some other python programs.
2. **Array -** To store the plaintext where each letter was to be operated on individually
3. **Time -** To calculate the runtime of the program.

3.3.3 Look and Feel Requirements

Basic designing using python has been done to improve the look and feel of the tool. The main focus has been on optimizing the speed, space and security

of the encryption methods.

3.4 Summary

The project requirements have been described above in detail. To sum up, it can be said that the tool doesn't require much special hardware or software and is mainly based on python and linux.

CHAPTER-4:

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and goal

1. **i. Conversion of Genetic Code** : This is the process of converting the binary form to DNA in Genetic Code.
   ii. Transcription : This is the process of converting the DNA to mRNA form.
   iii. Translation : This is the process of converting the mRNA form to cipher text containing protein bases.
   The encryption algorithm involves converting the readable text of the sender into the ciphertext which is sent to the receiver. The string is then padded with k zeroes to make it a multiple of 16. This is to facilitate XOR with the 16-bit key in each round. The binary form obtained is XORed with the key and converted to the genetic code of DNA. Translation is then carried out to convert the DNA into mRNA. The mRNA is padded with 'N' or 'NN' in order to make it in 3-mRNA form. This is followed by translation in order to convert the mRNA to protein form, which yields the cipher text. All of the above processes were simulated using Python for the implementation of the cryptosystem.

2. The modified Affine cipher uses the equation $E(x) = (jx + d) \bmod 36$, where j must be a coprime of mod 36, x is the character index, and d is a value from the unique keystream. Decryption is done using the equation $D(y) = j^{-1} (y - d) \bmod 36$, where j^{-1} is the modular multiplicative inverse of mod 36, x is the character index, and d is a value from the unique keystream.

3. In Hill cipher each character is assigned a numerical value like:
   a=1, b=2, ...z=26.
   The substitution of cipher text letters in place of plaintext leads to m linear equations. For m=3, the system can be described as follows:
   C1=(K11P1+K12P2+K13P3)MOD 26---------1
   C1=(K21P1+K22P2+K23P3)MOD 26---------2
   C1=(K31P1+K32P2+K33P3)MOD 26---------3
   This can be expressed in terms of column vectors and matrices: C=KP where C and P are column vectors of length 3, representing the plaintext and the ciphertext and K is a 3*3 matrix, which is the encryption key. All

operations are performed mod 26 here. Decryption requires the inverse of matrix K. The inverse K-1 of a matrix K is defined by the equation. K K-1= I where I is the Identity matrix. The inverse of a matrix doesn't always exist, but when it does it satisfies the preceding equation. K-1 is applied to the cipher text, and then the plain text is recovered. In general terms we can write as follows:

For encryption: C=EK(P)=KP

4. **Step 1:** Generate the ASCII value of the letter

   Step 2: Generate the corresponding binary value of it. [Binary value hould be 8 digits e.g. for decimal 32 binary number should be 00100000]

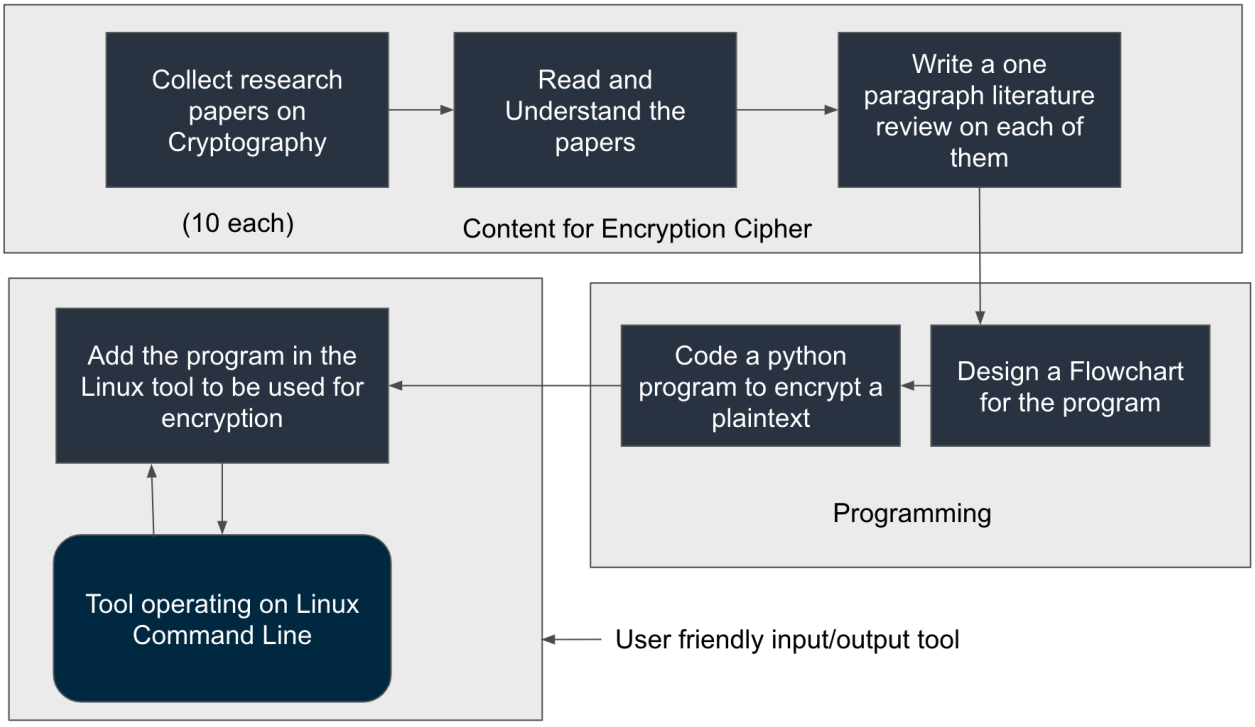
   Step 3: Reverse the 8 digit's binary number

   Step 4: Take a 4 digits divisor (>=1000) as the Key

   Step 5: Divide the reversed number with the divisor

   Step 6: Store the remainder in first 3 digits & quotient in next 5 digits (remainder and quotient wouldn't be more than 3 digits and 5 digits long respectively. If any of these are less than 3 and 5 digits respectively we need to add the required number of 0s (zeros) in the left hand side. So, this would be the cipher text.

4.2 Functional modules design and analysis



4.3 User Interface designs



CHAPTER-5:

TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Technical coding and code solutions

i. A Bio Inspired Cryptosystem

```
#plaintext ⟶ Binary ⟶ DNA ⟶ mRNA

def DecimalToBinary(n):
    return bin(n).replace("0b", "0")

ptext = input('Enter a plaintext to be encoded : ')
blist = []
bstring =''

for i in ptext:
    blist.append(DecimalToBinary(ord(i)))
    bstring += DecimalToBinary(ord(i))

for i in bstring:
    i = int(i)^int(1)
# 00 ⟶ A
# 01 ⟶ C
# 10 ⟶ G
# 11 ⟶ T

DNAstr = ''


for i in range(len(bstring)-1):
    a = i+1
    if bstring[i] == '0' and bstring[a] == '0':
        DNAstr += 'U'
    elif bstring[i] == '0' and bstring[a] == '1':
        DNAstr += 'G'
    elif bstring[i] == '1' and bstring[a] == '0':
        DNAstr += 'C'
    elif bstring[i] == '1' and bstring[a] == '1':
        DNAstr += 'A'

mRnastr = DNAstr
print('Ciphertext : \n')
print(mRnastr)
```

ii. A Keystream-Based Affine Cipher for Dynamic Encryption

```
from string import *
from array import *

plaintext=input("Enter the plaintext : ")
lptext=plaintext.lower()
b=int(input("Enter the random seed value : "))
print("Ciphertext : \n")
for i in range(len(plaintext)):
    x=ord(lptext[i])-97
    c=i+1
    a=x+b+c
    d=(a*(x^2))+(b*x)+c
    num=((7*x)+d)%26
    print(chr(num+97), end='')
print('\n')


```

iii. A Hill's Cipher Algorithm

```
import array as arr
from string import *

plaintext =input("Enter the plain text : ")
lptext=plaintext.lower()
key=[]

for j in range (4):
    print("Enter the key no. ",j+1," :",end=' ')
    key.append(int(input()))
print("Key : ",key)
print("Cipher text :", end=' ')
for i in range(len(plaintext)):
    k=chr(((ord(lptext[i])+key[i%4]-96)%26)+96)
    print(k,end=' ')


```

iv. A Secure and Fast Approach for Encryption and Decryption of Message Communication

```python
import array as arr

def DecimalToBinary(n):
    return bin(n).replace("0b", "0")

plaintext = input('Enter a plaintext to be encoded : ')
ASCII_list = []
binary = ''

for i in plaintext:
    #print(f'ASCII of : {i} is : {ord(i)}' )
    ASCII_list.append(ord(i))

for i in ASCII_list:
    bi = DecimalToBinary(i)
    binary += bi


#print(binary)
a = len(binary)

binary_new = ''

for i in range(a):
    if i%2 == 0:
        if binary[i] == '1':
            binary_new += '0'
        else:
            binary_new += '1'
    else:
        binary_new += binary[i]

#print(binary_new)

num = arr.array('i', [])

for i in binary_new:
    num.append(int(i))

#print(num)

for i in range(len(num)-1):
    temp = num[i]
    num[i] = num[i+1]
    num[i+1] = temp

#print(num)
```

```
48
49 array1 = arr.array('i', [])
50 for i in num:
51     array1.append(int(i))
52
53 array2 = arr.array('i',[])
54 for i in range(len(num)):
55     array2.append(0)
56
57 #print(array2)
58
59 for i in range(len(array1)):
60     if i%8 == 0 or i%8 == 1:
61         array2[i] = array1[i]
62     if i%8 == 2 or i%8 == 3 or i%8 == 4 or i%8 == 5:
63         array2[i+2] = array1[i]
64     if i%8 == 6 or i%8 == 7:
65         array2[i-4] = array1[i]
66
67 print('The ciphertext is given as : ')
68 for i in array2:
69     print(i,end="")
70
```

v. A Symmetric key algorithm to boost security of applications on cloud computing

```
plaintext = input(str('Enter the plaintext you want to encrypt : '))


def DecimalToBinary(n):
    return bin(n).replace("0b", "")


for i in plaintext:
    pDec = ord(i) + 100
    pBin = DecimalToBinary(pDec)
    pRev = pBin[::-1]
    # print('Reverse : ',pRev)
    pPart1 = pRev[0:4]
    pPart2 = pRev[4:8]
    # print('Part 1 : ',pPart1)
    # print('Part 2 : ',pPart2)

    p11 = ''
    for i in pPart1:
        if i == '0':
            p11 += '1'
        else:
            p11 += '0'
    # print('1s : ',p11)

    p12 = bin(int(p11, 2) + int('1', 2))
    ans1 = p12[2:]

    while len(ans1) < 4:
        ans1 = '0' + ans1
    # print('2s : ', ans1)

    p21 = ''
    for i in pPart2:
        if i == '0':
            p21 += '1'
        else:
            p21 += '0'
    # print('1s : ',p21)

    p22 = bin(int(p21, 2) + int('1', 2))
    ans2 = str(p22[2:])

    while len(ans2) < 4:
        ans1 = '0' + ans2
    # print('2s : ', ans2)
```

```
47
48      ans = ans2 + ans1
49
50      # print(ans)
51
52      s1 = ''
53      for i in ans:
54          if i == '0':
55              s1 += '1'
56          else:
57              s1 += '0'
58      # print('1s : ',s1)
59
60      s2 = bin(int(s1, 2) + int('1', 2))
61      ANS = s2[2:]
62
63      while len(ANS) < 4:
64          ANS = '0' + ANS
65      # print('2s : ', ANS)
66
67      # print('The ciphertext is : ')
68      a = str(hex(int(ANS)))
69      b = a.replace('0x', '')
70
71      while len(b) < 8:
72          b = '0' + b
73
74      print(b, end=" ")
```

vi. Encryption using modified Solvability Equation

```
plaintext = input(str('Enter the plaintext you want to encrypt : '))
alphabet = {
    'a': 1,
    'b': 2,
    'c': 3,
    'd': 4,
    'e': 5,
    'f': 6,
    'g': 7,
    'h': 8,
    'i': 9,
    'j': 10,
    'k': 11,
    'l': 12,
    'm': 13,
    'n': 14,
    'o': 15,
    'p': 16,
    'q': 17,
    'r': 18,
    's': 19,
    't': 20,
    'u': 21,
    'v': 22,
    'w': 23,
    'x': 24,
    'y': 25,
    'z': 26
}
pstring = ''
for i in plaintext:
    pstring += str(alphabet[i]) + ' '
def GCD(x, y):
    if x > y:
        small = y
    else:
        small = x
    for i in range(1, small + 1):
        if ((x % i == 0) and (y % i == 0)):
            gcd = i
    return gcd
# Key equation : 77x = 33 (mod 55)
a = 77
n = 55
b = 33
```

```
g = GCD(a, n)
a1 = a / g
b1 = b / g
n1 = n / g
y = n1
k = int((y * n1 + b1) / a1)
kstring = []
for m in range(1, 27):
    b = k + (m - 1) * n1
    kstring.append(int(b))
i = 0
for k, v in alphabet.items():
    alphabet[k] = kstring[i - 1]
    i += 1
cstring = ''
for i in plaintext:
    cstring += str(alphabet[i]) + ' '
print('The encrypted ciphertext is : ',cstring)

```

5.2 Prototype submission

The link to the prototype code has been uploaded to the google drive and is attached below for reference.

https://drive.google.com/drive/folders/14n-cJOboqrMMhEXb727mM6Q79_8lhjOD?usp=sharing

5.3 Test and validation



```
Enter the plaintext you want to encrypt : helloworld
The encrypted ciphertext is :  34 19 54 54 69 109 69 84 54 14
Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
2
Enter the plaintext you want to encrypt : jac
0001870f 00018a92 00018af6 Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
```

```
cryphi@kali: ~

Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
3
Enter a plaintext to be encoded : helloabc
The ciphertext is given as :
1001000110110111100100111001001110110010101101100101001011010 0



Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
4
Enter the plain text : good morning
Enter the key no.  1  : 1234
Enter the key no.  2  : 12
Enter the key no.  3  : 1
Enter the key no.  4  : 43
Key :  [1234, 12, 1, 43]
Cipher text : s a p u ` y p i ` u o x



Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
```

```
Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
4
Enter the plain text : good night
Enter the key no.  1  : 3
Enter the key no.  2  : 5
Enter the key no.  3  : 2
Enter the key no.  4  : 5
Key :  [3, 5, 2, 5]
Cipher text : j t q i q s k l k y



Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
5
Enter the plaintext : new york outlandish
Enter the random seed value : 24
Ciphertext :

mubavgqeigvcgmcwbcs



Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
```

```
Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
6
Enter a plaintext to be encoded : helloworld
Ciphertext :

GACGCUUUGACUGCGCGACGACUUGACGACUUGACGAAACGAACGAACGACGAAACGAACUGCUGACGACUUGACUGCU



Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
6
Enter a plaintext to be encoded : drstrange
Ciphertext :

GACUGCUUGAACUGCUGAACUGACGAACGCUUGAACUGCUGACUUUGCGACGAACUGACUGAACGACUGCG



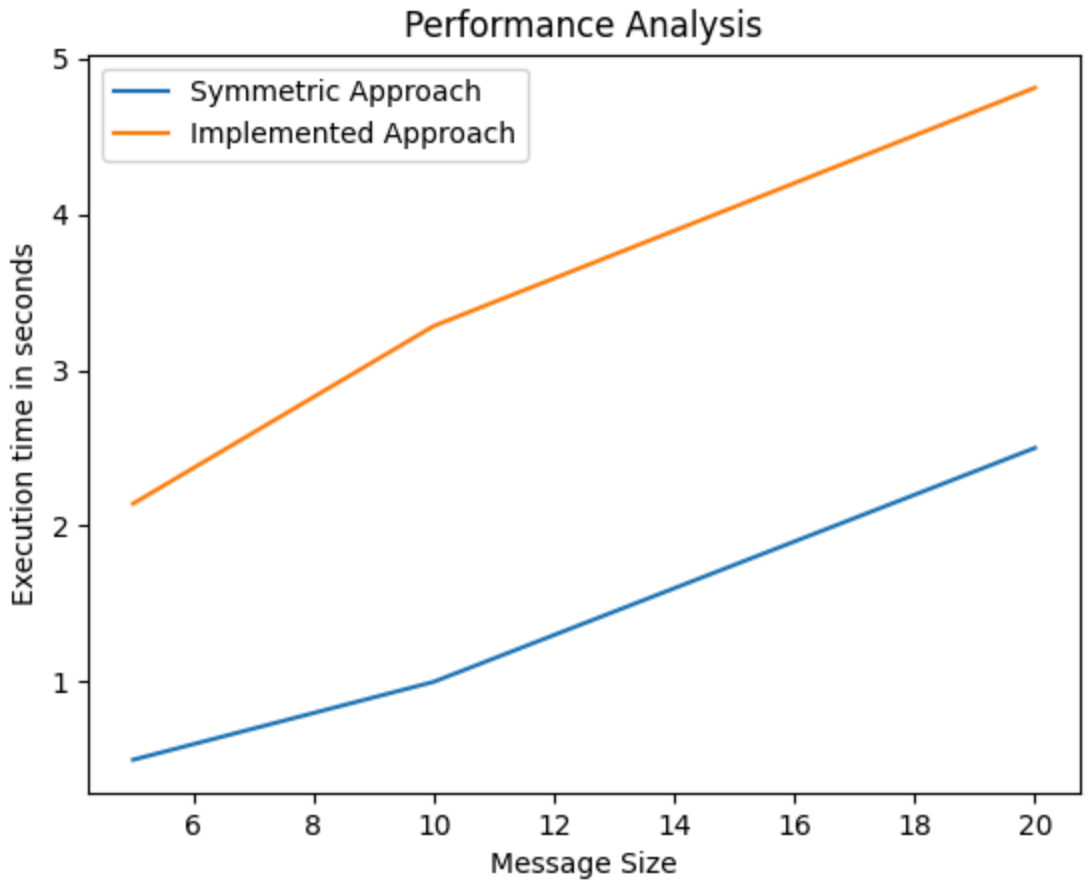
Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
6
Enter a plaintext to be encoded : spiderman
Ciphertext :

GAACUGACGAACUUUUGACGCUGCGACUGCUUGACUGCGCGAACUGCUGACGACGCGACUUUGCGACGAAC
```

```
Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
7


Enter your choice :
1. Encryption using Modified Solvability Equation
2. Enhanced symmetric key algorithm for enhanced Data Security
3. Secure and Fast approach for message encryption
4. Modified Hill Cipher for encryption
5. Key stream based Affine cipher for Dynamic Encryption
6. Bio Inspired DNA encryption system
7. Continue
8. Exit
8
                        Thank you for using crypt!
```

5.4 **Performance Analysis(Graphs/Charts)**



CHAPTER-6:

PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

The significant outcomes of the project include an implementation of an interesting bio- based cryptosystem which also achieves the properties of confusion, diffusion and avalanche effect. It can withstand the various attacks such as plaintext only and brute force attacks etc. The vulnerability of the usual affine cipher has also been patched by the modified affine cipher implemented as part of the project.

6.2 Key implementations outlines of the System

1. The Encryption Algorithm has been designed by simulating the natural processes of Genetic Coding (conversion from binary to DNA bases), Transcription (conversion from DNA to mRNA) and Translation (conversion from mRNA to Protein) to allow for encryption. All inputs are considered to be in the form of blocks of 16-bits. The final outputs from the blocks can be concatenated to form the final cipher text in the form of protein bases.
2. A random seed value is added to the Affine cipher to generate a unique keystream that enables the cipher to produce unique ciphertext characters despite having identical letters from the plaintext.
3. The Hill cipher is based on linear algebra and overcomes the frequency distribution problem of the Caesar cipher that was previously discussed. encryption, algorithm takes m successive plaintext letters and instead of that substitutes m cipher letters.

6.3 Significant project outcomes

1. The Bio-Inspired Cryptosystem facilitates the properties of confusion and diffusion. The binary conversion of ciphertext depends on more than one bit of the key. The processes of transcription and translation produce changes in the ciphertext when a single bit of plaintext is changed. It also achieves the avalanche effect which is the property where a slight change in the plaintext results in a significant change in the ciphertext. It also provides a critical and

complex mapping between the plaintext and ciphertext.

2. The cryptosystem is resilient to the brute-force attack, withstands ciphertext only attacks and known plaintext attacks and is secure to differential cryptanalysis attacks. It also achieves the confidentiality, integrity and availability principle.
3. The vulnerability issue of the standard Affine cipher that is rooted in the plaintext structure is addressed by increasing the complexity. The enhanced Affine cipher generates results with distinct substitution values enabling a more secure encryption process.

6.4 **Project applicability on Real-world applications**

1. The bio- inspired cryptosystem is capable of providing security as well as efficient storage to data, which solves the future storage problems of data in case of conventional cryptographic algorithms and it's encryption and decryption.
2. Real-world work in this direction includes incorporating additional security and authentication measures to strengthen this scheme. Artificial Intelligence also has enormous scope of application in the area of cryptography. DNA cryptography is still in a nascent state and more powerful cryptosystems can be built using this computing paradigm.
3. The enhancement of usual affine cipher's capability and complexity in masking plaintext has paved the way to a more secure and dynamic data encryption. With the large-scale use of computers in this modern world, message sharing becomes very easy via e- mail, mobile phone communication etc.,. Also the information that is being sent via computers can be intercepted and read by an enemy. So the necessity of data hiding is obvious.
4. The implemented method is used for message communication. Short messages can be sent securely using encryption techniques.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1 Outline

It can be concluded from the project that the ciphers always have some or the other vulnerabilities and there is a scope of improving upon them. New age technologies like DNA cryptography and machine learning can be leveraged to create more secure cryptosystems which also solve the future problems of storage and working on large data.

7.2 Limitation/Constraints of the System

1. The scheme implemented doesn't give sufficient performance compared to scheme proposed in [24], it performs relatively better than that in [25] with regard to encryption times.
2. The implemented cipher is vulnerable to a known-plaintext attack because it is completely linear. While matrix multiplication alone does not result in a secure cipher it is still a useful step when combined with other non-linear operations, because matrix multiplication can provide diffusion.
3. Symmetric Key Encryption takes less execution time and is always easy to implement in real-life applications, yet their drawback is that both clients need to move their key's security. This approach highlights improving the standard method of encryption by using the Substitution technique.

7.3 Future Enhancements

1. Incorporating additional security and authentication measures to strengthen the bio-inspired cryptosystem. Also, with advancement in AI and DNA cryptography, more powerful cryptosystems can be built.
2. The proposed approach is based on the number of characters in the message and simple calculation and operations are performed to minimize the execution time. In future this work for special characters will be implemented.

7.4 **Inference**

It can be inferred that the artificial intelligence and machine learning techniques, if combined with DNA cryptography and molecular biology can help in simulating natural processes for genetic coding.

REFERENCES

1. An overview of cryptanalysis research for the advanced encryption standard | IEEE Conference Publication | IEEE Xplore

2. https://www.ijitee.org/wp-content/uploads/papers/v8i8s3/H10870688S319.pdf

3. [2005.00157] A Novel Structure of Advance Encryption Standard with 3-Dimensional Dynamic S-box and Key Generation Matrix (arxiv.org)

4. Automatically eliminating speculative leaks from cryptographic code with blade | Proceedings of the ACM on Programming Languages

5.https://drive.google.com/drive/folders/1QbBGiP3hBYyxueXFbU6arBEkl8rkvw_2?usp=sharing