PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER
UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT
ON

Cryptographic Applications of Linear Algebra - The Hill Cipher and analysis of the non - invertible key matrix problem

Submitted by

1.	Sujan CR	PES1201801286
2.	Adithya Bennur	PES1201801891

Branch & Section : CSE - F

PROJECT EVALUATION

(For Official Use Only)

Sl.No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor : Ramesh Bhat H

Signature of the Course Instructor :

Cryptographic Applications of Linear Algebra - The Hill Cipher and analysis of the non - invertible key matrix problem

Table of Contents:

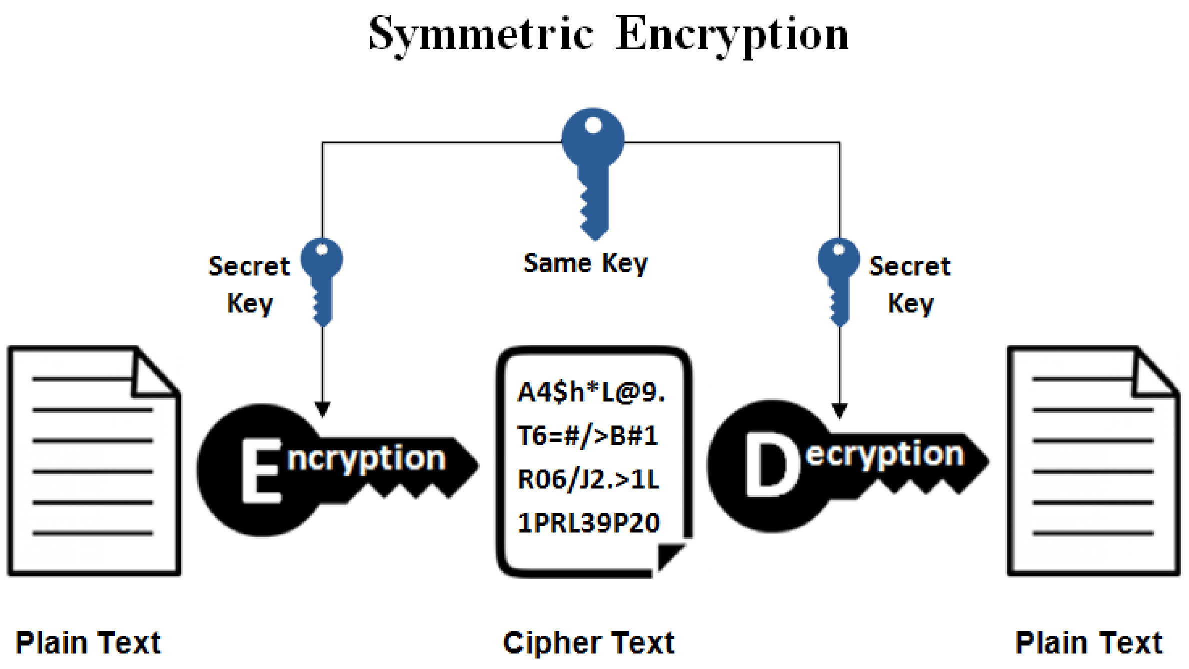
INTRODUCTION

Information is organized or classified data, which has some meaningful values for the receiver. Information is the processed data on which decisions and actions are based. Information security, sometimes abbreviated to *infosec,* is a set of practices intended to keep data secure from unauthorized access or alterations, both when it's being stored and when it's being transmitted from one machine or physical location to another. Cryptography, or cryptology, is the practice and study of techniques for secure communication in the presence of third parties called adversaries.
The Roman ruler Julius Caesar (100 B.C. – 44 B.C.) used a very simple cipher for secret communication. He substituted each letter of the alphabet with a letter three positions further along. Later, any cipher that used this "displacement" concept for the creation of a cipher alphabet, was referred to as a Caesar cipher. Of all the substitution type ciphers, this Caesar cipher is the simplest to solve, since there are only 25 possible combinations.

If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others.
— Suetonius, Life of Julius Caesar 56 .

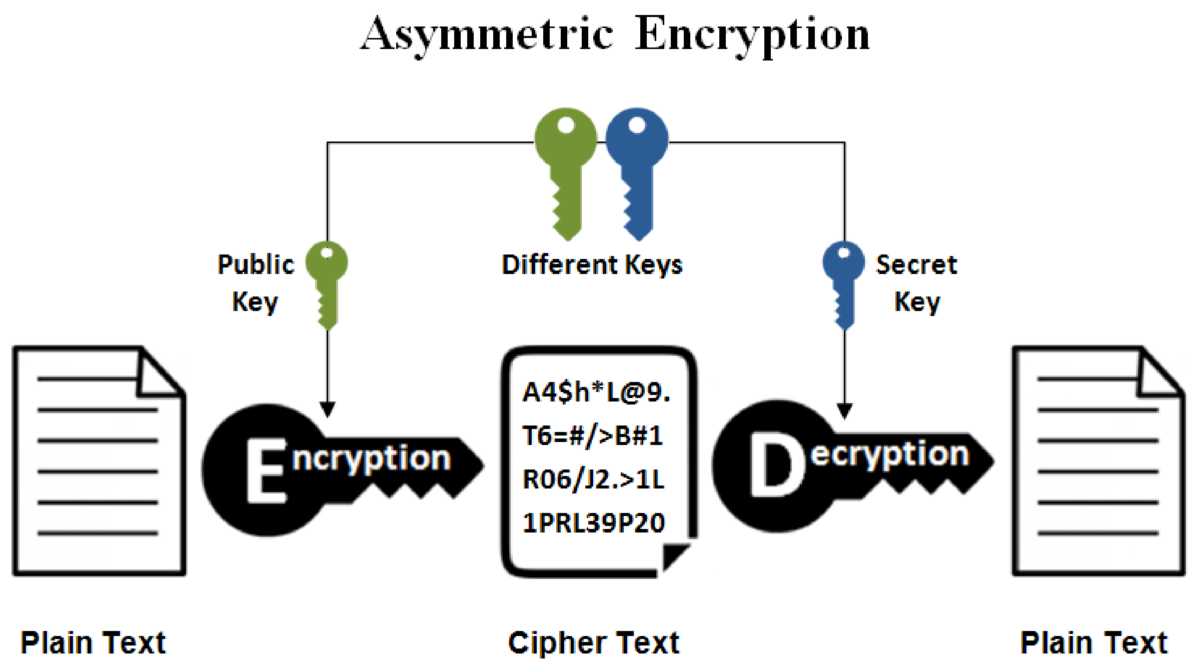
Symmetrical Encryption

This is the simplest kind of encryption that involves only one secret key to cipher and decipher information. Symmetrical encryption is an old and best-known technique. The sender and the recipient should know the secret key that is used to encrypt and decrypt all the messages. Blowfish, AES, RC4, DES, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithm is AES-128, AES-192, and AES-256.
The main disadvantage of the symmetric key encryption is that all parties involved have to exchange the key used to encrypt the data before they can decrypt it.



Asymmetrical Encryption

Asymmetrical encryption is also known as public key cryptography, which is a relatively new method, compared to symmetric encryption. Asymmetric encryption uses two keys to encrypt a plain text. It is important to note that anyone with a secret key can decrypt the message and this is why asymmetrical encryption uses two related keys to boosting security. A public key is made freely available to anyone who might want to send you a message. The second private key is kept a secret so that you can only know. A message that is encrypted using a public key can only be decrypted using a private key, while also, a message encrypted using a private key can be decrypted using a public key. Asymmetric encryption is mostly used in day-to-day communication channels, especially over the Internet. Popular asymmetric key encryption algorithm includes EIGamal, RSA, DSA, Elliiptic Curve techniques, PKCS.



Hill Cipher

Hill cipher is a polygraphic substitution cipher based on linear algebra.Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. The Hill cipher is an example of a block cipher. A block cipher is a cipher in which groups of letters are enciphered together in equal length blocks. The Hill cipher was developed by Lester Hill and introduced in an article published in 1929. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

Sample Encryption and Decryptions:

In order to encrypt a message using the Hill cipher, the sender and receiver must first agree upon a key matrix *A* of size *n* x *n*. *A* must be **invertible** mod 26 which means, the determinant of A should be relatively prime to 26(The alphabet size). The plaintext will then be enciphered in blocks of size *n*. In the following example *A* is a 2 x 2 matrix and the message will be enciphered in blocks of 2 characters.For Example

Let

$$K = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$$

be the key and suppose the plaintext message is HELP. Then this plaintext is represented by two pairs

$$HELP \rightarrow \begin{pmatrix} H \\ E \end{pmatrix}, \begin{pmatrix} L \\ P \end{pmatrix} \rightarrow \begin{pmatrix} 7 \\ 4 \end{pmatrix}, \begin{pmatrix} 11 \\ 15 \end{pmatrix}$$

As we have a 2X2 key matrix we have to take the plain text of the form 2X1 i.e We take 2 characters of plain text at a time:

$$\begin{matrix} 3 & 3 \\ 2 & 5 \end{matrix} \quad \begin{matrix} 7 \\ 4 \end{matrix} = \begin{matrix} 33 \\ 34 \end{matrix} \mathit{mod}\, 26 = \begin{matrix} 7 \\ 8 \end{matrix}$$

$$\begin{matrix} 3 & 3 \\ 2 & 5 \end{matrix} \quad \begin{matrix} 11 \\ 15 \end{matrix} = \begin{matrix} 78 \\ 97 \end{matrix} \mathit{mod}\, 26 = \begin{matrix} 0 \\ 19 \end{matrix}$$

after encrypting we get “HIAT”.

The matrix K is invertible, hence K^{-1} exists such that $KK^{-1}=K^{-1}K=I_2$. The inverse of K can be computed by following these steps:

1) Find the multiplicative invers of the determinant which basically means if we have a matrix

{ a , b

c, d }

we have to solve for the equation (ad-bc) x ? = 1 modulo 26 , once we find the value of “?” . In this case solving we get ad-bc=9 therefore 9 x 3 = 1 modulo 26 , hence 3 is the multiplicative inverse of 9 over modulo 26.

2) We find inverse by $(ad-bc)^{-1}$ x adjoint(K) adjoint of K for 2x2 can be found by swapping the principal diagnol elements and flipping the sign of secondary diagnol elements

$$K^{-1} = \begin{matrix} 15 & 17 \\ 20 & 9 \end{matrix}$$

Finally for decryption We can use the formula D = K^{-1} x ciphertext (2 at a time in this case)

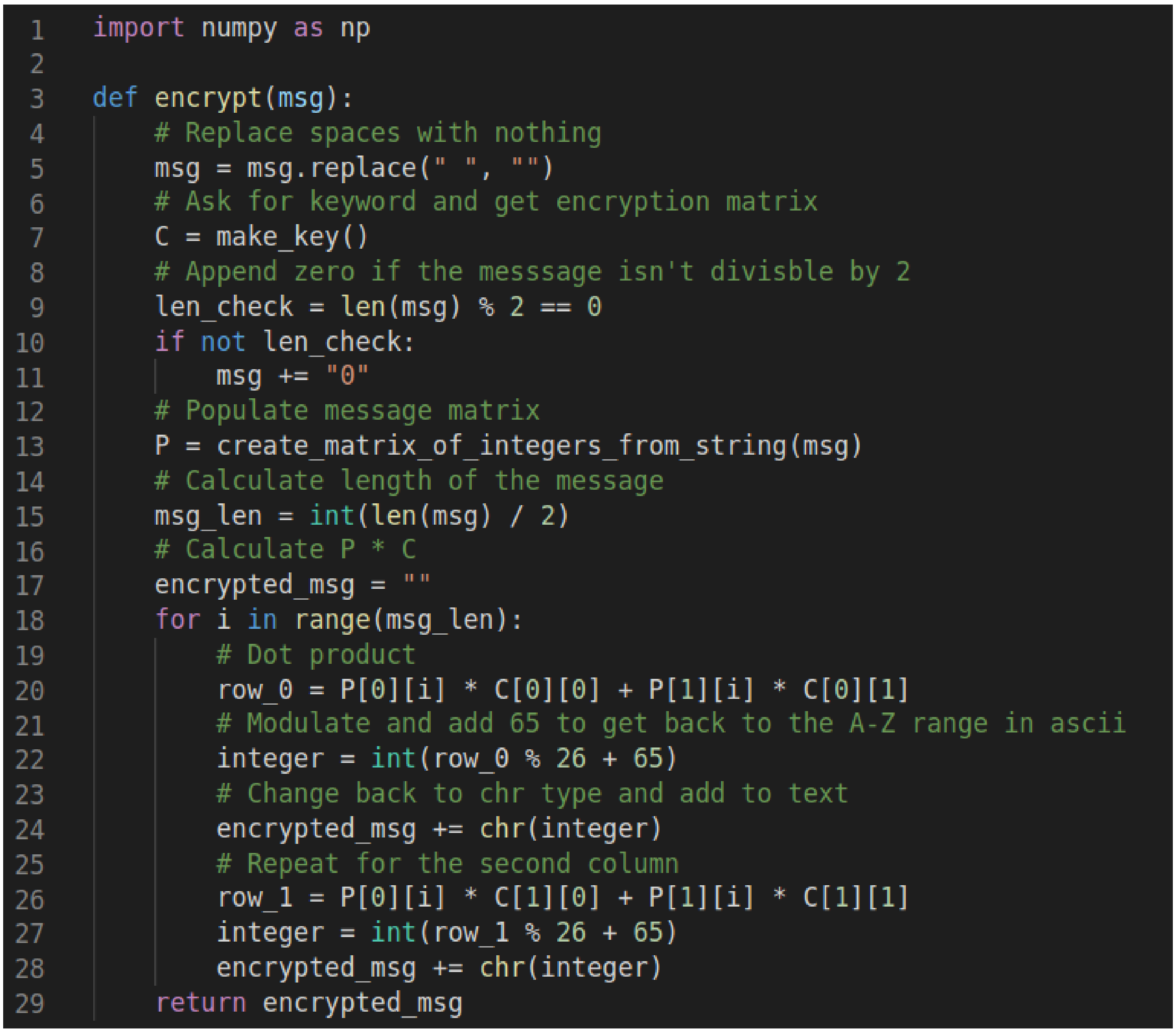
$$\begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \begin{pmatrix} 7 \\ 8 \end{pmatrix} \equiv \begin{pmatrix} 7 \\ 4 \end{pmatrix} \pmod{26},$$

$$\begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \begin{pmatrix} 0 \\ 19 \end{pmatrix} \equiv \begin{pmatrix} 11 \\ 15 \end{pmatrix} \pmod{26}$$

So now we have got back the original text “HELP”.

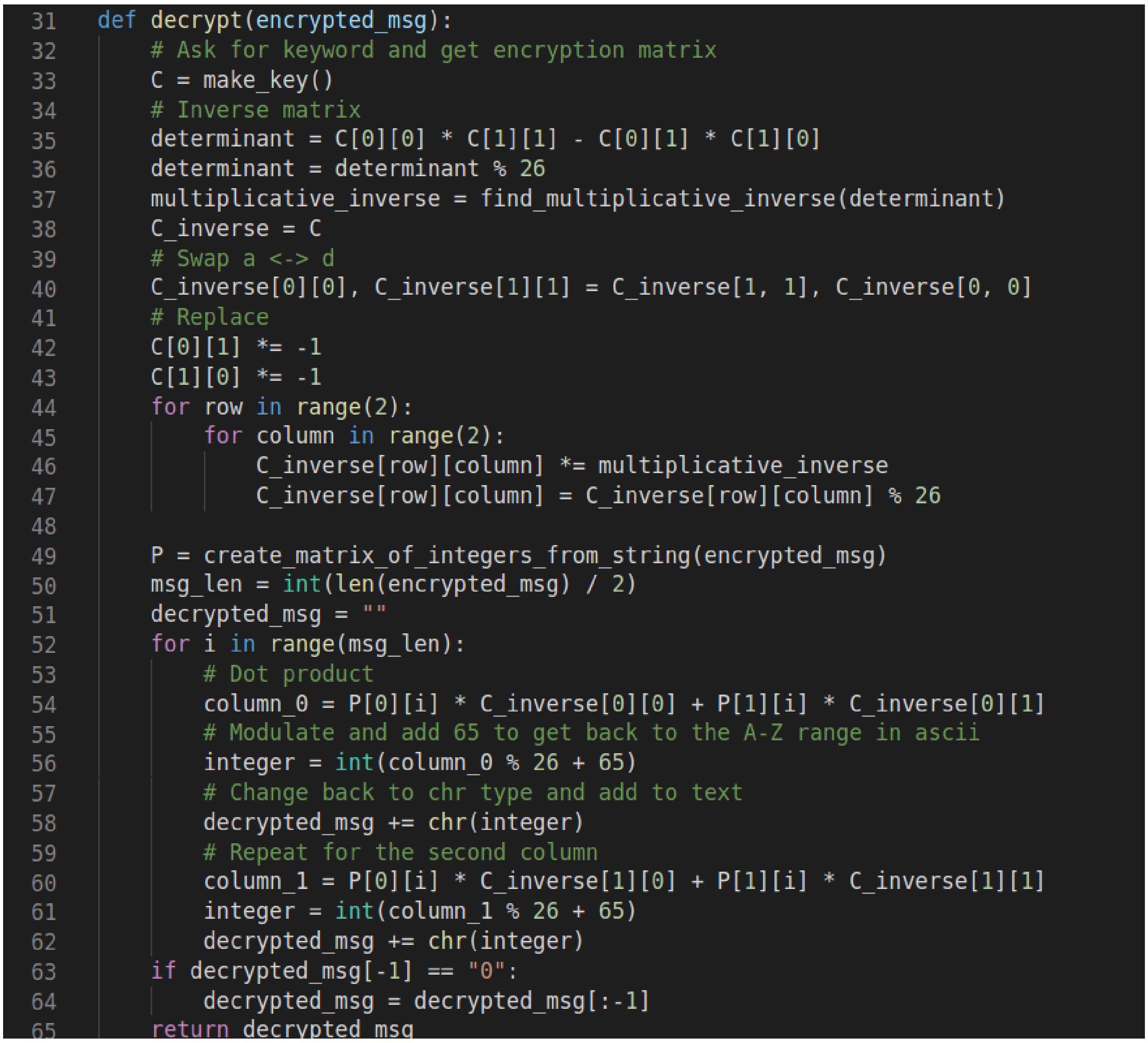
Python3 (3.8) implementaion:

(Pastebin link:https://pastebin.com/Ldq3TYTZ)



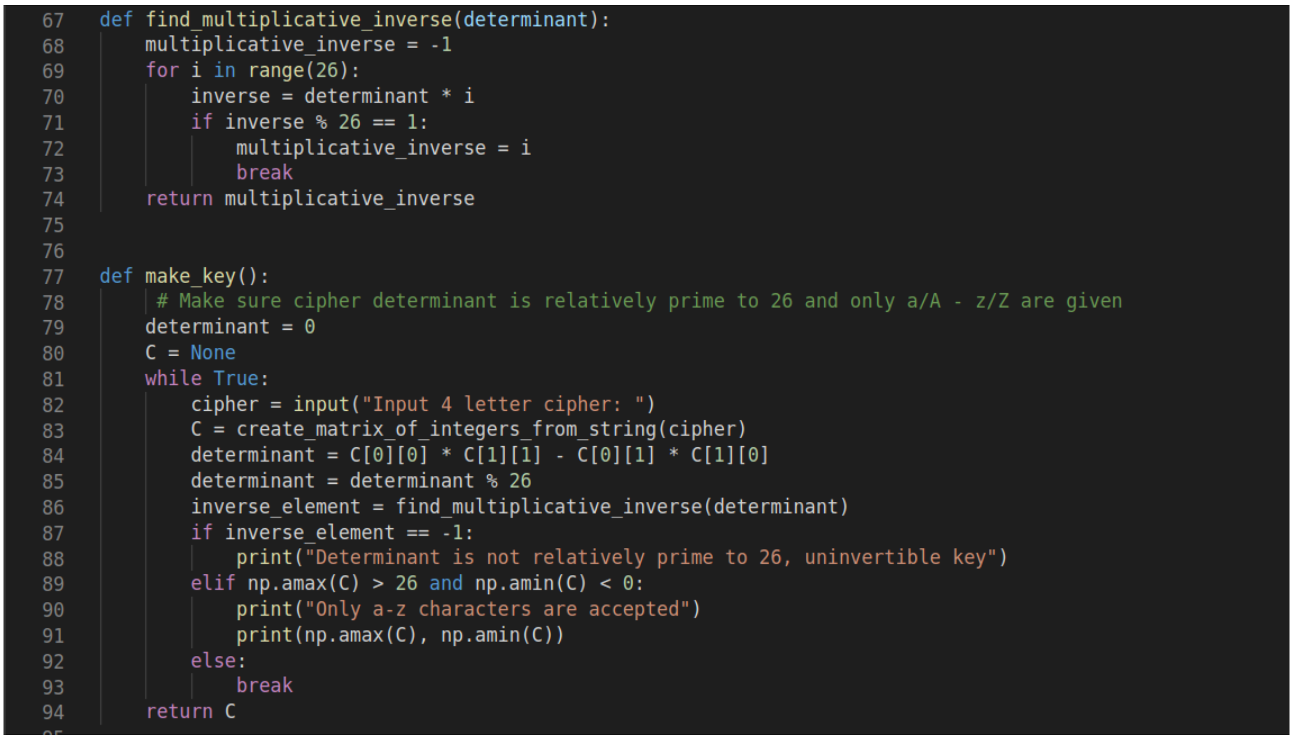
```python
import numpy as np

def encrypt(msg):
    # Replace spaces with nothing
    msg = msg.replace(" ", "")
    # Ask for keyword and get encryption matrix
    C = make_key()
    # Append zero if the messsage isn't divisble by 2
    len_check = len(msg) % 2 == 0
    if not len_check:
        msg += "0"
    # Populate message matrix
    P = create_matrix_of_integers_from_string(msg)
    # Calculate length of the message
    msg_len = int(len(msg) / 2)
    # Calculate P * C
    encrypted_msg = ""
    for i in range(msg_len):
        # Dot product
        row_0 = P[0][i] * C[0][0] + P[1][i] * C[0][1]
        # Modulate and add 65 to get back to the A-Z range in ascii
        integer = int(row_0 % 26 + 65)
        # Change back to chr type and add to text
        encrypted_msg += chr(integer)
        # Repeat for the second column
        row_1 = P[0][i] * C[1][0] + P[1][i] * C[1][1]
        integer = int(row_1 % 26 + 65)
        encrypted_msg += chr(integer)
    return encrypted_msg
```



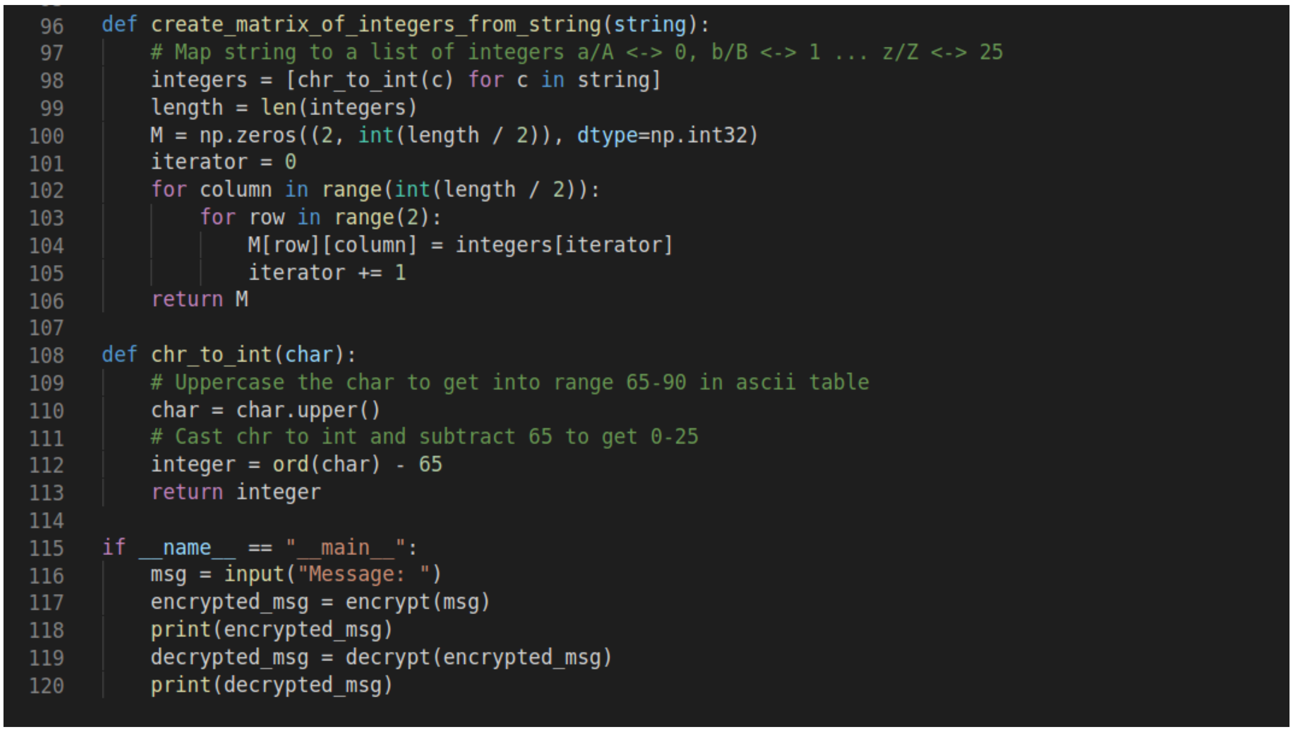
```python
def decrypt(encrypted_msg):
    # Ask for keyword and get encryption matrix
    C = make_key()
    # Inverse matrix
    determinant = C[0][0] * C[1][1] - C[0][1] * C[1][0]
    determinant = determinant % 26
    multiplicative_inverse = find_multiplicative_inverse(determinant)
    C_inverse = C
    # Swap a <-> d
    C_inverse[0][0], C_inverse[1][1] = C_inverse[1, 1], C_inverse[0, 0]
    # Replace
    C[0][1] *= -1
    C[1][0] *= -1
    for row in range(2):
        for column in range(2):
            C_inverse[row][column] *= multiplicative_inverse
            C_inverse[row][column] = C_inverse[row][column] % 26

    P = create_matrix_of_integers_from_string(encrypted_msg)
    msg_len = int(len(encrypted_msg) / 2)
    decrypted_msg = ""
    for i in range(msg_len):
        # Dot product
        column_0 = P[0][i] * C_inverse[0][0] + P[1][i] * C_inverse[0][1]
        # Modulate and add 65 to get back to the A-Z range in ascii
        integer = int(column_0 % 26 + 65)
        # Change back to chr type and add to text
        decrypted_msg += chr(integer)
        # Repeat for the second column
        column_1 = P[0][i] * C_inverse[1][0] + P[1][i] * C_inverse[1][1]
        integer = int(column_1 % 26 + 65)
        decrypted_msg += chr(integer)
    if decrypted_msg[-1] == "0":
        decrypted_msg = decrypted_msg[:-1]
    return decrypted_msg
```



```python
def find_multiplicative_inverse(determinant):
    multiplicative_inverse = -1
    for i in range(26):
        inverse = determinant * i
        if inverse % 26 == 1:
            multiplicative_inverse = i
            break
    return multiplicative_inverse


def make_key():
    # Make sure cipher determinant is relatively prime to 26 and only a/A - z/Z are given
    determinant = 0
    C = None
    while True:
        cipher = input("Input 4 letter cipher: ")
        C = create_matrix_of_integers_from_string(cipher)
        determinant = C[0][0] * C[1][1] - C[0][1] * C[1][0]
        determinant = determinant % 26
        inverse_element = find_multiplicative_inverse(determinant)
        if inverse_element == -1:
            print("Determinant is not relatively prime to 26, uninvertible key")
        elif np.amax(C) > 26 and np.amin(C) < 0:
            print("Only a-z characters are accepted")
            print(np.amax(C), np.amin(C))
        else:
            break
    return C
```



```python
def create_matrix_of_integers_from_string(string):
    # Map string to a list of integers a/A <-> 0, b/B <-> 1 ... z/Z <-> 25
    integers = [chr_to_int(c) for c in string]
    length = len(integers)
    M = np.zeros((2, int(length / 2)), dtype=np.int32)
    iterator = 0
    for column in range(int(length / 2)):
        for row in range(2):
            M[row][column] = integers[iterator]
            iterator += 1
    return M

def chr_to_int(char):
    # Uppercase the char to get into range 65-90 in ascii table
    char = char.upper()
    # Cast chr to int and subtract 65 to get 0-25
    integer = ord(char) - 65
    return integer

if __name__ == "__main__":
    msg = input("Message: ")
    encrypted_msg = encrypt(msg)
    print(encrypted_msg)
    decrypted_msg = decrypt(encrypted_msg)
    print(decrypted_msg)
```

This investigation is on lines of this specific fail case of the cipher.

Lets look at a scenario where in the key matrix is “Non invertible”.

Lets take 2 plain text ciphers being P_1 and P_2 , Let K be the Key Matrix :

$P_1 = \begin{matrix} 1 \\ 5 \end{matrix}$, $P_2 = \begin{matrix} 1 \\ 18 \end{matrix}$, $K = \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$

After encrypting we surprisingly get the same cipher text which is :

$C = \begin{matrix} 11 \\ 23 \end{matrix}$

This Happens since the determinant of the key matrix is -2 and when you take modulus of this with the alphabet size (in this case 26) we get 24 , Now 24 is not relatively prime to 26 as it has a common factor which is 2.

REVIEW OF LITERATURE

Rushdi A. Hamamreh, Mousa Farajallah[1], a team of a faculty and a student in 2009 pitched in a method to overcome the problem of non-invetible matirices as keys, and could be coined as the first successful paper to thereotically and pratically overcome the problem. They presented the idea and the mathematical proof of concept in their paper alongside with the other cryptographical conceptual constraints including the method for sending the key, method for generating the specialised key which is a development over the classical hill cipher, and the complete algorithm of the whole process. The mathematical concpet of handling the non-invertible key matrix is clearly cited in the paper as we quote them in the investigation section of the paper.

Another research paper released in 2010[ref:2] by Mr. Bibhudendra Acharya and team of students from Department of E & TC, NIT Raipur, Chhattisgarh provided non-algebral solutions for further protection of the method pitched by Rushdi A. Hamamreh, Mousa Farajallah[ref:1] using XOR encryption and Steganographical method of LSB insertion driven by random number seeded generation. This concept “singly” just enchances the complexity of the cipher but does not change its “Symmetrical Nature”.

In 2014, Neha Sharma, Sachin Chirgaiya from Oriental University, Indore published a paper[3] on the concept of the compensations (offsets). They published the detailed algorithm of the key transformation and modification process. The paper also showed illustrations on how the modification could help giving out the solution needed. The algorithm is presented in the investigation section of the paper as it has been cited by them.

A research paper released in December 2017[4] by Ms. Anjali Saxena from Sri Sathya Sai University (ISSN : 2319-8354) had already analysed a potential method for overcoming this particular problem using the concept of compensations and thus deriving transformed keys. However, the paper did not explain any solid method to arrive at the transformed matrix nor demonstrate an illustration to facilitate the readers as this was already clearly explained by Neha Sharma and Sachin Chirgaiya in their paper[3]. The paper also describes the process as one which has zero vulnerabilities which is a huge step in the development of the classical Hill cipher.

PRESENT INVESTIGATION

All the below given algorithms are cited in the respective papers as we quote them only.

Classical Hill Cipher

A. Encryption:

1- C= K x P modulo (alphabet size)

where C= cipher text , K=Key matrix (nxn) , P=Plain text (nX1)

B. Decryption:

1- K^{-1}= d^{-1} x adj(K)

where d is det(K) and d^{-1}=d modulo (aplhabet size)

The method of Extra Identity[1]

A. Encryption:

1- The plaintext characters are converted into numerical numbers.

2- Only if the resultant determinant of key matrix K is zero then identity matrix is added.

3- The column vector $C = K \times X$ is calculated.

4- Calculate C 1 = fix (C / P) , C 2 = mod (C , P) .

5- The numerical numbers (C 1 , C 2) is converted into characters.

B. Decryption:

1- The two sequence of cipher-text (C1, C2) is converted back into numerical numbers (Y 1 , Y 2).

2- Only if the resultant determinant of key matrix K is zero then identity matrix is added.

3- The column vector $P = inv (K) \times ((Y 1 \times 256) + Y 2)$ is calculated.

4- The numerical numbers P is converted into characters.

The method of Compensations(offsets)[3]

A. Encryption:

i) Firstly read the plain text P and chosen key matrix K

ii) Find the determinant of key matrix K that is |K|

iii) If |K| > = 0 set compensate =1 or |K| < 0 set compensate = -1

iv) After set compensate value modified the chosen key K according to above condition

v) set K to Km after modify the key matrix

vi) Find the cipher text $C = Km \times P \bmod 33$

B. Decryption:

i) Firstly read the Cipher text C and modified key matrix as Km

ii) Find the determinant of key matrix Km set X = |Km| mod 33

iii) If X < = 0 set X = X+ 33 or X > 0 set X = X

iv) After set values of X find the value of i for this i × X mod 33 = = 1 this function is helpful for find the value of i

v) After finding value of i ser it to Y = i then find the inversion of Km-1 with Km-1 = adj Km × Y mod 33

vi) Now next step shows transpose of Km-1 = (Km-1)’

vii) Finally For plain text P = C × Km-1 mod 33

viii) If Pij < = 0 set Pij = Pij + 33 where i = 0 to 2 and j= 0

RESULTS AND CONCLUSIONS

Important observations made:

1) Prefer alphabet size to be prime so that many possible det-values are relatively prime and thus inverse is possible (The risk of determinant having common factors with the modulus can be reduced by taking a prime number as the modulus).

2) If Dimensions of Key matrix is nxn. Then the dimensions of plaintext is chosen as nx1 so that matrix multiplication is possible enabling smooth encryption.

A matrix being invertible was an essential property of the matrix that was chosen to be the key ‘A’, or atleast people thought so until they had to combat with scenarios where either of the key or the Plain text matrix was non-invertible.

When the given Key is non-invertible, either of the two methods given below can be followed so as to transmit the message using the hill cipher.

BIBILOGRAPHY

1. Rushdi A. Hamamreh, Mousa Farajallah, Computer Engineering Department, Faculty Of Engineering, Al-Quds University, Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher, JCSNS International Journal of Computer Science and Network Security, VOL.9 No.5, published in May 2009
(http://paper.ijcsns.org/07_book/200905/20090502.Pdf)
2. Bibhudendra Acharya, Himanshu Agrawal, Ankit Modi and Upendra Kumar Agrawal Department of E & TC, NIT Raipur, Chhattisgarh-492010, India published in 2010
(http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.671.6921&rep=rep1&type=pdf)
3. A Novel Approach to Hill Cipher, Neha Sharma, Sachin Chirgaiya, Department of Computer Science and Engineering, Oriental University, Indore, India published in December 2014
4. Miss Anjali Saxena, Mr. Harsh Lohiya, Mr. Kailash Patidar, Department of Computer Science, Shree Sahtya Sai University, published in 2017
(https://www.ijarse.com/images/fullpdf/1512740686_1139_IJARSE.pdf)