

Pandit Deendayal Energy University

School of Technology

Information Security Lab

B.Tech-Computer Science & Engineering (Sem-V)

PATEL VEDANT H.

19BCP138

DIVISION – 2

Lab 9 Assignment

❖ **Aim:** Study and Implement three programs using Crypto Library.

❖ **Introduction:**

Cryptography includes both high-level recipes and low-level interfaces to common cryptographic algorithms such as symmetric ciphers, message digests, and key derivation functions. Cryptography Library is generally used to encrypt real-time data. So here we have used SymPy. SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible to be comprehensible and easily extensible. SymPy is written entirely in Python. Using SymPy we can use ciphers such as:

- Shift cipher
- Affine cipher
- Substitution ciphers
- Vigenere's cipher
- Hill's cipher
- Bifid ciphers
- RSA
- Kid RSA
- Linear-feedback shift registers (for stream ciphers)
- ElGamal encryption

❖ Program:

➤ Encrypt and Decrypt Python Program:

```
D:\Sem5\Information Security\Lab\Lab9\IS-lab-9-Encrypt-Decrypt.py

temp.py x IS-lab-9-Encrypt-Decrypt.py x

1  #-*- coding: utf-8 -*-
2  """
3  Created on Thu Nov 11 19:39:38 2021
4
5  @author: vedpa
6  """
7
8  import rsa
9  from sympy.crypto.crypto import encipher_vigenere, decipher_vigenere
10 from sympy.crypto.crypto import encipher_hill, decipher_hill
11 from sympy import Matrix
12
13 message = input("Enter Message: ")
14 print("Original string: ", message)
15
16 #Asymmetric-key (RSA) Algorithm
17 publicKey, privateKey = rsa.newkeys(512)
18 encryptMessage = rsa.encrypt(message.encode(), publicKey)
19 print("\nAsymmetric-key Encryption (RSA): ", encryptMessage)
20 decryptMessage = rsa.decrypt(encryptMessage, privateKey).decode()
21 print("Asymmetric-key Decryption: ", decryptMessage)
22
23 #Vigenere Algorithm
24 vigenere_encrypt = encipher_vigenere(message, "ved")
25 print("\nEncryption by Vigenere algorithm (RSA): ", vigenere_encrypt)
26 vigenere_decrypt = decipher_vigenere(vigenere_encrypt, "ved")
27 print("Decryption by Vigenere algoithm: ", vigenere_decrypt)
28
29 #Hill Cypher Algorithm
30 hill_key = Matrix([[1,2,3],[0,1,4],[5,6,0]])
31 hill_encrypt = encipher_hill(message, hill_key)
32 print("\nEncryption by Hill algorithm: ", hill_encrypt)
33 hill_decrypt = decipher_hill(hill_encrypt, hill_key)
34 print("Decryption by Hill algoithm: ", hill_decrypt)
35
```

➤ Encrypt and Decrypt Output:

```
Console 1/A x

Python 3.8.11 (default, Aug 6 2021, 09:57:55) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/Sem5/Information Security/Lab/Lab9/IS-lab-9-Encrypt-
Decrypt.py', wdir='D:/Sem5/Information Security/Lab/Lab9')

Enter Message: ILOVEINDIA
Original string: ILOVEINDIA

Asymmetric-key Encryption (RSA): b'\x0c\x0f\xa7\xb1\x1d\xbb\x84\x9cN
\xb7\xa6f\x1c\x03\xfc\xd2\xff\x9f\x55SD\xcd?\xbdj\x97\xb0\x7f\x8c~0
\xa9\xde-Ft\x03 \xab\x96\xb2\xab\x01\xe0f\xa9u\xbbf\x94\xcf\xa8\xdd
\xa2\xd6\x83u\xdc\xfd'
Asymmetric-key Decryption: ILOVEINDIA

Encryption by Vigenere algorithm (RSA): DPRQILHLV
Decryption by Vigenere algoithm: ILOVEINDIA

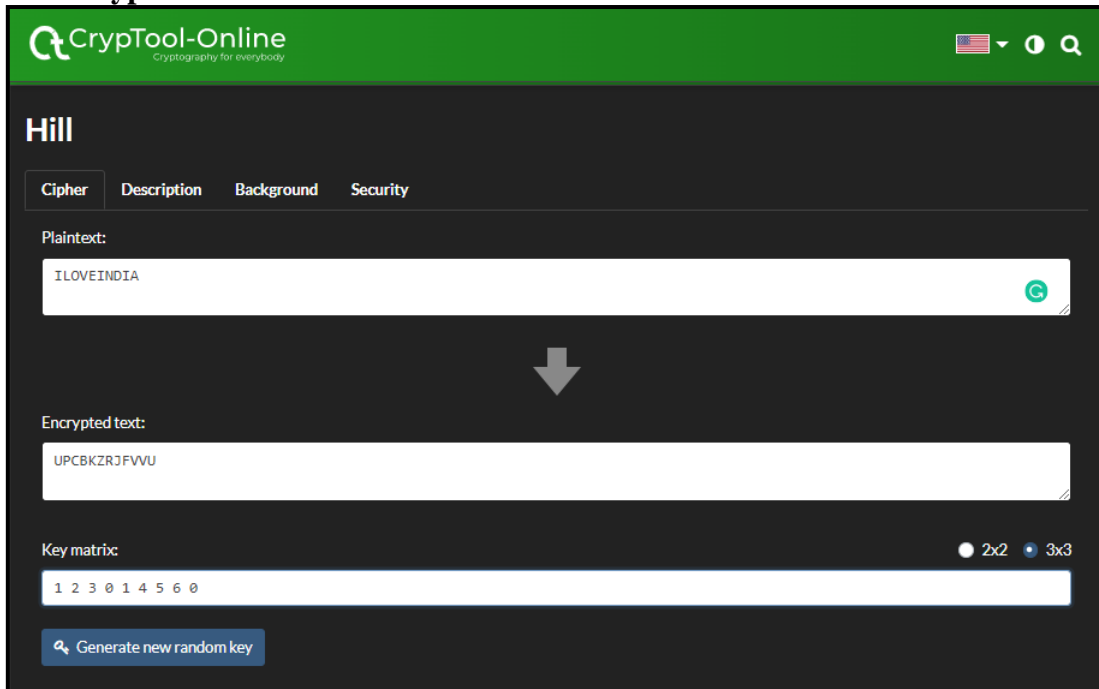
Encryption by Hill algorithm: UPCBKZRJFCCS
Decryption by Hill algoithm: ILOVEINDIAQQ

In [2]: |

IPython console History
```

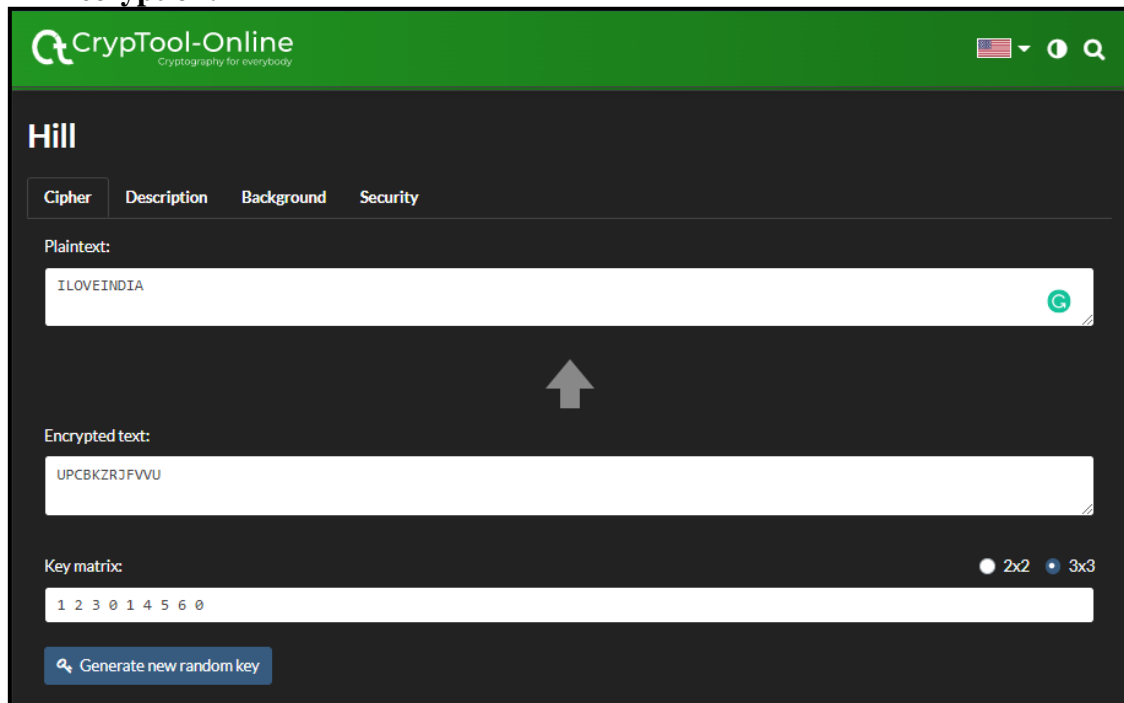
➤ CrypTool Online Encrypt and Decrypt Output:

- **Encryption:**



The image shows the CrypTool-Online encryption interface. At the top is a green header with the logo and text "CrypTool-Online" and "Cryptography for everybody". Below the header, the word "Hill" is displayed. There are four tabs: "Cipher", "Description", "Background", and "Security". The "Cipher" tab is selected. Under "Plaintext:", there is a text input field containing "ILOVEINDIA". A large grey arrow points down to the "Encrypted text:" section, which contains a text input field with the output "UPCBKZRJFVVU". Below this, the "Key matrix:" section shows a 2x2 matrix with the values "1 2 3 0 1 4 5 6 0". To the right of the matrix are radio buttons for "2x2" (selected) and "3x3". At the bottom left is a button labeled "Generate new random key".

- **Decryption:**



The image shows the CrypTool-Online decryption interface. It has the same header and "Hill" title as the encryption interface. The "Cipher" tab is selected. Under "Plaintext:", there is a text input field containing "ILOVEINDIA". A large grey arrow points up to the "Encrypted text:" section, which contains a text input field with the input "UPCBKZRJFVVU". Below this, the "Key matrix:" section shows the same 2x2 matrix with the values "1 2 3 0 1 4 5 6 0". To the right of the matrix are radio buttons for "2x2" (selected) and "3x3". At the bottom left is a button labeled "Generate new random key".

❖ Cryptanalysis:

➤ Cryptanalysis for Vigenere Cipher:

Once the length of the keyword is known we can break Vigenere Cipher easily. Suppose the length of the keyword is n , we can break the ciphertext into n cosets and attack the cipher using frequency analysis if the ciphertext sample is long enough. Two methods are used to determine the length of keyword Friedman and Kasiski tests. In the Kasiski Test occasional alignment of groups of letters with the keyword is used to determine the length of the keyword. This will produce repeated groups of letters in the ciphertext. By counting the number of letters between the beginnings of these repeated groups of letters and finding a number which is the multiple of those distances, we can estimate the length of the keyword. In Friedman given equation is used to determine Keyword length.

$$k \approx \frac{0.0265n}{(0.065 - I) + n(I - 0.085)}$$

➤ Cryptanalysis for Hill Cipher:

When attempting to crack a Hill cipher, frequency analysis will be practically useless, especially as the size of the key block increases. For very long ciphertexts, frequency analysis may be useful when applied to bigrams (for a 2 by 2 hill cipher), but for short ciphertexts, this will not be practical. For a guide on how to break Hill ciphers with a crib, see Cryptanalysis of the Hill Cipher. The basic Hill cipher is vulnerable to a known-plaintext attack, however, (if you know the plaintext and corresponding ciphertext the key can be recovered) because it is completely linear. An opponent who intercepts several plaintext/ciphertext character pairs can set up a linear system that can (usually) be easily solved; if this system is indeterminate, it is only necessary to add a few more plaintexts/ciphertext pairs. The known ciphertext attack is the best one to try when trying to break the hill cipher, if no sections of the plaintext are known, guesses can be made.

➤ Cryptanalysis for RSA:

We can prove, just using the specification of CBC-MAC, that the messages $b || (M(b) \oplus M(a) \oplus b)$ and $a || b$ shares the same tag. This approach is a common method used in cryptanalysis. If you were to use CBC-MAC in a protocol, it provides information about specific weaknesses and how not to use them.

❖ Applications:

- The Vigenère cipher was used during the Civil War.
- Vigenère ciphers are often used in pop culture and fun cryptographical activities like geocaching and CTFs.
- Used by fans to solve a puzzle hidden in the expansion pack of the video game Destiny 2.
- The protection of valuable data in a multimedia system is one of today's most challenging tasks for information technology. Hill cipher belongs in the polygram substitution case of ciphers and gives an inexpensive, easy, and robust tool for multimedia security.
- RSA encryption is often used in combination with other encryption schemes, or for digital signatures which can prove the authenticity and integrity of a message.
- RSA encryption can be used in several different systems. It can be implemented in OpenSSL, wolfCrypt, cryptlib, and several other cryptographic libraries.
- As one of the first widely used public-key encryption schemes, RSA laid the foundations for much of our secure communications. It was traditionally used in TLS and was also the original algorithm used in PGP encryption. RSA is still seen in a range of web browsers, email, VPNs, chat, and other communication channels.

❖ Reference:

- [Practical Cryptography](#)
- <https://www.boxentriq.com/code-breaking/vigenere-cipher>
- [Cryptography — SymPy 1.9 documentation](#)
- <https://www.comparitech.com/blog/information-security/rsa-encryption/>
- <https://www.geeksforgeeks.org/how-to-encrypt-and-decrypt-strings-in-python/>