



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

## APPLIED CRYPTOGRAPHY AND NETWORK SECURITY

**NAME** : MOTHISHWARAN C.  
**REG No** : 19MID0017  
**COURSE CODE** : CSI3002  
**ASSESSMENT NO** : 01  
**FACULTY** : DR. A. MARY MEKALA

# CIPHER ALGORITHMS

## 1) *CEASER CIPHER:*

```
ceaser_cipher.py (Lab2) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

ceaser_cipher.py
1 def a2d(text):
2     return [ord(i) for i in text]
3
4
5 def encrypt(text, key):
6     dtext = a2d(text)
7     result = []
8     for i in dtext:
9         if (i >= 65 and i <= 90):
10            result.append((((i - 65) + key) % 26) + 65)
11        elif (i >= 97 and i <= 122):
12            result.append((((i - 97) + key) % 26) + 97)
13        else:
14            result.append(i)
15    final = list(map(chr, result))
16    return ''.join(final)
17
18
19 def decrypt(text, key):
20     dtext = a2d(text)
21     result = []
22     for i in dtext:
23         if (i >= 65 and i <= 90):
24            result.append((((i - 65) - key) % 26) + 65)
25        elif (i >= 97 and i <= 122):
26            result.append((((i - 97) - key) % 26) + 97)
27        else:
28            result.append(i)
29    final = list(map(chr, result))
30    return ''.join(final)
31
32
33 text = input("Enter you plain text: ")
34
```

```
caesar_cipher.py
31
32
33 text = input("Enter you plain text: ")
34 key = int(input("Enter your key: "))
35 cipher = encrypt(text, key)
36 print("Encrypting...")
37 print(f"Cipher text : {cipher}")
38 plain = decrypt(cipher, key)
39 print("Decrypting...")
40 print(f"Plain text : {plain}")
41
```

## Output:

```
*REPL* [python]
31
32
33 text = input("Enter you plain text: ")
34 key = int(input("Enter your key: "))
35 cipher = encrypt(text, key)
36 print("Encrypting...")
37 print(f"Cipher text : {cipher}")
38 plain = decrypt(cipher, key)
39 print("Decrypting...")
40 print(f"Plain text : {plain}")
41
```

```
Enter you plain text: this is mothish.
Enter your key: 3
Encrypting...
Cipher text : wklv lv prwklvk.
Decrypting...
Plain text : this is mothish.

***Repl Closed***
```

## 2) PLAYFAIR CIPHER:

```
playfair.py (Lab2) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
playfair.py
FOLDERS
Lab2
.vscode
* 19MDD0017.docx
* cipher_cipher.py
* cipher.py
* playfair.py
* vigen.py
* vigen.py
* tempCodeRunnerFile.py

playfair.py
1 import string
2
3
4 def makelist(key_word):
5     azlist = list(string.ascii_lowercase)
6     for i in key_word:
7         if i in azlist:
8             azlist.remove(i)
9             # to make i and j in the same cell
10            azlist.remove("j")
11            return key_word + azlist
12
13
14 def unic(key_word):
15     final = []
16     for i in key_word:
17         if i not in final:
18             final.append(i)
19     return final
20
21
22 def check(matrix, pair):
23     lst = []
24     a = matrix.index(pair[0])
25     b = matrix.index(pair[1])
26     for column in range(5):
27         # Same row
28         j = a // 5
29         temp = 5 * j
30         if pair[1] == matrix[temp + column]:
31             ind_a = ((a + 1) % 5) + 5 * j
32             ind_b = ((b + 1) % 5) + 5 * j
33             lst.append(matrix[ind_a])
```

```
playfair.py (Lab2) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
playfair.py
FOLDERS
Lab2
.vscode
* 19MDD0017.docx
* cipher_cipher.py
* cipher.py
* playfair.py
* vigen.py
* vigen.py
* tempCodeRunnerFile.py

playfair.py
34         lst.append(matrix[ind_b])
35         break
36     else:
37         # Same column
38         j = a % 5
39         for row in range(5):
40             if pair[1] == matrix[j + (row * 5)]:
41                 ind_a = (a + 5) % 20
42                 ind_b = (b + 5) % 20
43                 lst.append(matrix[ind_a])
44                 lst.append(matrix[ind_b])
45                 break
46     else:
47         # diff row and column
48         x = a % 5
49         y = b % 5
50         z = x - y
51         if(z < 0):
52             z = abs(z)
53             lst.append(matrix[a + z])
54             lst.append(matrix[b - z])
55         else:
56             z = abs(z)
57             lst.append(matrix[a - z])
58             lst.append(matrix[b + z])
59     return lst
60
61
62 def split(word):
63     tempList = [word[0]]
64     for i in range(1, len(word)):
65         if word[i] != word[i - 1]:
66             tempList.append(word[i])
```

playfair.py (Lab2) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES

playfair.py

FOLDERS

Lab2

.vscode

\* 19MID0017.docx

caesar\_cipher.py

hallophen.py

playfair.py

vsigs.py

tempCodeRunnerFile.py

```
64 for i in range(1, len(word)):
65     if word[i] != word[i - 1]:
66         templst.append(word[i])
67     else:
68         templst.append('x')
69         templst.append(word[i])
70 if len(templst) % 2 != 0:
71     templst.append('x')
72 return [(templst[i], templst[i + 1]) for i in range(0, len(templst), 2)]
73
74
75 def encipher(matrix, word_pairs):
76     result = []
77     for i in word_pairs:
78         a = (check(matrix, i))
79         result.extend(a)
80     return "".join(result)
81
82
83 key_text = input("Enter ur key_word: ")
84 key_text = key_text.replace("j", "i")
85 key_list = list(key_text)
86 finalized = uniq(key_list)
87 toMatrix = makeList(finalized)
88 word = input("Enter the word: ")
89 word_pairs = split(list(word))
90 print("Encrypting...")
91 print(f"Your cipher text is {encipher(toMatrix, word_pairs)}")
92
```

Line 16, Column 23

UTF-8 | Spaces: 4 | Python

## Output:

\*REPL\* [python] (Lab2) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

GROUP 1

playfair.py

GROUP 2

\*REPL\* [python]

FOLDERS

Lab2

.vscode

\* 19MID0017.docx

caesar\_cipher.py

hallophen.py

playfair.py

vsigs.py

tempCodeRunnerFile.py

```
64 for i in range(1, len(word)):
65     if word[i] != word[i - 1]:
66         templst.append(word[i])
67     else:
68         templst.append('x')
69         templst.append(word[i])
70 if len(templst) % 2 != 0:
71     templst.append('x')
72 return [(templst[i], templst[i + 1]) for i in range(0,
73
74
75 def encipher(matrix, word_pairs):
76     result = []
77     for i in word_pairs:
78         a = (check(matrix, i))
79         result.extend(a)
80     return "".join(result)
81
82
83 key_text = input("Enter ur key_word: ")
84 key_text = key_text.replace("j", "i")
85 key_list = list(key_text)
86 finalized = uniq(key_list)
87 toMatrix = makeList(finalized)
88 word = input("Enter the word: ")
89 word_pairs = split(list(word))
90 print("Encrypting...")
91 print(f"Your cipher text is {encipher(toMatrix, word_pairs)}")
92
```

Enter ur key\_word: playfairexample  
Enter the word: welcome  
Encrypting...  
Your cipher text is vxrnsexm  
\*\*\*Repl Closed\*\*\*

Line 7, Column 1

UTF-8 | Tab Size: 4 | Python

### 3) HILL CIPHER:

hillocipher.py (Lab2) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

playfas.py x hillocipher.py x

```
1 import numpy as np
2 import string
3 import sympy
4
5
6 def gcd(a, b):
7     # Euclidean GCD
8     Q = [] # quotient list
9     while(True):
10         Q.append(a // b)
11         a, b = b, a % b # change value
12         if b == 0:
13             break
14     return a, Q
15
16
17 def mmi(Q):
18     t1 = 1
19     t2 = 0
20     t = 0
21     for i in Q:
22         t = t1 - t2 * i
23         t1, t2 = t2, t
24     return t1
25
26
27 def split(msg):
28     alpha = {}
29     for i, j in enumerate(string.ascii_lowercase):
30         alpha[j] = i
31     lst = []
32     n = len(msg)
33     if n % 2 != 0:
```

Line 13, Column 18

UTF-8 Spaces: 4 Python

hillocipher.py (Lab2) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

playfas.py x hillocipher.py x

```
31 lst = []
32 n = len(msg)
33 if n % 2 != 0:
34     msg = msg + "a"
35 for i in range(0, n, 2):
36     lst.append([alpha[msg[i]], alpha[msg[i + 1]]])
37 return lst
38
39
40 def number2Str(word, n):
41     result = []
42     for i in range(len(word)):
43         for j in range(n):
44             result.append(chr(word[i][j] + 97))
45     return ''.join(result)
46
47
48 def matrixMultiply(m1, m2):
49     return list(np.dot(m1, m2) % 26)
50
51
52
53 def getKey():
54     n = int(input("Enter 'n' value : "))
55     while(True):
56         key_text = input("Enter your key: ")
57         if(len(key_text) == n * n):
58             break
59         else:
60             print("Try a different key: ")
61     return n, key_text
62
63
```

Line 13, Column 18

UTF-8 Spaces: 4 Python

hilcipher.py (Lab2) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

playfair.py hilcipher.py

```
64 def encipher(skey, msg, n):
65     result = []
66     sMsg = split(msg)
67     for i in sMsg:
68         result.append(matrixMultiply(skey, i))
69
70     return number2Str(result, n)
71
72
73 def decipher(skey, msg, n):
74     sMsg = split(msg)
75     det = int(np.linalg.det(skey))
76     if det == 0:
77         return "Decryption is not possible with this key"
78     nGcd, Q = gcd(det, 26)
79     if nGcd != 1:
80         return "Inverse doesn't exist. Hence we can't decrpt"
81     else:
82         keyMatrix = sympy.Matrix(skey)
83         invKey = keyMatrix.adjugate().tolist()
84         invK = mmi(Q)
85         for i in range(n):
86             for j in range(n):
87                 if invKey[i][j] < 0:
88                     invKey[i][j] = (invKey[i][j] + 26) * invK
89                 else:
90                     invKey[i][j] = invKey[i][j] * invK
91     sMsg = split(msg)
92     result = []
93     for i in sMsg:
94         result.append(matrixMultiply(invKey, i))
95
96     return number2Str(result, n)
```

Line 13, Column 18

UTF-8 Spaces: 4 Python

hilcipher.py (Lab2) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

playfair.py hilcipher.py

```
91 sMsg = split(msg)
92 result = []
93 for i in sMsg:
94     result.append(matrixMultiply(invKey, i))
95
96     return number2Str(result, n)
97
98
99
100 # driver code:
101 n, key_text = getKey()
102 msg = input("Enter your message: ")
103 skey = split(key_text)
104 encrypted = encipher(skey, msg, n)
105 print("Encrypting...")
106 print(f"Your cipher text is '{encrypted}'")
107 decrypted = decipher(skey, encrypted, n)
108 print("Decrypting...")
109 print(f"Your plain text is '{decrypted}'")
110
```

Line 13, Column 18

UTF-8 Spaces: 4 Python

## Output:

The screenshot shows a Sublime Text editor window titled "\*REPL\* [python] (Lab2) - Sublime Text (UNREGISTERED)". The editor has three tabs: "playfair.py", "hillcipher.py", and "\*REPL\* [python]". The "playfair.py" tab is active, showing the following code:

```
91 sMsg = split(msg)
92 result = []
93 for i in sMsg:
94     result.append(matrixMultiply(invKey, i))
95
96 return number2Str(result, n)
97
98
99
100 # driver code:
101 n, key_text = getKey()
102 msg = input("Enter your message: ")
103 sKey = split(key_text)
104 encrypted = encipher(sKey, msg, n)
105 print("Encrypting...")
106 print(f"Your cipher text is '{encrypted}'")
107 decrypted = decipher(sKey, encrypted, n)
108 print("Decrypting...")
109 print(f"Your plain text is '{decrypted}'")
110
```

The "\*REPL\* [python]" tab shows the output of the script:

```
Enter 'n' value : 2
Enter your key: test
Enter your message: welcome
Encrypting...
Your cipher text is 'sejccmyu'
Decrypting...
Your plain text is 'welcomes'
***Repl Closed***
```

The status bar at the bottom indicates "Line 10, Column 1", "UTF-8", "Tab Size: 4", and "Python".