Practical5

Aim: Implement Hill cipher encryption-decryption.

Theory:

TheHillcipherisa**symmetrickeysubstitutioncipher**thatencryptsblocksofplaintextusing linear transformations based on matrix multiplication.

- **Block Cipher**: Operates on*n*-lengthblocksoftextinsteadofsinglecharacters.
- **Key**: A square matrix (size*nxn*) ofnumbersmodulo 26.
- **Encryption**:Usesmatrixmultiplicationoftheplaintextvectorwiththekeymatrix.
- **Decryption**:Requirestheinverseofthekeymatrixmodulo26.

EncryptionProcess

1. **KeyMatrix Setup**:
   - Chooseean*n*×*n*matrixwith integersmodulo26 (A=0,B=1,..., Z=25). o Ensure the matrix is invertible modulo 26 (i.e., its determinant has a modular inverse mod 26).
2. **Plaintext Preparation**:
   - Converteachlettertoanumber(A=0to Z=25).
   - Groupplaintext intoblocksofsize *n*. Ifneeded, padwith filler characters.
3. **MatrixMultiplication**:
   - Multiplyeachblockasacolumnvectorbythekeymatrix.
   - Takemodulo26oftheresult.
4. **Conversionto Ciphertext**:

 Convertresultingnumericvaluesbacktoletters.

DecryptionProcess

1. **InverseKeyMatrix**:

o Computeinverseofthekeymatrixmodulo26.Thisrequiresmodulararithmetic and matrix operations.

2. **CiphertextProcessing**: o Convertciphertextintonumber blocks.

3. **DecryptBlocks**: o Multiplyeach blockby theinversekeymatrix modulo 26.
4. **Convertto Plaintext**:

- Mapnumericresultsbackto letters.

Code:

```
importnumpy asnp fromsympyimport Matrix
fromnumpy.linalgimportLinAlgError
defletter_to_num(letter):
 returnord(letter.upper())-ord('A') def
num_to_letter(num):
 returnchr((num%26)+ord('A')) def
text_to_nums(text):
return[letter_to_num(c)forcintextifc.isalpha()]

defnums_to_text(nums):
return''.join([num_to_letter(n)for nin nums])

defpad_text(text,block_size): pad_len=block_size- len(text)%block_size
returntext+'X'*pad_len ifpad_len!=block_sizeelse text

defencrypt(text,key): size=key.shape[0]
text=pad_text(text.upper().replace("",""),size)
nums = text_to_nums(text)
cipher_nums=[]
```

```
for i in range(0, len(nums), size): block=np.array(nums[i:i+size])
enc_block=np.dot(key,block)%26 cipher_nums.extend(enc_block)
returnnums_to_text(cipher_nums)

def decrypt(cipher, key): size=key.shape[0] try:

sym_key=Matrix(key.tolist()) inv_key=np.array(sym_key.inv_mod(26)).astype(int)

except (ValueError, LinAlgError):

return"Keymatrixisnotinvertiblemod 26!"


nums=text_to_nums(cipher)

plain_nums = []


for i in range(0, len(nums), size): block=np.array(nums[i:i+size])
dec_block=np.dot(inv_key,block)%26 plain_nums.extend(dec_block)
returnnums_to_text(plain_nums)

if name_____== "main":

key_3x3=np.array([[6,24, 1],

[13,16, 10],

[20,17, 15]])


message="ACT"

cipher=encrypt(message,key_3x3) print("Encrypted:", cipher)


decrypted=decrypt(cipher,key_3x3) print("Decrypted:", decrypted)
```

Output:

```
D:\College\Information And Network Security>python -u "d:\College\Information
 And Network Security\prac5.py"
Encrypted: POH
Decrypted: ACT
```

CryptanalysisofillCipherCryptanalysis:

1. **TheHillcipherisvulnerabletoa *known-plaintext attack*.**
   - **If $n \times n$ key size is used, acquiring n^2 plaintext–ciphertext character pairs can compromise the key.**
2. **Theattackmethodinvolveslinearalgebraovermodulararithmetic.**
   - **Constructmatricesofplaintextblocksandciphertextblocks.**
   - **Solveforthekeyusingmatrixinversionandmultiplicationmodulo 26.**
3. **KeyRecoveryEquation: K=C .P^-1(mod 26) Where:**
   - **K=Key matrix o C=Ciphertext matrix**
   - **P^{-1}=Inverseofplaintext matrixmodulo 26**
4. **Thecipherlacksresistancetostatistical analysis.**
   - **Becauseit'sdeterministic,patternsinciphertextcloselyreflectpatternsin plaintext blocks.**
5. **Poorkeymatrixchoicecancreatevulnerabilities.**
   - **Anon-invertiblekeymatrix(mod26)breaksthecipher's reversibility. o Ifdeterminantofkey≡0(mod26)orhasnomodularinverse→cipherbecomes unusable. 6. Hillcipherdoesn'tprovidediffusionandconfusion.**
   - **Changesininputdon'tripplefar;outputchangesarelocalizedwithinthebl ock.**
7. **Attackfeasibilityincreaseswithblocksizeandplaintextvolume.**

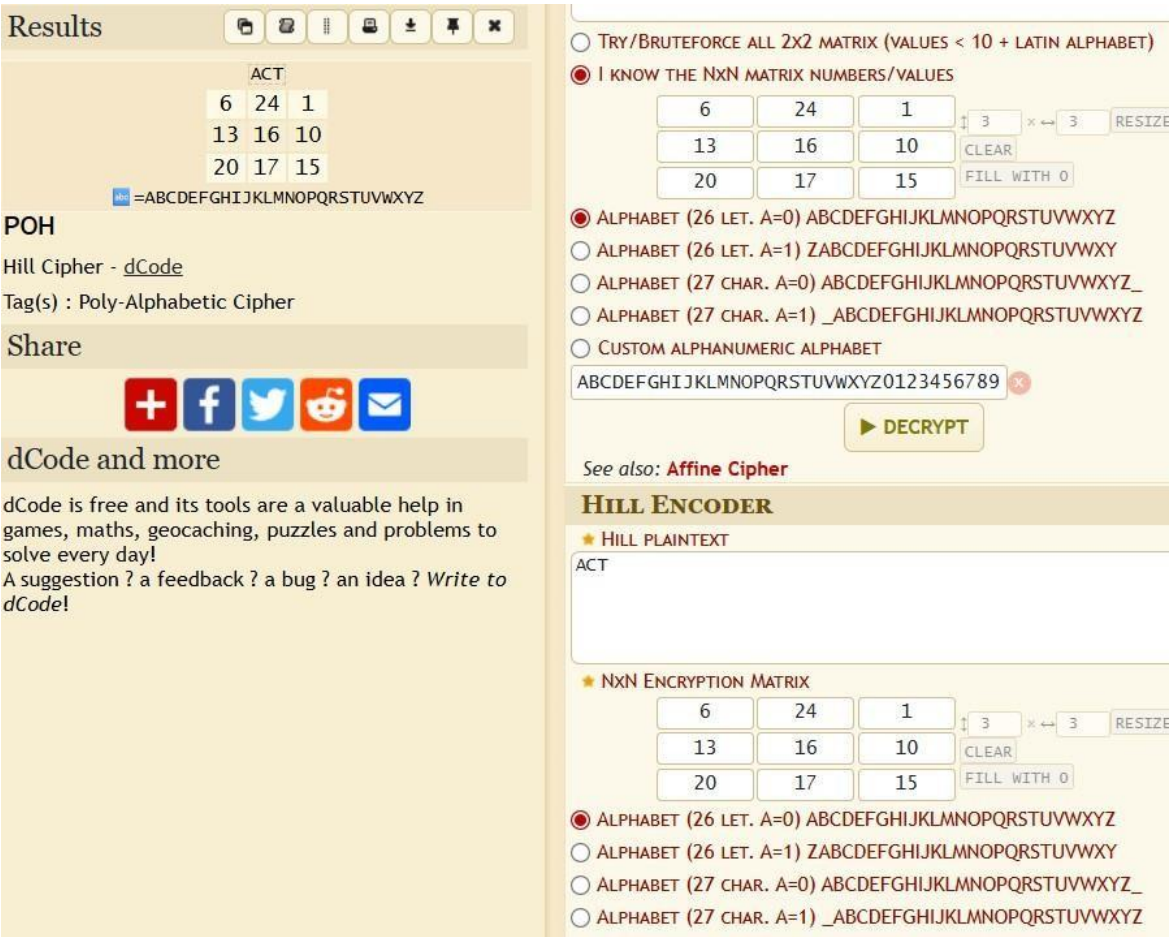
o Largerblocksneedmoreplaintext–ciphertextpairsbutmakestatisticalrecovery easier if data is available.

8. **Nopaddingstandardmakesplaintextrecoveryeveneasier.**

- **Attackerscanguesspaddingschemesorinferprobabletextstructure.**

Tool:


Search for a tool
SEARCH A TOOL ON DCODE
e.g. type 'sudoku'
BROWSE THE FULL DCODE TOOLS' LIST
Results
POH
6 24 1
13 16 10
20 17 15
=ABCDEFGHIJKLMNOPQRSTUVWXYZ
ACT
Hill Cipher - dCode
HILL CIPHER
Cryptography › Poly-Alphabetic Cipher › Hill Ciphe
HILL DECODER
HILL CIPHERTEXT
POH
TRY/BRUTEFORCE ALL 2X2 MATRIX (VALUES < 10 + LAT
I KNOW THE NXN MATRIX NUMBERS/VALUES
6 24 1
13 16 10
20 17 15
CLEAR
ALPHABET (26 LET. A=0) ABCDEFGHIJKLMNOPQRST
ALPHABET (26 LET. A=1) ZABCDEFGHIJKLMNOPQRS
ALPHABET (27 CHAR. A=0) ABCDEFGHIJKLMNOPQRS


Results
ACT
6 24 1
13 16 10
20 17 15
=ABCDEFGHIJKLMNOPQRSTUVWXYZ
POH
Hill Cipher - dCode
Tag(s) : Poly-Alphabetic Cipher
Share
dCode and more
dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!
A suggestion ? a feedback ? a bug ? an idea ? Write to dCode!
TRY/BRUTEFORCE ALL 2X2 MATRIX (VALUES < 10 + LATIN ALPHABET)
I KNOW THE NXN MATRIX NUMBERS/VALUES
6 24 1
13 16 10
20 17 15
RESIZE
CLEAR
FILL WITH 0
ALPHABET (26 LET. A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
ALPHABET (26 LET. A=1) ZABCDEFGHIJKLMNOPQRSTUVWXY
ALPHABET (27 CHAR. A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ_
ALPHABET (27 CHAR. A=1) _ABCDEFGHIJKLMNOPQRSTUVWXYZ
CUSTOM ALPHANUMERIC ALPHABET
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
DECRYPT
See also: Affine Cipher
HILL ENCODER
HILL PLAINTEXT
ACT
NXN ENCRYPTION MATRIX
6 24 1
13 16 10
20 17 15
RESIZE
CLEAR
FILL WITH 0
ALPHABET (26 LET. A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
ALPHABET (26 LET. A=1) ZABCDEFGHIJKLMNOPQRSTUVWXY
ALPHABET (27 CHAR. A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ_
ALPHABET (27 CHAR. A=1) _ABCDEFGHIJKLMNOPQRSTUVWXYZ