

Implementation of Modular Matrix Inverse for the Hill Cipher

This document outlines the method used to compute the modular inverse of a 2x2 matrix, which is the crucial step for decryption in the Hill Cipher. The implementation follows the standard mathematical formula for a matrix inverse, adapted for modular arithmetic:

$$K^{-1} \equiv (\det(K))^{-1} \cdot \text{adj}(K) \pmod{26}$$

The process was broken down into four distinct steps in the code:

1. Calculate the Determinant

The first step is to compute the determinant of the key matrix $K=(acbd)$.

- **Action:** The formula $\det(K)=ad-bc$ is applied.
- **Implementation:** The result is immediately reduced modulo 26 to keep the value within the required range. A critical check is performed to ensure the determinant is coprime with 26 (i.e., their greatest common divisor is 1). If not, an error is thrown as the matrix is not invertible.

2. Find the Modular Multiplicative Inverse

This step finds the modular multiplicative inverse of the determinant, which is the equivalent of "dividing" by the determinant in modular arithmetic.

- **Action:** Find a number, d_{inv} , such that $(\det(K) \cdot d_{inv}) \pmod{26} = 1$.
- **Implementation:** For the small modulus of 26, a simple and efficient iterative search was implemented. The code loops through numbers from 1 to 25, checking which one satisfies the equation. For larger moduli, the Extended Euclidean Algorithm would be used.

3. Determine the Adjugate Matrix

The adjugate of the matrix is calculated next.


- **Action:** For a 2x2 matrix, this involves swapping the elements on the main diagonal and negating the elements on the off-diagonal.
- **Implementation:** The adjugate matrix, $\text{adj}(K)=(d-c-ba)$, is constructed in memory.

4. Compute the Final Inverse Matrix

The final step combines the results from the previous steps to form the inverse matrix.

- **Action:** Each element of the adjugate matrix is multiplied by the modular inverse of the determinant (d_{inv}).
- **Implementation:** Each product is taken modulo 26. A custom modulo function $(a \% m + m) \% m$ is used to ensure that any negative results (from the negated elements of the

adjugate matrix) are correctly converted to their positive equivalents in the range $[0, 25]$. This yields the final inverse key matrix, K^{-1} , ready for decryption.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Sweta Rana\OneDrive\Desktop\ISC LAB> cd "c:\Users\Sweta Rana\OneDrive\Desktop\ISC LAB\ISC LAB_U23AI065\LAB6\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?) { .\1 }
● Encrypting...
Plaintext block: [7, 4]
Ciphertext block: [3, 15]
Plaintext block: [11, 15]
Ciphertext block: [11, 4]
Final Ciphertext: DPLE

Decrypting...
Inverse Key Matrix K_inv:
[ 15 17 ]
[ 20 9 ]
Ciphertext block: [3, 15]
Decrypted block: [7, 4]
Ciphertext block: [11, 4]
Decrypted block: [11, 15]
Final Decrypted Plaintext: HELP
○ PS C:\Users\Sweta Rana\OneDrive\Desktop\ISC LAB\ISC LAB_U23AI065\LAB6>
```