103cipher: Matrix-Based Cryptography

Alejandro Laguna, Pol Cortadellas

January 7, 2025

Contents

1 Introduction

The 103cipher program implements a matrix-based encryption and decryption system.

2 Mathematical Foundations

2.1 Matrix Theory Background

In matrix-based cryptography, we rely heavily on several key mathematical concepts:

1. **Matrix Multiplication**: For matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$:

$$C = \sum_{k=1}^{n} a_{ik} b_{kj}$$

2. **Matrix Invertibility**: A matrix A is invertible if there exists A^{-1} such that:

$$AA^{-1} = A^{-1}A = I$$

3. **Determinant**: For a 2×2 matrix:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

For larger matrices, we use the Laplace expansion:

$$\det(A) = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(M_{1j})$$

where M_{1j} is the minor matrix.

2.2 Key Matrix Generation

Given a key string of length k, we generate a square matrix of size $n \times n$ where:

$$n = \left\lceil \sqrt{k} \right\rceil$$

The key matrix K is populated as follows:

$$K_{ij} = \begin{cases} \text{ASCII}(\text{key}[i \cdot n + j]) & \text{if } i \cdot n + j < k \\ 0 & \text{otherwise} \end{cases}$$

3 Encryption Process

3.1 Message Matrix Formation

For a message of length m, we create a matrix M with dimensions:

$$\left\lceil \frac{m}{n} \right\rceil \times n$$

The message matrix is populated similarly to the key matrix:

$$M_{ij} = \begin{cases} \text{ASCII}(\text{message}[i \cdot n + j]) & \text{if } i \cdot n + j < m \\ 0 & \text{otherwise} \end{cases}$$

3.2 Encryption Algorithm

The encryption process follows the matrix multiplication:

$$C = MK$$

Where:

- C is the cipher matrix
- M is the message matrix
- K is the key matrix

4 Decryption Process

4.1 Matrix Inversion Algorithm

The decryption process requires finding K^{-1}. This is done through the following steps:

4.1.1 Adjugate Matrix Calculation

For each element (i, j), we calculate:

$$\text{adj}(A)_{ij} = (-1)^{i+j} \det(M_{ji})$$

Where M_{ji} is the minor matrix obtained by removing row j and column i.

4.1.2 Inverse Matrix

The inverse is calculated as:

$$K^{-1} = \frac{1}{\det(K)} \text{adj}(K)$$

4.2 Message Recovery

The original message matrix is recovered through:

$$M = CK^{-1}$$

5 Implementation Details

5.1 Key Functions

5.1.1 Determinant Calculation

```python
def determinant(matrix):
    size = len(matrix)
    if size == 1:
        return matrix[0][0]
    if size == 2:
        return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0]

    det = 0
    for col in range(size):
        submatrix = [row[:col] + row[col+1:]
                     for row in matrix[1:]]
        det += ((-1) ** col) * matrix[0][col] * determinant(submatrix)

    return det
```

Listing 1: Determinant Function

5.1.2 Matrix Multiplication

```
def matrix_multiply(mat1, mat2):
    rows_mat1 = len(mat1)
    cols_mat1 = len(mat1[0])
    rows_mat2 = len(mat2)
    cols_mat2 = len(mat2[0])

    result = [[0 for _ in range(cols_mat2)]
              for _ in range(rows_mat1)]

    for i in range(rows_mat1):
        for j in range(cols_mat2):
            for k in range(cols_mat1):
                result[i][j] += mat1[i][k] * mat2[k][j]

    return result
```

Listing 2: Matrix Multiplication

6 Usage Examples

6.1 Encryption Example

Consider the message "HELLO" with key "ABC":

Initial key matrix:

$$K = \begin{pmatrix} 65 & 66 \\ 67 & 0 \end{pmatrix}$$

Message matrix:

$$M = \begin{pmatrix} 72 & 69 \\ 76 & 76 \\ 79 & 0 \end{pmatrix}$$

6.2 Decryption Example

For the above key matrix:

Determinant:

$$\det(K) = 65 \cdot 0 - 66 \cdot 67 = -4422$$

Adjugate matrix:

$$\operatorname{adj}(K) = \begin{pmatrix} 0 & -67 \\ -66 & 65 \end{pmatrix}$$

Inverse matrix:

$$K^{-1} = \frac{1}{-4422} \begin{pmatrix} 0 & -67 \\ -66 & 65 \end{pmatrix}$$