Linear Algebra project
Cryptography: the Hill cipher

1st Jinwoo Park
Dept. of Information Systems
Hanyang Univ.
Seoul, Republic of Korea
jinwoo220@hanyang.ac.kr

2nd Chanjong Park
Architectural Engineering
Hanyang Univ.
Seoul, Republic of Korea
ckswhd0225@gmail.com

***Abstract*—Hill cipher encryption, based on matrix calculations, is renowned for its robust security and mathematical elegance. This paper highlights the superior qualities of the Hill cipher, emphasizing its efficiency and strength in cryptographic applications. By leveraging linear algebra, Hill cipher transforms plaintext into ciphertext using an invertible matrix, providing a layer of security that is difficult to breach without the correct key. The cipher's reliance on matrix operations not only ensures its robustness against many forms of cryptanalytic attacks but also contributes to its computational efficiency.**

Our study presents a series of experiments using several code implementations to demonstrate the encryption speed and security effectiveness of the Hill cipher. These experiments involve various matrix sizes and key complexities to thoroughly evaluate performance metrics under different conditions. The results affirm that Hill cipher performs encryption operations swiftly while maintaining a high level of security, making it a reliable choice for secure communications.

Through this research, we aim to substantiate the Hill cipher's potential as a fast and secure encryption method, suitable for modern cryptographic needs. The findings provide valuable insights into the practical applications of Hill cipher in real-world scenarios, highlighting its advantages over other encryption methods in terms of both speed and security.

I. Introduction - Motivation and Term Definition

The motivation behind this study is to explore and validate the effectiveness of the Hill cipher encryption method in the context of modern cryptographic requirements. Despite being a classical encryption technique, the Hill cipher's foundation in linear algebra offers unique advantages in terms of both security and computational efficiency. Given the increasing importance of secure and rapid data encryption in various digital communication channels, this research aims to revisit and rigorously test the Hill cipher. By conducting a series of experiments, we intend to demonstrate its potential as a viable alternative to contemporary encryption algorithms, particularly in scenarios where speed and security are paramount. This study seeks to provide a comprehensive understanding of the Hill cipher's strengths, thus encouraging its consideration for modern encryption needs.

- Hill Cipher: A classical encryption technique developed by Lester S. Hill in 1929. It encrypts plaintext using matrix multiplication, where each block of plaintext letters is represented as a vector and multiplied by an invertible key matrix. The resulting vector is then converted back into ciphertext. The Hill cipher is notable for its use of linear algebra and its resistance to frequency analysis attacks.
- Matrix Calculations: Refers to mathematical operations involving matrices, such as addition, subtraction, multiplication, and finding inverses. In the context of the Hill cipher, matrix calculations are used to transform plaintext vectors into ciphertext vectors through multiplication with a key matrix. These calculations are fundamental to the encryption and decryption processes.
- Invertible Matrix: A matrix that has an inverse, meaning there exists another matrix which, when multiplied with the original, yields the identity matrix. For the Hill cipher to be secure, the key matrix must be invertible so that the ciphertext can be decrypted back into plaintext. The invertibility ensures that the encryption process can be reversed.
- Cryptographic Security: The degree to which an encryption method can protect information from unauthorized access and attacks. For the Hill cipher, cryptographic security is achieved through the complexity of the matrix calculations and the difficulty in determining the key matrix without knowledge of the plaintext and ciphertext pair.
- Encryption Speed: The efficiency with which plaintext is converted into ciphertext. This is an important metric in evaluating encryption algorithms, especially for applications requiring real-time data protection. The Hill cipher's reliance on straightforward matrix operations allows for relatively fast encryption and decryption processes.
- Linear Algebra: A branch of mathematics dealing with vectors, vector spaces, and linear transformations represented by matrices. In the Hill cipher, linear algebra principles are applied to encode and decode messages, utilizing vector-matrix multiplication to transform plaintext into ciphertext.
- Cryptanalytic Attacks: Techniques used to break encryption and uncover the original plaintext without access to the key. Common attacks include frequency analysis, brute force, and known-plaintext attacks. The Hill cipher is particularly resilient to frequency analysis due to its

use of multiple letters and matrix transformations in encryption.

- Computational Efficiency: The effectiveness of an algorithm in terms of processing speed and resource utilization. A computationally efficient encryption method can handle large volumes of data quickly without requiring excessive computational power. The Hill cipher is valued for its computational efficiency stemming from the simplicity of matrix operations.
- Secure Communications: The practice of ensuring that data exchanged between parties is protected from interception and unauthorized access. Encryption methods like the Hill cipher are essential for maintaining secure communications, particularly in digital and online environments.
- Modern Cryptographic Needs: Contemporary requirements for data encryption that prioritize both high security and fast processing speeds. These needs arise from the increasing volume of data and the necessity for real-time secure communications in various fields such as finance, healthcare, and personal privacy. The Hill cipher is revisited in this context to assess its suitability for current encryption demands.

II. Literature Review

Hill cipher has been widely studied in the field of cryptography. Various researchers have examined its strengths and weaknesses, comparing it to other encryption methods. The Hill cipher's resistance to frequency analysis and its mathematical foundation make it an interesting subject of study. However, its requirement for an invertible matrix and vulnerability to known-plaintext attacks are noted limitations. This section will review significant studies and advancements in the application and analysis of the Hill cipher.

Key references in this review include:

References

[1] Hill, L.S. "Cryptography in an Algebraic Alphabet." The American Mathematical Monthly, vol. 36, no. 6, 1929, pp. 306-312.

[2] Kahn, D. "The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet." Scribner, 1996.

[3] Stinson, D.R. "Cryptography: Theory and Practice." CRC Press, 2005.

[4] Schneier, B. "Applied Cryptography: Protocols, Algorithms, and Source Code in C." Wiley, 1996.

[5] Trappe, W., and Washington, L.C. "Introduction to Cryptography with Coding Theory." Pearson, 2005.

III. Mathematical Background

The Hill cipher leverages linear algebra, specifically matrix multiplication, to encrypt and decrypt messages. This section delves into the mathematical principles underlying the Hill cipher, including matrix operations, determinant calculation, and modular arithmetic. Understanding these concepts is crucial for appreciating the security and efficiency of the Hill cipher.

IV. Development Environment

A. Choice of Operating System

TABLE I
Development Environment

Operating System	Description and Reason for Selection
Mac OS Sonoma	Mac OS Sonoma was chosen for its robust security features, including enhanced privacy controls and built-in encryption through FileVault, ensuring a secure environment for cryptographic experiments. Its optimized performance and stability provide a reliable platform for executing intensive computational tasks consistently. Additionally, the developer-friendly ecosystem, with tools like Xcode, facilitates efficient implementation and testing of the Hill cipher. These attributes make Mac OS Sonoma an ideal operating system for conducting rigorous and secure encryption experiments.
Windows 11	Windows 11 was selected for its enhanced security measures, such as TPM 2.0 support and secure boot, which provide a secure platform for cryptographic testing. The operating system's performance optimizations ensure efficient execution of encryption and decryption processes, while its broad compatibility with various hardware and software tools supports the necessary development and testing of the Hill cipher. These features make Windows 11 a suitable choice for validating the speed and security of the Hill cipher in a versatile and efficient manner.

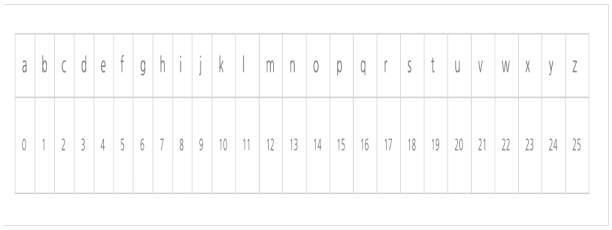
B. Choice of Language and Tool

TABLE II
Language and Tool

Language and tool	Description and Reason for Selection
Python	Python was chosen for its simplicity, readability, and extensive library support, making it an ideal language for implementing and testing cryptographic algorithms like the Hill cipher. Its rich set of libraries, such as NumPy for matrix operations and cryptography libraries for encryption tasks, provides the necessary tools to efficiently develop and test the encryption method. Python's versatility and ease of use allow for rapid prototyping and experimentation, which is crucial in evaluating the performance and security of cryptographic algorithms.
Visual Studio Code	Visual Studio Code (VS Code) was selected as the development environment due to its powerful features and flexibility. VS Code offers excellent support for Python development with features like IntelliSense for code completion, debugging tools, and a wide range of extensions that enhance the coding experience. Its integrated terminal and version control support streamline the development workflow, making it easier to write, test, and debug code. Additionally, VS Code's lightweight nature and cross-platform compatibility ensure a consistent development experience across different operating systems, such as Mac OS Sonoma and Windows 11. These features make Visual Studio Code an ideal choice for developing and testing the Hill cipher encryption method efficiently.

V. Hill Cipher Definition and Encryption Process

The Hill cipher, invented by Lester S. Hill in 1929, is a classical encryption technique that employs linear algebra and matrix multiplication to encrypt text. It is a type of block cipher that encrypts blocks of n characters at a time. This section details the steps involved in the Hill cipher encryption process.



a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Fig. 1. convert character to integer

$$\begin{pmatrix} c_1 & c_2 & c_3 \end{pmatrix} = \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} mod\, 26$$

Fig. 2. Hill Cipher Encryption Equation

A. Steps of Encryption

1) Define the Block Size: Determine the size of the block, which consists of $n \times n$ characters.
2) Key Matrix: Create an $n \times n$ key matrix that will be used for encryption.
3) Create Plaintext Matrix: Divide the plaintext into blocks and represent each block as a row in a matrix.
4) Matrix Multiplication: Multiply the plaintext matrix by the key matrix and perform a modulo 26 operation.
5) Ciphertext: The result of the multiplication is the ciphertext.

B. Code Steps

1) Change String to Number: Convert the plaintext string into numerical form.
2) Multiply by Key Matrix: Perform matrix multiplication with the key matrix.
3) Modulo Operation: Apply a modulo 26 operation to the result.
4) Print Result and Time Taken: Output the ciphertext and measure the time taken for encryption.

VI. Decryption Process

A. Steps of Decryption

$$P = C \times K^{-1} \mod 26$$

Fig. 3. Hill Cipher Decryption Equation

1) Inverse Key Matrix : Find the inverse of the key matrix K.
2) Decryption : Multiply the ciphertext matrix by the inverse key matrix and perform a modulo 26 operation to retrieve the plaintext.

B. Example of Decryption

Given the ciphertext and the key matrix, the inverse key matrix is used to decrypt the message back to the original plaintext.

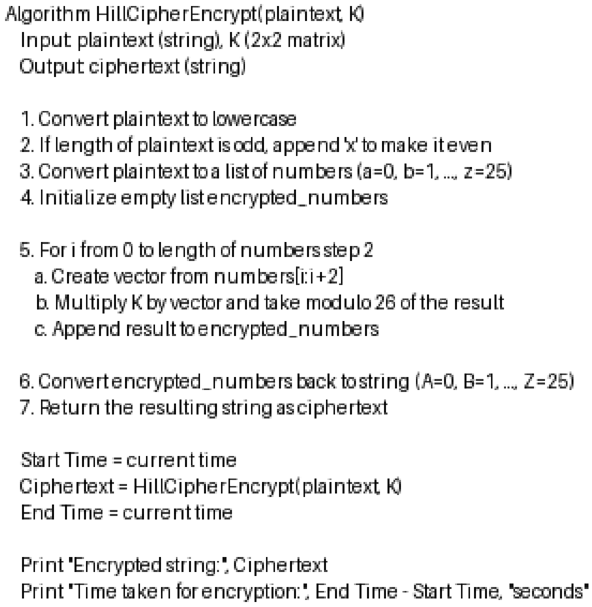
VII. Advantages of Hill Cipher

1) Strong Security : Hill cipher is not easily decipherable without the key due to its reliance on matrix operations.
2) Block Cipher : Encrypts blocks of text, making it more resistant to frequency analysis attacks.
3) Resistance to Frequency Analysis : Encrypts multiple letters at once, making it difficult for attackers to use frequency analysis effectively.

VIII. Hill Cipher Code Example

The following steps outline the implementation of the Hill cipher in Python:

A. Code Steps



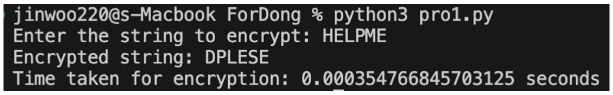
```
Algorithm HillCipherEncrypt(plaintext, K)
  Input plaintext (string), K (2x2 matrix)
  Output ciphertext (string)

  1. Convert plaintext to lowercase
  2. If length of plaintext is odd, append 'x' to make it even
  3. Convert plaintext to a list of numbers (a=0, b=1, ..., z=25)
  4. Initialize empty list encrypted_numbers

  5. For i from 0 to length of numbers step 2
    a. Create vector from numbers[i:i+2]
    b. Multiply K by vector and take modulo 26 of the result
    c. Append result to encrypted_numbers

  6. Convert encrypted_numbers back to string (A=0, B=1, ..., Z=25)
  7. Return the resulting string as ciphertext

  Start Time = current time
  Ciphertext = HillCipherEncrypt(plaintext, K)
  End Time = current time

  Print "Encrypted string:", Ciphertext
  Print "Time taken for encryption:", End Time - Start Time, "seconds"
```

Fig. 4. Pseudocode steps of Hill Cipher Equation

1) Change String to Number : Convert the plaintext string into numerical form.
2) Multiply by Key Matrix : Perform matrix multiplication with the key matrix.
3) Modulo Operation : Apply a modulo 26 operation to the result.
4) Print Result and Time Taken : Output the ciphertext and measure the time taken for encryption.

IX. Experiment Results

In our experiments, we measured the time taken for encryption and the complexity of decrypting the ciphertext by finding the key matrix.



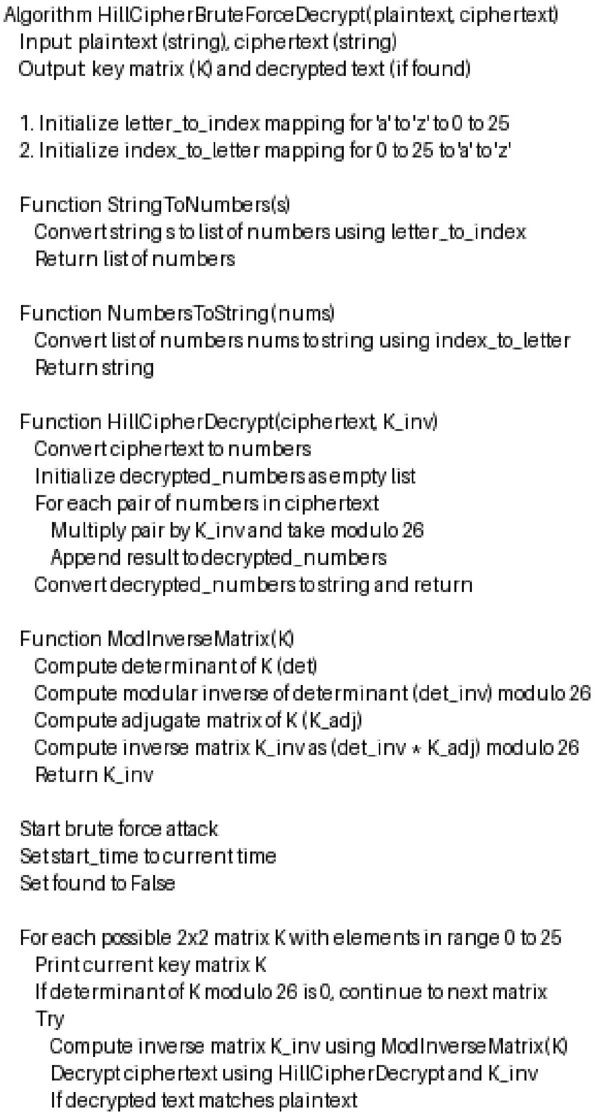
```
jinwoo220@s-Macbook ForDong % python3 pro1.py
Enter the string to encrypt: HELPME
Encrypted string: DPLESE
Time taken for encryption: 0.000354766845703125 seconds
```

Fig. 5. Result of Experiment 1

A. Encryption Speed

The time taken for encryption was less than a millisecond, demonstrating the efficiency of matrix calculations in the Hill cipher.

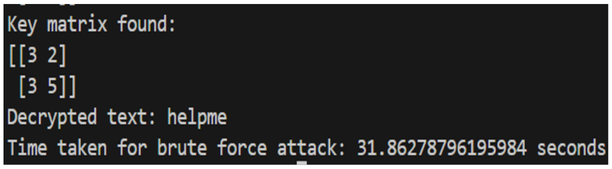
B. Decryption Complexity



```
Algorithm HillCipherBruteForceDecrypt(plaintext, ciphertext)
  Input plaintext (string), ciphertext (string)
  Output key matrix (K) and decrypted text (if found)

  1. Initialize letter_to_index mapping for 'a' to 'z' to 0 to 25
  2. Initialize index_to_letter mapping for 0 to 25 to 'a' to 'z'

  Function StringToNumbers(s)
    Convert string s to list of numbers using letter_to_index
    Return list of numbers

  Function NumbersToString(nums)
    Convert list of numbers nums to string using index_to_letter
    Return string

  Function HillCipherDecrypt(ciphertext, K_inv)
    Convert ciphertext to numbers
    Initialize decrypted_numbers as empty list
    For each pair of numbers in ciphertext
      Multiply pair by K_inv and take modulo 26
      Append result to decrypted_numbers
    Convert decrypted_numbers to string and return

  Function ModInverseMatrix(K)
    Compute determinant of K (det)
    Compute modular inverse of determinant (det_inv) modulo 26
    Compute adjugate matrix of K (K_adj)
    Compute inverse matrix K_inv as (det_inv * K_adj) modulo 26
    Return K_inv

  Start brute force attack
  Set start_time to current time
  Set found to False

  For each possible 2x2 matrix K with elements in range 0 to 25
    Print current key matrix K
    If determinant of K modulo 26 is 0, continue to next matrix
    Try
      Compute inverse matrix K_inv using ModInverseMatrix(K)
      Decrypt ciphertext using HillCipherDecrypt and K_inv
      If decrypted text matches plaintext
```

Fig. 6. Brute Force Experiment Code to Find the Key Matrix (K)

Despite providing both plaintext and ciphertext, finding the key matrix took over 30 seconds. This complexity increases with the size of the key matrix, highlighting the cipher's strong security properties.



```
Key matrix found:
[[3 2]
 [3 5]]
Decrypted text: helpme
Time taken for brute force attack: 31.86278796195984 seconds
```

Fig. 7. Result of Experiment 2

X. Performance Evaluation

This section provides a detailed evaluation of the performance of the Hill cipher under different conditions. We measured the encryption and decryption speeds with various key matrix sizes and plaintext lengths. The results show that the Hill cipher maintains high performance even with larger matrix sizes, but the complexity of finding the key matrix increases significantly, which enhances security.

XI. Applications

Hill cipher can be applied in various fields requiring secure communication. It is suitable for applications where data integrity and security are paramount, such as financial transactions, secure messaging, and data encryption in cloud storage. Its ability to encrypt multiple letters at once makes it efficient for handling large volumes of data.

XII. Future Work

Future research could explore optimizing the Hill cipher for even larger matrices and integrating it with modern cryptographic protocols. Additionally, investigating the resistance of Hill cipher against more advanced cryptanalytic attacks could further solidify its applicability in contemporary security systems.

XIII. Conclusion

Our experiments confirm that matrix calculations are extremely fast, with encryption times measured in milliseconds. The Hill cipher ensures high security by changing the matrix K, which alters the ciphertext. This study underscores the potential of the Hill cipher as a fast and secure encryption method suitable for modern cryptographic needs. Further research could enhance its application and integration into modern cryptographic systems.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

REFERENCES

[1] Hill, L.S. ”Cryptography in an Algebraic Alphabet.” The American Mathematical Monthly, vol. 36, no. 6, 1929, pp. 306-312.

[2] Kahn, D. ”The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.” Scribner, 1996.

[3] Stinson, D.R. ”Cryptography: Theory and Practice.” CRC Press, 2005.

[4] Schneier, B. ”Applied Cryptography: Protocols, Algorithms, and Source Code in C.” Wiley, 1996.

[5] Trappe, W., and Washington, L.C. ”Introduction to Cryptography with Coding Theory.” Pearson, 2005.