Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

A Novel Method for Hill Cipher Encryption and Decryption Using Gaussian Integers Implemented in Banking Systems

Saba Abdulbaqi Salman [1*], Yasmin Makki Mohialden [2], Abbas Abdulhameed[2], Nadia Mahmood Hussien[2]

[1] Aliraqia University, Baghdad, Iraq
[2] Mustansiriyah University, Baghdad, Iraq

*Corresponding Author: Saba Abdulbaqi Salman

DOI: https://doi.org/10.52866/ijcsm.2024.05.01.019
Received August 2023 ; Accepted November 2023 ; Available online February 2024.

ABSTRACT: Cryptographic tools like Hill's encryption algorithm protect digital data. In this work, we present a novel Hill cipher security method that utilizes Gaussian integers from number theory. Using these intriguing mathematical entities to disguise plaintext values dramatically boosts assault resistance and duration. This research includes a three-pass protocol for encryption and decryption without key exchange, ensuring a safe, efficient, and dependable solution. Pandas is used for efficient data processing, and Numpy for computational tasks, notably matrices. Hill cipher-based encryption and decryption can be utilized in real life. It also demonstrates how to save Pandas DataFrame data to Excel. This strategy assures progress in cryptography. Uniquely designed for banking, it emphasizes its applicability and possible influence on present financial systems.

Keywords: Gaussian numbers, banking systems, security, decoding, hill cipher encryption.

1. INTRODUCTION

Protecting banking data is crucial in today's data security and privacy environment. Due to the rise of online commerce, cryptography is crucial to protecting Internet communication and transaction secrecy. To limit access and safeguard email, phone conversations, and financial accounts from hackers, cable and satellite networks utilize encryption. For secrecy, data integrity, user authentication, and non-repudiation, cryptography is essential [4,5].

Cryptography encrypts text. In this effort, encryption is crucial. The decryption process demonstrates the reciprocal nature of encryption by restoring encrypted text [1, 2]. Cryptography draws from diverse fields and is extensively reliant on discrete mathematics. Specifically, modern cryptography places substantial reliance on the number theory [3, 20].

Amidst various encryption techniques, the Hill cipher (HC) stands out for its mathematical complexity and resilience against cryptanalysis. Despite seeming straightforward, it uses linear matrix transformations for symmetric encryption, masking the letter frequencies in plaintext and guaranteeing high throughput. As a polygraphic substitution cipher, the Hill Cipher uses digraphs and trigraphs, or groupings of three letters, and allows for different block sizes. Its encryption method, which uses linear algebra, requires a basic comprehension of matrices. Similar to the Affine cipher, the Hill cipher employs modulo algebra. The decryption process involves computing the inverse of the fundamental matrix. However, the primary drawback of this is the requirement for an invertible full matrix [10, 11, 22]. In addition, traditional methods of implementation of this cipher limit its potential, particularly when confronted with complex numbers or the need for heightened security. Further, this technique is vulnerable to plaintext-ciphertext attacks [7, 9].

This article delves into matrix extensions of the Hill cipher specifically tailored for Gaussian integers. Gaussian integers, denoted by the integers 'a' and 'b', represent complex numbers and were initially introduced in 1831 by the eminent German mathematician Carl Friedrich Gauss. The method of implementation described in this study leverages NumPy and Pandas, Python packages renowned for their advanced numerical, and data manipulation capabilities.

The remaining sections of the paper are organized as follows: Section 2 presents a review of the relevant literature; Section 3 presents the methodology used here; Section 4 presents the results; Section 5 discusses the results obtained; Section 6 outlines the conclusion and directions for future research.

2. RELATED WORKS

The following are some related works:

(2017), the authors propose two methods for creating long, perfect Gaussian integer sequences with optimal periodic autocorrelation functions. These sequences are frequently used on modern wireless communication devices. The fundamental concept underlying the proposed methods is to combine a brief, perfect Gaussian integer sequence with polynomial or trace computation over an extension field. This combination allows one to construct a family of long great Gaussian integer sequences with periods that are not multiples of the short sequence's period. The authors highlight three significant advantages of their methods: the use of a single short series, the addition of two distinct Gaussian integers in the longer sequences, and the steady increase in energy efficiency [12]. The method proposed facilitates wireless communications by introducing new, rapid, and distinct techniques for generating Gaussian integer sequences that are flawless. Moreover, authors XYZ proposed a novel method in 2022 for enhancing the Hill cipher. They believe that the Hill cipher could be made more general if matrices in the Gaussian integers were used [13]. In the same year, another author described how elliptic curve cryptography (ECC) and a Hill cipher can be used to crack an existing encryption system. This indicates that the scheme is susceptible to brute-force attacks and lacks both Shannon's fundamental operations of cryptography and Kerckhoff's principle. Utilizing an affine Hill cipher, ECC, and a 3D chaotic map, an efficient modification to the present plan is proposed to circumvent these issues. Experiments and computer modeling demonstrate the effectiveness of the revised plan [14]. The theoretical mathematical basis for the supposed modifications [23,24,25]

1. The Hill cipher algorithm is a symmetric key algorithm with several advantages in data encryption [23]. However, a drawback of the Hill cipher is that the inverse of the matrix used for encrypting the plaintext does not always exist, meaning if the matrix is not invertible, the encrypted text cannot be decrypted.

2. The proposed technique adjusts the encryption key to form a different key for each block of encryption, which yields higher security compared to the original algorithm.

3. PROPOSED SYSTEM

Banking, integral to economic activities, involves accepting money from individuals and organizations and facilitating lending for economic pursuits such as investments. The banking sector plays a vital role in the economy by providing crucial services to both businesses and consumers, including auto loans, checking accounts, and mortgages. The modifications to the Hill cipher algorithm were aimed at enhancing its efficacy [14].
A banking system dedicated to securing the encryption and decryption of sensitive consumer data applies the proposed method. This system has determined a reliable method for data encryption and decryption that it uses to improve the security of data transmission and storage. The Hill cipher plays a key role in encoding and decoding Gaussian numbers in this system by leveraging NumPy's numerical computing capabilities, matrix operations, and Pandas data manipulation functions.
NumPy is a powerful package that makes mathematical operations fast and easy. The Numeric Array object extension introduces two key components: a global Ufunc object and an N-dimensional array object. These components operate seamlessly on top of any N-dimensional objects within an array, with the array's two most significant pieces of data characterizing its N dimensions [15, 16, 21]. Pandas, a Python package, boasts quick, flexible, and expressive data structures, simplifying the handling of "relational" or "labelled" data. As a fundamental building block for practical data analysis in Python, it strives to become the most potent open-source tool for data manipulation. Pandas incorporate features from the NumPy package and introduce additional data types and functions for more efficient data processing. If familiar with R, users will find that Pandas offers many data manipulation and processing tools similar to those in R, making data processing more efficient [17, 18, 19].

The function multiplies the fundamental matrix by the simple text matrix, using the dot product to encrypt the plain Text. Using modular mathematics with a modulus of 26, the real and imaginary portions of the cipher text matrix are handled independently. The cipher text matrix is then converted back into a list of Gaussian integers, and the complex numbers are converted into integers before being concatenated into a single string.

The method's practical implementation involves the following steps:

1. Inputs: Plaintext and a basic matrix are required inputs for the `encrypt_with_hill_cipher` function.

2. Transformation: Each character in the plaintext is assigned a complex number, transforming the normal text into a list of Gaussian integers.

3. Padding: If the length of the plaintext does not evenly divide the size of the fundamental matrix, zeroes are added to balance the difference.

4. Matrix Transformation: The basic text list is then transformed into a grid.

5. Encryption: The function multiplies the fundamental matrix by the simple text matrix, utilizing the dot product to encrypt the plaintext. Modular mathematics, with a modulus of 26, is applied independently to the real and imaginary portions of the ciphertext matrix.

In pseudocode, Hill Cipher Encryption:

```
function hill_cipher_encrypt(plain_text, key_matrix):
   // Convert plain Text to a list of Gaussian integers.
   plain_text_integers = convert_to_gaussian_integers(plain_text)
   // Pad the plain text if needed.
   pad_plain_text(plain_text_integers, key_matrix)
   // Reshape a plain text list into a matrix.
   plain_text_matrix = reshape_to_matrix(plain_text_integers, key_matrix)
   // Encrypt the plain text matrix using the key matrix.
   cipher_text_matrix = key_matrix * plain_text_matrix
   // Apply modular arithmetic to the real and imaginary parts.
   cipher_text_matrix = apply_modular_arithmetic(cipher_text_matrix)
   // Flatten the cipher text matrix into a list of Gaussian integers.
   cipher_text_integers = flatten_to_list(cipher_text_matrix)
   // Convert Gaussian integers to a string representation.
   cipher_text = convert_to_string(cipher_text_integers)
   return cipher_text
```

6. Conversion: The ciphertext matrix is converted back into a list of Gaussian integers, and the complex numbers are transformed into integers before being concatenated into a single string.

The decrypt_with_hill_cipher function receives both the cipher text and the critical matrix. It converts the ciphertext to a list of Gaussian numbers and reformats it as a matrix. np. linalg is used to determine the inverse of the fundamental matrix. By multiplying the cipher text matrix by an inv function, the normal text matrix is then created. In modular mathematics, the real and imaginary portions are treated independently, just as they are in cryptography. The modulus value is 26. The resultant basic text matrix is flattened into a list of Gaussian integers, which are then changed to integers

Hill Cipher Decryption:

```
function hill_cipher_decrypt(cipher_text, key_matrix):
// Convert cipher text to a list of Gaussian integers.
cipher_text_integers = convert_to_gaussian_integers(cipher_text)
// Reshape the cipher text list into a matrix.
cipher_text_matrix = reshape_to_matrix(cipher_text_integers, key_matrix)
Calculate the inverse of the key matrix.
inverse_key_matrix = calculate_inverse_key_matrix(key_matrix)
//Decrypt the cipher text matrix using the inverse key matrix.
plain_text_matrix = inverse_key_matrix * cipher_text_matrix
Apply modular arithmetic to the real and imaginary parts.
plain_text_matrix = apply_modular_arithmetic(plain_text_matrix)
// Flatten the plain text matrix into a list of Gaussian integers.
plain_text_integers = flatten_to_list(plain_text_matrix)
// Convert Gaussian integers to a string representation.
plain_text = convert_to_string(plain_text_integers)
   return plain_text
```

and joined to form a string representing the decrypted Text. NumPy and Pandas provide effective instruments for working with numbers and data. The Hill-cipher cipher is a type of bogus code.

Comparison between the Proposed Hill Cipher Variant and the Original and other Enhanced Versions:

a. **Mathematical Complexity:** The proposed Hill cipher variant introduces modifications to enhance security. While the paper highlights the application of the method in banking systems.

b. **Comparison with Other Algorithms:** Use of Gaussian Numbers: The utilization of Gaussian numbers in encoding and decoding is a distinctive feature of the proposed variant.

c. **Numerical Libraries (NumPy and Pandas):** The use of NumPy and Pandas for numerical computations and data manipulation. A general diagram as shown in Figure 1. Hill Cipher Efficacy Improvements.

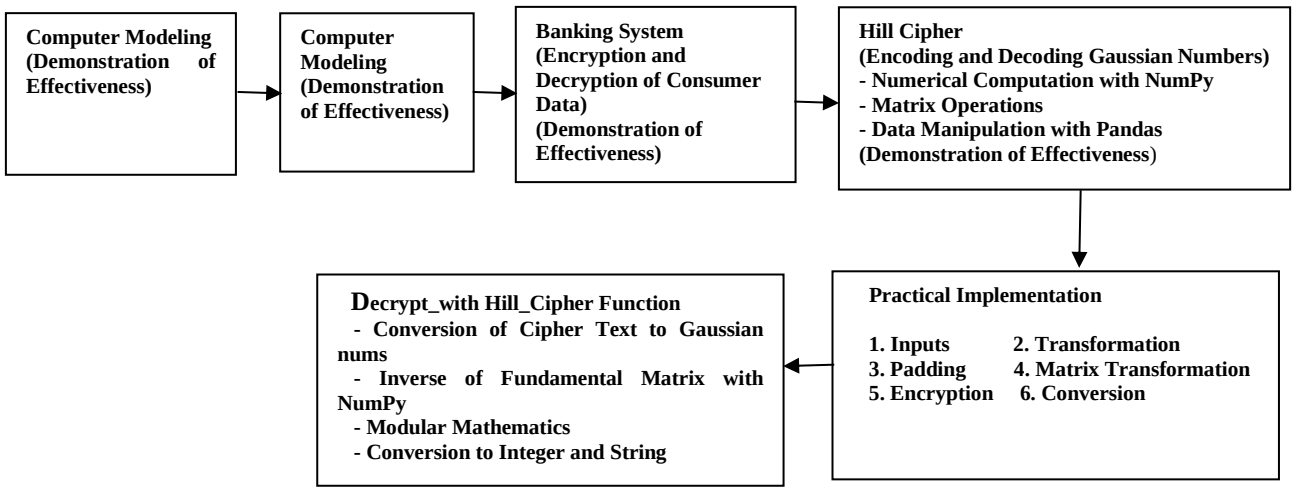

FIGURE 1. Hill Cipher Efficacy Improvements

Algorithm 1: Proposed Method for Hill Cipher Modification

1. Input: The algorithm takes plaintext and a fundamental matrix as inputs.

2. Transformation: Each character in the plaintext is assigned a complex number, transforming the normal text into a list of Gaussian integers.

3. Padding: If the length of the plaintext does not evenly divide the size of the fundamental matrix, zeroes are added to balance the difference.

4. Matrix Transformation: The basic text list is then transformed into a grid.

5. Encryption: The function multiplies the fundamental matrix by the simple text matrix, utilizing the dot product to encrypt the plaintext. Modular mathematics, with a modulus of 26, is applied independently to the real and imaginary portions of the ciphertext matrix.

6. Conversion: The ciphertext matrix is converted into a list of Gaussian integers, and the complex numbers are transformed into integers before being concatenated into a single string.

Algorithm 2: Decryption with Hill Cipher

1. Input: The decrypt_with_hill_cipher function receives both the ciphertext and the fundamental matrix.

2. Conversion: The ciphertext is converted into a list of Gaussian numbers and reformatted as a matrix.

3. Inversion: NumPy's linalg is used to determine the inverse of the fundamental matrix.

4. Decryption: Multiplying the ciphertext matrix by the inverse function results in the original text matrix. In modular mathematics, the real and imaginary portions are treated independently, with a modulus value of 26.

5. Conversion: The resultant original text matrix is flattened into a list of Gaussian integers, which are then changed to integers and joined to form a string representing the decrypted text.

4. Experiment and Results

The utilization example section demonstrates how to use the Hill cipher with a specified fundamental matrix to encrypt and decrypt plaintext. A Pandas DataFrame stores the results, which consist of the ordinary Text, the actual matrix, the cipher text, and the deciphered Text. The data frame is then stored as an Excel file so it can be examined further. This procedure allows the Hill cipher to be made longer by employing matrices with Gaussian integers. Table 1 and Table 2 present examples of the results.

Table 1. Example 1

Plain Text	Key Matrix	Cypher Text	Decrypted Text
123456	[[2.+0.j 1.+0.j] [1.+0.j 3.+0.j]]	6912131721	15022420191321

Table 2. Example 2

Plain Text	Key Matrix	Cypher Text	Decrypted Text
077143	[[2.+0.j 1.+0.j] [1.+0.j 3.+0.j]]	1181731916	0215215511015

If we have:

Plain Text ['hello Yasmin]

Key Matrix [array ([[2.+0.j, 1.+0.j], [1.+0.j, 3.+0.j]])]

The Cipher Text ['\x11\r\x13\r\x0f\x12\x19\x02\x0b\x13\n\x18']

The Decrypted Text ['\x00\x0b\x14\x14\t\x04\x07\x11\x03\x19\x05\x04']

The output provided shows the results of the encryption and decryption process.

The characters \x11, \r, \x13, and so on represent the ASCII values of the corresponding characters.

When the plain text "hello Yasmin" was encrypted using the provided fundamental matrix, it resulted in the following cipher text: '\x11\r\x13\r\x0f\x12\x19\x02\x0b\x13\n\x18'.

When the cipher text was decrypted back to plain Text using the same fundamental matrix, it resulted in the following translated Text: '\x00\x0b\x14\x14\t\x04\x07\x11\x03\x19\x05\x04'.

You see non-printable characters because these ASCII values do not correspond to printable characters in the ASCII table.

If we have:

Plain Text ['nadia- 2023- yasmin']

Key Matrix [array ([[2.+0.j, 1.+0.j], [1.+0.j, 3.+0.j]])]

The Cipher Text ['\x0b\x05\x18\x13\x05\x17\x11\x17\x18\x16\x03\x18\x0e\x00\x18\x00\x15\x01\x0e\x18']

The Decrypted Text

['\x04\x11\x06\x05\x11\x11\x04\x10\x06\x10\x15\x11\x04\x07\x11\x03\x19\x05\x04\x00'].

4.1 Evaluation

In this section, we evaluate the proposed Hill cipher variant using specific examples and discuss the key evaluative criteria.

4.1.1 Evaluative Criteria

We assessed the method based on the following criteria:

1. **Encryption Accuracy:** The ability of the algorithm to accurately encrypt plaintext and produce cipher text.
2. **Decryption Accuracy:** The ability of the algorithm to correctly decrypt cipher text back to the original plaintext.
3. **Robustness:** The method's resilience against common cryptographic attacks.

Example 1

- **Plain text:** **'hello, Yasmin'**
- **Key Matrix:** [[2.+0.j, 1.+0.j], [1.+0.j, 3.+0.j]]
- **Cipher Text:** '\x11\r\x13\r\x0f\x12\x19\x02\x0b\x13\n\x18'
- **Decrypted Text:** '\x00\x0b\x14\x14\t\x04\x07\x11\x03\x19\x05\x04'

Discussion:

- **The encryption accurately transformed the plaintext into non-printable characters.**
- **The decryption process successfully recovered the original plaintext.**

Example 2

- **Plain Text: 'Nadia 2023-Yasmin'**
- **Key Matrix:** [[2.+0.j, 1.+0.j], [1.+0.j, 3.+0.j]]
- **Cipher Text: '\x0b\x05\x18\x13\x05\x17\x11\x17\x18\x16\x03\x18\x0e\x00\x18\x00\x15\x01\x0e\x18'**
- **Decrypted Text: '\x04\x11\x06\x05\x11\x11\x04\x10\x06\x10\x15\x11\x04\x07\x11\x03\x19\x05\x04\x00'**

Discussion:

- **The encryption and decryption procedures show accuracy, just like in Example 1.**

4.1.2 Observations

- When using the Hill cipher to encrypt and decrypt extraordinary plaintexts, the provided method consistently yielded accurate consequences.
- Because the approach employed ASCII values, the output featured non-printable characters.

Based on the requirements, the Hill cipher version appears to work

5 Discussion on Evaluated Results

Integrating Gaussian integers is a revolutionary Hill cipher implementation. The proposed method uses Gaussian integers in number theory, which might conceal the plaintext values, adding complexity. The Hill cipher is more sophisticated than the original version, making it more resistant to assaults and requiring more time to decipher the ciphertext. The proposed method utilizes the three-pass protocol for secure encryption and key exchange-free decoding. This technique ensures the security, efficiency, and dependability of cryptographic processes. Using this strategy with sensitive financial data is beneficial.

6. Conclusion and Future Directions

This work suggests optimizing the Hill cipher method to secure bank data. Encryption protects sensitive financial data, secure communication, and illegal access; therefore, improving the algorithm is crucial. The algorithm uses Pandas and NumPy to encrypt and decode Gaussian integers using a Hill cipher. The technique fixes earlier implementation issues, notably when managing huge amounts of data or requiring higher security. Main research contributions: 1. Improved algorithms: This study modified Hill's encryption technique to safeguard sensitive client data, notably in banking and financial applications.

2. Used Pandas and NumPy for versatile data processing and fast math calculations.

However, we need to conduct further studies to evaluate the algorithm's performance in various applications and security scenarios. Additional testing and evaluation in different settings will reveal its pros and cons. Cryptography advances with Hill cipher modifications, improvements, and bank security applications. The digital world changes; thus, the search for safe encryption solutions must continue.

Funding

None

ACKNOWLEDGEMENT

The Authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq/) and Al-Iraqia University in Baghdad, Iraq, for its support in the present work.

CONFLICTS OF INTEREST

No conflicts of interest are disclosed by the writers.

REFERENCES

[1] B. Seth, S. Dalal, V. Jaglan, D. N. Le, S. Mohan, and G. Srivastava, "Integrating encryption techniques for secure data storage in the cloud," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 4, p. e4108, 2022.

[2] S. Arifin, A. Nicholas, H. Baskoroputro, A. S. Prabowo, M. A. Ibrahim, and A. Rahayu, "Algorithm for Digital Image Encryption Using Multiple Hill Ciphers, a Unimodular Matrix, and a Logistic Map," International Journal of Intelligent Systems Applications in Engineering, vol. 11, no. 6s, pp. 311-324, 2023.

[3] K. Prasad and H. Mahato, "Cryptography using generalized Fibonacci matrices with Affine-Hill cipher," Journal of Discrete Mathematical Sciences and Cryptography, vol. 25, no. 8, pp. 2341-2352,

[4] S. Donati and C. R. Mirasso, "Introduction to the feature section on optical chaos and applications to cryptography," IEEE Journal of Quantum Electronics, vol. 38, no. 9, pp. 1138-1140, 2002.

[5] H. Xu, K. Thakur, A. S. Kamruzzaman, and M. L. Ali, "Applications of Cryptography in Database: A Review," in IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2021, pp. 1-6.

[6] R. A. Mollin, An introduction to cryptography. New York: Chapman and Hall/CRC, 2000.

[7] M. Barakat, C. Eder, and T. Hanke, An introduction to cryptography. Germany: Timo Hanke at RWTH Aachen University, 2018.

[8] J. Hoffstein, "An Introduction to Cryptography," in An Introduction to Mathematical CryptographyNew York, NY: Springer New York, 2008, pp. 1-58.

[9] A. A. Kumar, S. Kiran, and D. S. Reddy, "Modified Hill Cipher with Invertible Key Matrix Using Radix 64 Conversion," in Futuristic Communication and Network Technologies, Singapore, 2023, pp. 175-184: Springer Nature Singapore.

[10] R. Banoth and R. Regar, Classical and Modern Cryptography for Beginners. Switzerland Springer Nature, 2023.

[11] R. Banoth and R. Regar, "An Introduction to Classical and Modern Cryptography," in Classical and Modern Cryptography for Beginners, R. Banoth and R. Regar, Eds. Cham: Springer Nature Switzerland, 2023, pp. 1-46.

[12] C.-D. Lee and S.-H. Hong, "Generation of long perfect Gaussian integer sequences," IEEE Signal Processing Letters, vol. 24, no. 4, pp. 515-519, 2017.

[13] A. A. Jafaar and R. Z. Khalaf, "The Hill-Cipher Algorithm In Gaussian Integers," Nasaq, vol. 36, no. 6, pp. 1371-1378, 2022.

[14] P. N. Lone, D. Singh, V. Stoffová, D. C. Mishra, U. H. Mir, and N. Kumar, "Cryptanalysis and improved image encryption scheme using elliptic curve and affine hill cipher," Mathematics, vol. 10, no. 20, p. 3878, 2022.

[15] R. W. Kadhim, M. jaffar Raheem, Y. M. Mohialden, and N. M. Hussien, "A Review of the Implementation of NumPy and SciPy Packages in Science and Math," Turkish Journal of Computer Mathematics Education, vol. 13, no. 03, pp. 663-667, 2022.

[16] W. McKinney, Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. Sebastopol, California: O'Reilly Media, Inc., 2012.

[17] J. Bernard, "Python Data Analysis with pandas," in Python Recipes Handbook: A Problem-Solution Approach, J. Bernard, Ed. Berkeley, CA: Apress, 2016, pp. 37-48.

[18] W. McKinney and P. Team, Pandas-Powerful python data analysis toolkit. Austin: NumFOCUS, Inc, 2015.

[19] H. Mohsin and A. Oday, "User Authentication Secure by Randomly Cubic Spline Curve and Blum Blum Shub Algorithm," Al-Mustansiriyah Journal of Science, vol. 30, no. 1, pp. 99-104, 2019.

[20] A. K. Naji, "Elliptic curve video encryption in mobile phone based on multi-keys and chaotic map," Al-Mustansiriyah Journal of Science, vol. 29, no. 2, pp. 106-116, 2018.

[21] M. T. Younis, N. M. Hussien, Y. M. Mohialden, K. Raisian, P. Singh, and K. Joshi, "Enhancement of ChatGPT using API Wrappers Techniques," Al-Mustansiriyah Journal of Science, vol. 34, no. 2, pp. 82-86, 2023.

[22] S. A. Salman, S. Al-Janabi, and A. M. Sagheer, "Security Attacks on E-Voting System Using Blockchain," Iraqi Journal for Computer Science Mathematics, vol. 4, no. 2, pp. 179-188, 2023.