Vol. 6 No. 3, 2025, pp. 188-195 **Jurnal Pepadun**

Hybrid Hill Cipher ASCII 256 and RSA Cipher in Securing Messages

[1]Sandi Saputra, *[2]Fitriani, [3]Ahmad Faisol, [4]Wamiliana, and [5]Siti Laelatul Chasanah

Department of Mathematics, Universitas Lampung, Bandar Lampung, Indonesia, 35145
e-mail: [1]sandisaputraa26@gmail.com, ***[2]fitriani.1984@fmipa.unila.ac.id,** [3]ahmadfaisol@fmipa.unila.ac.id,
[3]ahmadfaisol@fmipa.unila.ac.id, [4]wamiliana.1963@fmipa.unila.ac.id, [5]siti.chasanah@fmipa.unila.ac.id

Abstract *- Secure communication requires encryption methods that strike a balance between efficiency and key protection. This study proposes a hybrid cryptosystem integrating the Hill Cipher and the RSA Cipher. The Hill Cipher, based on modular matrix multiplication, achieves fast encryption but lacks secure key distribution, whereas RSA provides robust asymmetric key management at a higher computational cost. By encrypting the Hill Cipher key with RSA, the proposed method strengthens resistance to brute-force and key compromise attacks. A Python-based implementation was developed and tested on messages of varying lengths. Experimental results show that the hybrid system maintains the efficiency of the Hill Cipher while significantly improving key security, with time complexity increasing modestly due to RSA key operations. Two scenarios were analyzed depending on whether the message length is divisible by the key matrix size, confirming the scheme's flexibility. This approach demonstrates a practical and secure solution for protecting digital communication against cryptanalysis.*

Keywords**:** *Cryptography; Symmetric Key; Asymmetric Key; RSA Cipher; Hill Cipher; Encryption.*

1. INTRODUCTION

Cryptography is the science and art of securing information by transforming readable data into an unintelligible form [1]. This process, known as encryption, requires an algorithm and a secret key, while decryption reverses the process to recover the original message. Cryptographic methods can be broadly categorized into symmetric and asymmetric systems [2][3]. Symmetric algorithms, such as the Hill Cipher, use the same key for encryption and decryption, offering speed and efficiency [4][5]. Hill Cipher, based on modular matrix multiplication, provides a large key space since the key is an invertible $n \times n$ matrix modulo p [6][7]. However, like other symmetric algorithms, it suffers from weaknesses in key distribution and overall security [5].

To address these weaknesses, asymmetric cryptography emerged, with Rivest Shamir Adleman (RSA) being the most widely used system [3][12]. Unlike symmetric methods, RSA uses a public–private key pair that ensures secure key exchange over untrusted networks [13]. While RSA provides stronger key management, it requires higher computational resources and can be less efficient for encrypting large amounts of data [12][13]. Thus, combining the efficiency of Hill Cipher with the robust security of RSA has become an attractive research direction [14][15].

Hybrid cryptographic methods that integrate symmetric and asymmetric techniques have been widely studied in the last decade. Jamaludin [2] demonstrated the feasibility of combining Hill Cipher with RSA using a 2×2 matrix key, while Santoso [6] extended this work to a 3×3 matrix with RSA-512. Other studies have explored expanding character sets beyond alphabets to ASCII [8][9][10], introducing dynamic and randomized key generation [20][22], or even using mathematical constructs such as Fibonacci sequences to enhance matrix security [18][24]. These developments illustrate a consistent effort to address the limitations of early Hill–RSA hybrids, yet they remain constrained by scalability, limited character handling, and practical deployment challenges.

Recent works highlight the growing need for hybrid cryptosystems tailored to modern digital communication [16][19][21]. Applications such as cloud data storage, Internet of Things (IoT) devices, and secure email exchange require cryptographic systems that balance computational efficiency with high resistance to brute-

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.284

force and cryptanalytic attacks [23][25]. Hybrid approaches provide a natural fit in these contexts, as symmetric algorithms handle bulk data efficiently, while asymmetric methods ensure secure key management. Nevertheless, ensuring robustness against emerging threats demands further innovation in algorithm design and implementation [14][17].

This research advances the state of the art by generalizing the Hill Cipher key matrix to arbitrary $n \times n$ dimensions and extending the encryption process to cover all 256 ASCII characters. These modifications significantly enlarge the key space, increase resistance to brute-force and cryptanalytic attacks, and broaden applicability to diverse digital messages beyond simple alphabets [19][23]. Furthermore, a Python-based implementation was developed to evaluate the practicality and efficiency of the proposed hybrid system. The novelty of this research lies in unifying generalized key matrix design with full ASCII support, offering a more secure and flexible cryptographic solution applicable to real-world systems such as secure email, IoT devices, and cloud-based data transfer [21][25].

2. RESEARCH METHODOLOGY

In this section, several theoretical bases that support research and program stages using the Python programming language will be discussed.

2.1. *Hill Cipher*

Hill cipher, which is a polyalphabetic cipher, can be categorized as a block cipher, because the text to be encrypted is divided into blocks of a certain size. Each character in one block will influence other characters in the encryption and decryption process [26]. The basis of the Hill Cipher technique is modulo arithmetic over matrices. In its application, Hill Cipher uses matrix multiplication techniques and matrix inverse techniques. The key in Hill Cipher is a matrix $n \times n$ where n is the block size [28]. If the key is called K, then K is as follows:

$$K = \begin{bmatrix} k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots \\ k_{n1} & \cdots & k_{nn} \end{bmatrix} \tag{1}$$

The K matrix that is the key must be an invertible matrix. Because, K^{-1} which is the inverse of the key matrix is used to perform decryption [27]. The encryption process in Hill Cipher is carried out per plaintext block. The size of the block must be the same as the size of the key matrix [29]. Mathematically, the encryption process in Hill Cipher is:

$$C = K.P \tag{2}$$

with:
C = ciphertext,
K = key,
P = plaintext.

The decryption process for Hill Cipher is basically the same as the encryption process. However, the key matrix must be reversed first. Mathematically, the decryption process in Hill Cipher is as follows [30]:

$$P = K^{-1}C \tag{3}$$

2.2. *RSA Cipher*

The RSA algorithm consists of a public key and a private key where the public key can be known by everyone while the private key is only known by the data owner. The encryption process uses the public key and the decryption process uses the data owner’s private key [26]. This algorithm consists of three processes, namely the key formation process, the encryption process and the decryption process. The following is the key formation process in the RSA cryptographic algorithm:

1) Select two prime numbers denoted as p and q (value of $p \neq q$).
2) Calculate the value of $n = p.q$ ($p \neq q$, because if $p = q$, then the value of $n = p^2$ so that the value of p can be obtained by taking the square root of n).
3) Calculate the value of $\psi(n) = (p-1).(q-1)$.
4) Select a public key e that is prime relative to $\psi(n)$.
5) Generate the private key with the equation $e.d \equiv 1\big(mod\ \psi(n)\big)$ where $1 < d < \psi(n)$. Note that the equation $e.d \equiv 1\big(mod\ \psi(n)\big)$ is equivalent to $e.d = 1 + k.\psi(n)$, so to find the value of d it can be calculated by $d = \frac{1+k.\psi(n)}{e}$ [27].

The following is the encryption process in the RSA cryptographic algorithm:

1) Public key (e, n).
2) Select plaintext m and change the contents of message m to a message with the value ASCII.
3) Cut the message into message blocks $m_1, m_2, m_3, \ldots$ with the value of each block being $0 \leq m \leq n - 1$.
4) Each block m is calculated using the formula $c_i = m_i^e\ mod\ n$.
5) Arrange the value c resulting from encryption in the order $c_1, c_2, c_3, \ldots, c_n$ so that ciphertext is obtained from message m [27].

The following is the decryption process in the RSA cryptographic algorithm:

1) Retrieve the message ciphertext that has been received.
2) Then take the secret key (d, n).
3) Cut the message into message blocks $c_1, c_2, c_3, \ldots$ with the value of each block being $0 \leq c \leq n - 1$.
4) Calculate $m_i = c_i^d\ mod\ n$.
5) Arrange the decrypted m values with the order $m_1, m_2, m_3, \ldots, m_n$ so that you get plaintext (original message) from the received ciphertext [27].

3. RESULTS AND DISCUSSION

In this section, an example of encryption and decryption of a message consisting of several characters representing the 256 ASCII characters used is given.

3.1. *Encryption-Decryption Hybrid Hill Cipher and RSA Cipher*

A message "üýþÿòóôõÞßâáÊÉÎÐ"#$& bAECcKYk{|}~" having a length of 32 characters will be encrypted and decrypted using the Hybrid Hill Cipher and RSA Cipher. This process will be divided into two cases based on the length of the selected key matrix.

3.1.1. Case 1: If the selected key length divides the length of the message.
For example, the key matrix used is a matrix of size 4 × 4 as follows:

$$K = \begin{bmatrix} 1 & 2 & 0 & 2 \\ 2 & 1 & 2 & 1 \\ 0 & 1 & 2 & 0 \\ 2 & 2 & 1 & 1 \end{bmatrix}.$$

By using cofactor expansion, the determinant of the K matrix is 9. The determinant of the K matrix is relatively prime to 256, therefore the matrix used is an invertible matrix. Next, the message is divided

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.284

into blocks of size 4×1 and then each block is multiplied by the K matrix from the left. The results obtained are:

$$c_1 = \begin{bmatrix} 244 \\ 240 \\ 249 \\ 239 \end{bmatrix};\ c_2 = \begin{bmatrix} 194 \\ 180 \\ 219 \\ 179 \end{bmatrix};\ c_3 = \begin{bmatrix} 94 \\ 64 \\ 163 \\ 61 \end{bmatrix};\ c_4 = \begin{bmatrix} 252 \\ 201 \\ 206 \\ 208 \end{bmatrix};$$

$$c_5 = \begin{bmatrix} 180 \\ 213 \\ 107 \\ 212 \end{bmatrix};\ c_6 = \begin{bmatrix} 106 \\ 210 \\ 203 \\ 206 \end{bmatrix};\ c_7 = \begin{bmatrix} 207 \\ 46 \\ 253 \\ 32 \end{bmatrix};\ c_8 = \begin{bmatrix} 111 \\ 234 \\ 118 \\ 233 \end{bmatrix}.$$

Then the next step is to build a public key using $p = 193$ and $q = 251$. The respective public and private keys are obtained (13121, 48443) and (9281, 48443). By using the public key, the encryption results are:

$\{1582, 28236, 38858, 44973, 13125, 42061, 39626, 45534, 35220, 20329, 26868, 32257, 35643, 2131, 33682, 44779, 42061, 11600, 47009, 9316, 25389, 16482, 43733, 46719, 40737, 12935, 32677, 19861, 20094, 41079, 18727, 20344\}$.

Next, the encryption results will be decrypted using the private key. Then the process continues with Hill Cipher decryption by dividing the decryption results into blocks of size 4×1 and then looking for the inverse of the key matrix used. Then convert the value of each block into an ASCII table then combine it so that the decryption result is "üýþÿòóôõÞßâáÊÉÎÐ"#$& bAECcKYk{|}~".

3.1.2. **Case 2:** If the selected key length doesn't divide the length of the message.

In general, if the length of the selected key matrix does not divide by the length of the message, then the message needs to be added with null characters until the message is divisible by the length of the key matrix. Here is an example of a case where the length of the selected key matrix does not divide evenly into the length of the message. For example, the key matrix used is

$$K = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}.$$

Because 32 is not divisible by 3, the message needs to add one null character. Furthermore, by using the Sarrus method, it is obtained that the determinant of the K matrix is 3 which is relatively prime to 256. Then, divide the message into blocks of size 3×1 and convert the characters to ASCII values. Then encrypt it using the Hill Cipher encryption algorithm. Next, construct each public and private using $p = 157$ and $q = 197$, the results obtained are (10739, 30929) and (17291, 30929) respectively. By using the public key, the encryption results are:

{23399, 10186, 2335, 18583, 10621, 12302, 3575, 10286, 71, 20354, 30800, 9990, 25989, 30703, 25332, 17831, 10042, 16164, 23031, 8286, 17239, 18085, 11581, 18761, 4096, 28295, 29856, 28072, 10153, 11612, 23390, 23717, 17239}.

Next, decryption is carried out using the public key and inverse matrix as in case 1. Because the length of the decryption result is 33, the decryption message needs to be subtracted with null characters until the length of the decryption message is the same as the length of the initial message.

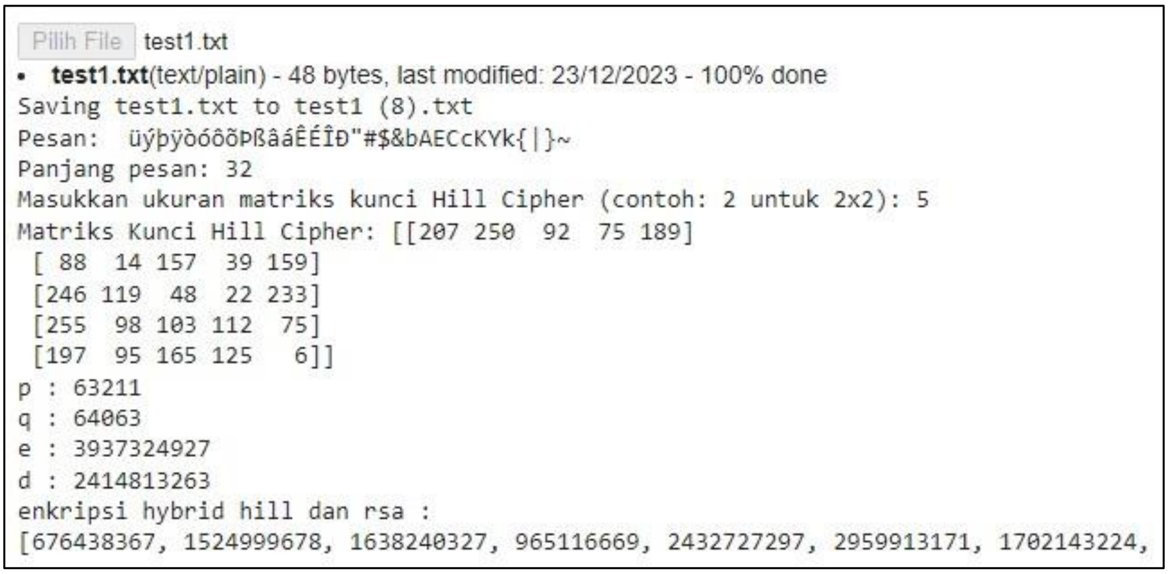
From the two cases that have been discussed, it can be concluded that the larger the size of the key matrix and the longer the bits used, the greater the number of possible keys, therefore the encrypted message is much more secure. This is because the number of all possible keys for a matrix of size $n \times n$ is 256^{n^2} and the number of possible keys for RSA Cipher is 2^L . So, the total possible keys of Hybrid Hill Cipher and RSA Cipher are $256^{n^2} \times 2^L$ where n is the size of the key matrix and L is the bits length of the RSA key used.

3.2. *Hybrid Hill Cipher and RSA Cipher program*

The hybrid Hill Cipher and RSA Cipher programs were developed using the Python programming language, chosen for its flexibility, readability, and wide range of libraries that support mathematical and cryptographic operations. Python provides efficient handling of matrix computations and modular arithmetic, which are essential in implementing the Hill Cipher, as well as large integer operations that are critical for RSA. By leveraging Python, the implementation not only becomes more accessible for researchers and developers but also ensures that the system can be easily tested, modified, and extended for future improvements.

In the Hill Cipher component, the program is designed to automatically generate and verify an invertible key matrix based on the size input by the user. The invertibility of the matrix modulo p is a crucial requirement, as a non-invertible matrix would make the decryption process impossible. Once the matrix is validated, the program proceeds with the encryption process, adjusting its operations according to the specific case division explained earlier, such as handling messages whose lengths are not divisible by the matrix size. In such cases, padding with a special character ensures that the encryption can proceed without data loss or misalignment.

Meanwhile, the RSA component of the program focuses on generating secure public and private keys. The program randomly selects two large prime numbers, p and q, based on the bit length specified by the user. This process ensures the creation of a valid key pair, enabling secure key management and addressing one of the major weaknesses of purely symmetric algorithms such as the Hill Cipher.



```
test1.txt
•  test1.txt(text/plain) - 48 bytes, last modified: 23/12/2023 - 100% done
Saving test1.txt to test1 (8).txt
Pesan:  üýþÿòóôõÞßâáÊÉÎÐ"#$&bAECcKYk{|}~
Panjang pesan: 32
Masukkan ukuran matriks kunci Hill Cipher (contoh: 2 untuk 2x2): 5
Matriks Kunci Hill Cipher: [[207 250  92  75 189]
 [ 88  14 157  39 159]
 [246 119  48  22 233]
 [255  98 103 112  75]
 [197  95 165 125   6]]
p : 63211
q : 64063
e : 3937324927
d : 2414813263
enkripsi hybrid hill dan rsa :
[676438367, 1524999678, 1638240327, 965116669, 2432727297, 2959913171, 1702143224,
```

Figure 1. Encryption results for matrix 5 × 5.

To illustrate the encryption process, the program includes an example using a 5 × 5 key matrix. This example demonstrates how plaintext characters are converted into numerical values, arranged into vector blocks, and then multiplied by the key matrix modulo p to produce the ciphertext. Following the Hill Cipher stage, the RSA encryption is applied to secure the key exchange process, ensuring that the hybrid system can maintain both efficiency in message encryption and robustness in key distribution. A visual representation of this process is provided in the figure below, offering a clearer understanding of how the hybrid system operates in practice.

Overall, the integration of Hill Cipher and RSA within the Python-based program illustrates the balance between theoretical cryptographic design and practical implementation. The program not only verifies the feasibility of the hybrid model but also serves as a foundation for further experimentation, optimization, and adaptation to real-world applications such as secure communication, cloud data protection, and lightweight IoT encryption solutions. An illustration of the encryption process applied to a 5×5 matrix is provided in Figure 1.

4. CONCLUSIONS

Based on the research that has been conducted, it can be concluded that hybrid cryptography combining Hill Cipher and RSA Cipher can be effectively implemented by modifying the size of the key matrix, extending the number of characters to the full 256 ASCII set, and adding the ’nul’ character when the message length is not divisible by the matrix size. These modifications not only ensure proper handling of message encoding but also significantly expand the key space, thereby enhancing security and making the encryption more resistant to brute-force and cryptanalytic attacks. In addition, the integration with RSA strengthens key management, addressing the main weakness of symmetric cryptography.

Furthermore, the Python-based implementation developed in this study demonstrates the practical applicability of the proposed hybrid system. It provides a tool that can be adapted for securing digital communication in real-world contexts, such as text messaging, email, and data transmission in online platforms. Thus, the proposed method offers both theoretical improvements in encryption security and practical contributions to the development of efficient and reliable cryptographic applications.

Beyond its immediate implementation, the combination of generalized Hill Cipher and RSA also highlights a promising pathway for future hybrid cryptosystems. The use of an arbitrary $n \times n$ key matrix together with full ASCII support provides flexibility for adapting the algorithm to different communication protocols and data formats. This scalability makes the method suitable not only for academic exploration but also for integration into practical systems where diverse data types must be secured.

In addition, the results of this study indicate that the hybrid approach has potential for broader application in modern digital ecosystems. For instance, the proposed model can be adapted to secure Internet of Things (IoT) devices, where lightweight but secure algorithms are needed, or cloud storage platforms that demand both efficiency and strong encryption. With further optimization and testing, the system can also contribute to the development of secure messaging applications and data protection tools that balance usability with robust cryptographic strength.

Finally, while the present research demonstrates significant improvements, it also opens opportunities for further exploration. Future work may involve optimizing the algorithm for larger datasets, integrating advanced key generation techniques, or combining the hybrid system with emerging technologies such as quantum-resistant cryptography. Such directions would ensure that the proposed method remains relevant and effective in addressing the evolving challenges of digital security.

ACKNOWLEDGMENTS

The authors would like to thank the Department of Mathematics, FMIPA, Universitas Lampung, for the facilities and support provided during this research. Gratitude is also extended to the anonymous reviewers for their valuable comments that helped improve this manuscript.

LITERATURE

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Boston, MA, USA: Pearson, 2017.

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.284

[2] A. Jamaludin, “Kombinasi Hill Cipher dan RSA Cipher,” *Jurnal Ilmiah Teknologi Informasi*, vol. 12, no. 2, pp. 45-52, 2016.

[3] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 2018.

[4] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[5] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2020.

[6] A. Santoso, “Hybrid Cryptography Using 3×3 Hill Cipher and RSA-512,” *Proceedings of the International Conference on Information Technology and Security*, pp. 233-240, 2018.

[7] N. Kaur and R. Singh, “Enhanced Hill Cipher for Secure Communication,” *International Journal of Computer Applications*, vol. 117, no. 5, pp. 25-29, 2015.

[8] S. Sharma and M. Singh, “Hybrid Cryptographic Algorithm Based on RSA and Hill Cipher,” *International Journal of Computer Science and Information Security*, vol. 14, no. 8, pp. 145-151, 2016.

[9] A. Verma and R. Gupta, “Modified Hill Cipher Algorithm Using ASCII Values,” *International Journal of Computer Applications*, vol. 162, no. 3, pp. 12-17, 2017.

[10] M. Z. Hasan, “Hybrid Cryptography with Hill Cipher and RSA for Secure Messaging,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 2, pp. 677-684, 2019.

[11] S. Aljawarneh, “Design and Analysis of a Hybrid Hill Cipher-RSA Algorithm,” *International Journal of Network Security & Its Applications*, vol. 11, no. 3, pp. 13-21, 2019.

[12] P. P. Patil and S. R. Patil, “Hybrid Cryptosystem Using RSA and Symmetric Key Cryptography,” *International Journal of Computer Applications*, vol. 975, no. 8887, pp. 21-25, 2020.

[13] H. Li and Q. Wang, “A Hybrid Hill-RSA Cryptosystem for Secure Communication,” *Journal of Information Security and Applications*, vol. 54, p. 102566, 2020.

[14] M. A. Khan and F. A. Khan, “Hybrid Cryptography Model for Secure Data Transmission,” *International Journal of Computer Science and Network Security*, vol. 20, no. 6, pp. 1-7, 2020.

[15] A. H. Mohammed and R. A. Rashid, “Implementation of Hybrid Cryptography Algorithm Based on RSA and Symmetric Key,” *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 45-52, 2021.

[16] N. Singh and S. Kumar, “Enhanced Security Using Hill Cipher with RSA,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 3, pp. 1123-1127, 2020.

[17] H. Alenezi, “Hybrid Cryptography for Cloud Security Using RSA and Symmetric Encryption,” *International Journal of Computer Science and Network Security*, vol. 21, no. 4, pp. 110-117, 2021.

[18] A. K. Sharma and R. Kumar, “Performance Analysis of Hybrid RSA and Hill Cipher,” *Journal of Physics: Conference Series*, vol. 1969, no. 1, p. 012041, 2021.

[19] A. Rahman and S. Islam, “Hybrid Encryption Approach Using Hill Cipher and RSA for Data Security,” *International Journal of Computer Applications*, vol. 183, no. 23, pp. 18-24, 2021.

[20] P. Gupta and V. Sharma, “A Modified Hybrid Encryption Model Based on Hill Cipher and RSA,” *International Journal of Information Technology and Computer Science*, vol. 13, no. 5, pp. 10–17, 2021.

[21] L. Zhang, Y. Chen, and H. Wu, “Hybrid Hill-RSA Encryption Scheme for IoT Security,” *Sensors*, vol. 22, no. 3, p. 934, 2022.

© 2025 The Authors | doi: 10.23960/pepadun.v6i3.284

[22] A. Setiawan, "Hybrid Cryptosystem Based on Hill Cipher and RSA with Python Implementation," *Journal of Computer Science Research*, vol. 4, no. 2, pp. 55-63, 2022.

[23] R. Yadav and N. Singh, "Hybrid Cryptography Using Hill Cipher and RSA for Cloud Data Security," *International Journal of Computer Science Trends and Technology*, vol. 10, no. 4, pp. 36-42, 2022.

[24] M. Hossain and T. Rahman, "A Secure Hybrid Cryptography Approach Using Hill Cipher and RSA," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, pp. 451-458, 2022.

[25] Y. Liu, K. Zhao, and J. Zhang, "Improved Hybrid Hill-RSA Algorithm for Secure Data Communication," *IEEE Access*, vol. 11, pp. 13456-13467, 2023.

[26] A. N. Pratama, E. Pratama, and E. T. Wulandari, "Implementation of the Elgamal Algorithm and HILL code for Database Security", *Informatics Engineering Paper*, 2020.

[27] A. P. Wahyadyatmika, R. R. Isnanto, and M. Somantri, "Implementation of the RSA Cryptographic Algorithm in Electronic Mail (E-Mail)", *Transient*, vol. 3, no. 4, pp. 1-9, 2014.

[28] R. Makhomah, K. A. Santoso, and A. Kamsyakawuni, "Text Encoding Using a Combination of Hill Cipher and XOR Operations", *PRISMA: Proceedings of the National Mathematics Seminar*, 4, pp. 548-552, 2021.

[29] G. Krisnawanti, K. A. Santoso, and A. Kamsyakawuni, "Huffman Modification with Hill Cipher in Text Data Encoding", *PRISMA: Proceedings of the National Mathematics Seminar*, 4, pp. 534-539, 2021.

[30] F. S. Putra, and D. Ariyus, "Text Encryption and Decryption Using Hill Cipher With a 3×3 Order Matrix", *Jurti*, vol. 5, no. 1, pp. 17-22, 2021.