Paper Al-Jabar-[29Nov2021]b

by Samsul Arifin

Al-Jabar: Jurnal Pendidikan Matematika
Vol. xx, No. x, 20xx, Hal xx – xx
p-ISSN: 2086-5872, e-ISSN: 2540-7562

Unimodular Matrix and Bernoulli Map on Text Encryption Algorithm using Python

Samsul Arifin[1], Indra Bayu Muktyas[2], Puguh Wahyu Prasetyo[3,*], Abdul Aziz Abdillah[4]

[1]Bina Nusantara University, samsul.arifin@binus.ac.id
[2]Sekolah Tinggi Keguruan dan Ilmu Pendidikan Surya, recobayu@gmail.com
[3]Universitas Ahmad Dahlan, puguh.prasetyo@pmat.uad.ac.id
[4]Politeknik Negeri Jakarta, abdul.aziz.a@mesin.pnj.ac.id

Article Info

Submitted : dd:mm:yyyy
Revised : dd:mm:yyyy
Accepted : dd:mm:yyyy
Published : dd:mm:yyyy

*Correspondence:
puguh.prasetyo@pmat.uad.ac.id

Abstract

One of the encryption algorithms is the Hill Cipher. The square key matrix in the Hill Cipher method must have an inverse modulo. The unimodular matrix is one of the few matrices that must have an inverse. In the encryption process, a unimodular matrix can be utilized as a key. The goal of this research is to demonstrate that there is another approach to protect text message data. Symmetric cryptography is the sort of encryption utilized. A Bernoulli Map is used to create a unimodular matrix. To begin, we use an identity matrix to generate a unimodular matrix. The series of real values in (0,1) of the Bernoulli Map is translated to integers between 0 and 255. The numbers are then inserted into the unimodular matrix's top triangular entries. To acquire the full matrix as the key, we utilize Elementary Row Operations. The data is then encrypted using modulo matrix multiplication.

Key Words : Bernoulli Map; Hill cipher; Python; Unimodular Matrix

http://ejournal.radenintan.ac.id/index.php/al-jabar/index

Introduction

A computer network is a collection of computers connected together in order to share resources, communicate, and access data. The goal of a computer network is to fulfill its goals, and any component of the network can request and deliver services. The client is the person who requests and receives the service, whereas the server is the person who delivers and sends the service. A client-server system is an architecture that is utilized in practically all computer network applications. Securing computer networks or data necessitates the use of appropriate procedures. In cryptography, this is a common strategy. To safeguard data, cryptography may be employed in a variety of ways. The Hill Cipher algorithm is one of the most often utilized methods. The Hill Cipher is a symmetric key cryptographic technique with various data encryption benefits. To prevent a non-invertible key matrix, the key matrix is constructed using the new binomial coefficient Newton. Plaintext may employ picture or text media and uses the same key for encryption and description. The key to encryption and decryption in the Hill Cipher technique is a $m \times m$ sized matrix. Multiplication between matrices and inverses the matrix is the basic matrix theory employed in Hill Cipher. Hill Cipher is a cryptographic algorithm that uses modulo arithmetic. This cryptography approach encrypts and decrypts using a square matrix as the key. When the Hill Cipher algorithm is used in the process of sending and receiving data, security is assumed to be assured. Users of data delivery services no longer have to be concerned about attackers attempting to compromise security systems and steal data thanks to this algorithm [1].

Text messaging is quite significant in this decade. When text messages are sent from one person to another, the process of protecting them is frequently done at the same time. This necessitates better text message security, both in terms of sending and keeping them. Encrypting text messages is one method. The Hill cipher is a well-known encryption algorithm. The Hill cipher has seen a number of changes in recent years. However, this approach only works with 3x3 or 4x4 finite key matrices. When the key matrix size exceeds 4, it is said that finding the inverse key is difficult or that finding the reversible key matrix would take a long time. A specific matrix, known as a unimodular matrix, can be used to overcome this problem [2]. Using basic row operations, we can create a unimodular matrix. It's unnecessary to utilize the complete matrix as a key. We'll create a unimodular matrix using a Bernoulli Map, which means we'll need fewer parameters. Furthermore, using the Python 2.7.14 programming language, the proposed approach will be implemented in many text messages. Some of the reasons we utilize Python are as follows. Python is a fairly simple language to learn. This application is also really simple to obtain. Python is available on a variety of operating systems. Python has several applications across a wide range of skill sets and disciplines [3], [4].

In this research, we use Python to implement a text message encryption technique using a unimodular matrix and a Bernoulli Map, as explained above. In Session 2, we'll talk about the research process and a review of the theory we employed, followed by our Python code. Session 3 will cover the implementation of the algorithm as well as some analysis, and Session 4 will wrap up this work.

the Research Methods

Cryptography is the study of encryption techniques in which plaintext is converted to ciphertext by encrypting it with a secret key. Someone without a decryption key will be unable to solve this document. The ciphertext will be decrypted using an agreed-upon key, and the original data will be returned. In the not-too-distant future, the chances of someone without a decryption key recovering the original text are extremely slim. In classical cryptography, symmetric encryption is utilized, in which the decryption key is the same as the encryption key. Asymmetric encryption techniques are necessary for public key cryptography when the decryption key is not the same as the encryption key. Because asymmetric encryption employs very big numbers, encryption, decryption, and key generation for asymmetric encryption techniques need more intense computer than symmetric encryption. Hill Cipher was designed by Lester S. Hill in 1929. This cryptographic approach was developed in order to build an encryption that could not be cracked via frequency analysis. Because Hill Cipher employs matrix multiplication for encryption and decryption, it does not replace each of the identical alphabets in plaintext with the same alphabet in ciphertext. Because the text to be processed will be separated into blocks of a specific size, the Hill Encryption, which is a polyalphabetic cipher, may be classified as a block cipher. In the encryption and decryption process, each character in a block has an impact on the other characters, ensuring that the same character is not mapped to the same character. Hill Cipher is one of the traditional cryptographic algorithms that cryptanalysts find extremely difficult to crack if they simply have access to the ciphertext file. However, if the cryptanalyst possesses a ciphertext file and a portion of the plaintext file, this approach can be quickly solved. A known-plaintext attack is the name of this cryptanalysis approach. Modulo arithmetic to the matrix is the foundation of the Hill Cipher algorithm. Hill Cipher employs matrix multiplication and inverse matrices methods in its implementation. The matrix n x n, where n is the block size, is the key to Hill Cipher. Because the K-1 matrix is the key required to decode, the K matrix that produces this key must be an invertible matrix with inverse K-1. As a result, the key must have an inverse [1], [5].

In the year Lester Hill found a way to encrypt a plaintext using the Linear Equation System. Before encrypting a plaintext, Hill Cipher divides it into blocks first. Ciphertext is generated by solving a System of Linear Equations consisting of n equations and n variables. The Linear Equation System can be solved using the matrix multiplication concept. Note that the concept of Hill Cipher belongs to a symmetrical cryptography, which means that the resulting key must have an inverse, which can be guaranteed if we use a unimodular matrix as the key matrix. The phrase "unimodular matrix" or "Nice Matrix" was coined by Hanson in 1982. A square matrix A is called a unimodular matrix if it occurs det(A) = -1 or det(A) = 1. Some examples of unimodular machins are the upper triangle matrix and the lower triangle matrix where all the main diagonal entries are 1 or -1. This statement is in accordance with the following theorem [6], [7].

Theorem 1. Anton (3) *Let A is a triangle matrix. Then* $\det(A) = a_{11}a_{22}\ldots a_{nn}$.

The following lemma describes the steps needed in generating a unimodular matrix. Note that Lemma 2 below will be the starting point for writing Python program code to build a unimodular matrix [8][9].

Lemma 2. Hanson (8) *A unimodular matrix A_n can be constructed in the following ways:*

1. *First, make a diagonal matrix with the diagonal entry $a_{ii} = 1$ or $a_{ii} = -1$.*
2. *Second, fill in any random integers at each entry with $i < j$. From this, it has formed a top triangular matrix whose determinants are 1 or -1. This is a unimodular matrix.*
3. *Third, to be a complete matrix, use elementary row operations (ERO) "add a row with multiply of another row".*

Based on Lemma 2 above, we will select the value of a_11 is 1 in Step one, as it will be applied to the positive integer value in this case. In the second stage, we will use the Bernoulli Map function to convert it into integers between 0 and 255. In the third step, we still using the Bernoulli Map function so that we can apply "add one line by multiplying another row". The derivative Bernoulli Map function is expressed as a set of real numbers between 0 and 1. Then we change from real to integer by selecting the first three digits after the decimal point and modulating with 256 [10], [11].

In 1976, May was the first to present the logistics map [12]. Despite the fact that one of the chaotic functions looks to be straightforward. The recursive formula is

$$x_{n+1} = rx_n \bmod 1.$$

One of the chaotic functions utilized in cryptography applications is the Bernoulli Map. This method creates rows of real numbers, which may then be utilized to create a keystream. The Bernoulli Map method, which was created to discover the keystream plant, was utilized to advance to the next phase by restricting the amount of integers raised to eight. This algorithm creates a sequence of real numbers that must be transformed into integers with a range of 0 to 255 in order to be used as a keystream. The method begins by luting the key rows generated from Equation (2.1), after which each of these keys is multiplied by 10000 and rounded down (floor) to form an integer row, following which the key is mapped in a range between 0 and 255 [13]. Bernoulli Maps produce a random series of numbers with a dispersion between 0 and 1 when r > 1. (0,1). There are several methods for constructing a row of integers with values ranging from 0 to 255, one of which is to modulate the results by 256 using three first digits following the decimal point of x given by the Bernoulli Map. In this inquiry, the first three digits after the decimal point were employed. In this paper, we offer four algorithms, each of which uses a unimodular matrix as the key. The matrix is created using Bernoulli Maps. Algorithms 1 and 2 are used in the encryption and decryption process [14], [15].

Algorithm 1: Generating Bernoulli Map sequence	**Algorithm 2: Generate Unimodular Matrix (Key)**
1. x = x0 #x0 as initial value 2. loop to 1000 times first to make sensitive sequence of Bernoulli Map 3. barisan = [] 4. for i in range(n): 5. x = r * x mod 1 # r > 1 6. barisan[i] = x*1000%256 # take 3 first digit after decimal point 7. return barisan	1. K=I_n 2. for i < j: K_ij = random numbers generated by Bernoulli Map 3. Use ERO add a row with multiply of another row modulo m to complete the matrix K.
Algorithm 3: Encryption	**Algorithm 4: Decryption**
1. Convert input text to a matrix P_(n) 2. Generate a key matrix K_m using Algorithm 2. If n is not divided by m, add more dummy character j to P such that m|(n+j) 3. Reshape the matrix P_(n+j) to a matrix P_(m, (n+j)/m) 4. C_(m, (n+j)/m) = K_m * P_(m, (n+j)/m) mod x 5. Reshape the matrix C_(m, (n+j)/m) to a matrix C_(n+j). 6. Convert the matrix C_(n+j) to encrypted text.	1. Convert input text to a matrix C_(n) 2. Generate a key matrix K_m using Algorithm 2. If n is not divided by m, add more dummy character j to P such that m|(n+j) 3. Find the invers matrix K_m^(-1) using ERO mod x 4. Reshape the matrix C_(n+j) to a matrix C_(m, (n+j)/m) 5. P_(m, (n+j)/m) = K_m^(-1) * C_(m, (n+j)/m) mod x 6. Reshape the matrix P_(m, (n+j)/m) to a matrix P_(n+j). 7. Convert the matrix P_(n+j) to decrypted text.

If we using the process in [6], one of the disadvantages is the existence of a key space for a small password 1, which is based only on the factor of the size of the image. As of this writing, password key 1 is expanded. This is still acceptable because the decrypted message still reads like the original message. Look at the following example. Suppose we want to encrypt a P statement that reads as follows:

kelompok kami berisi samsul arifin dan indra bayu muktyas

The sentence above we change first into the form of barisan or matrix line, which is as follows:

```
[107 101 108 111 109 112 111 107  32 107  97 109 105  32  98 101 114 105
 115 105  32 115  97 109 115 117 108  32  97 114 105 102 105 110  32 100
  97 110  32 105 110 100 114  97  32  98  97 121 117  32 109 117 107 116
 121  97 115  32  32  10]
```

Note that the length of P is 58 characters. Then we add P with the integer n in such a way up to 5l58+n. Note that in this case the value of n is 2 and the value of 58+2 is 60. As a result the set containing factors of 60 is {2, 3, 4, 5, 6, 10, 12, 15, 20, 60, 30}. Next, create a key matrix that is sized according to the user input, for example 5. This is password1. As a result, the key matrix built is a 5x5 unimodular matrix, obtained from algorithm 2. Furthermore, suppose password 2 is 211101, then based on the given 1 and 2 passwords, the 5x5 unimodular matrix built is as follows:

```
[[1.0000e+00 1.1400e+02 4.4800e+02 7.4700e+02 9.1500e+02]
 [3.3100e+02 3.7735e+04 3.8859e+04 2.7486e+04 2.8760e+04]
 [2.9200e+02 3.3288e+04 2.0817e+04 5.3616e+04 4.8102e+04]
 [1.3900e+02 1.5846e+04 7.2720e+03 4.8834e+04 1.7782e+04]
 [5.4400e+02 7.0160e+03 2.3712e+04 2.1368e+04 2.7610e+03]]
```

Note that the size of the P matrix is adjusted to the K matrix so that it can be multiplied, bringing the size of the P matrix to 5x12. Next is to do the encryption process that multiplies the matrix of K_5x5 and P_5x12, which produces a matrix C measuring 5x12, and if made into a matrix of rows will be as follows:

```
[23111  2615 18303 46451 16636 23994 36134 53987 14849 54983 17230 46872
 32439 51234  9942 12414 17972 48933 54825 52072 25259 53380 52352 43500
 29125 22321 15826 28258  5583 12514 48883  6464  2787 17866 48420  5476
 37375 52699 24222 36643 11393 49518 34977 27199 42698 16839 49171 31299
 32501 47592  1941 24461 30091 17852 22017 54025 47971 45784 23152 33378]
```

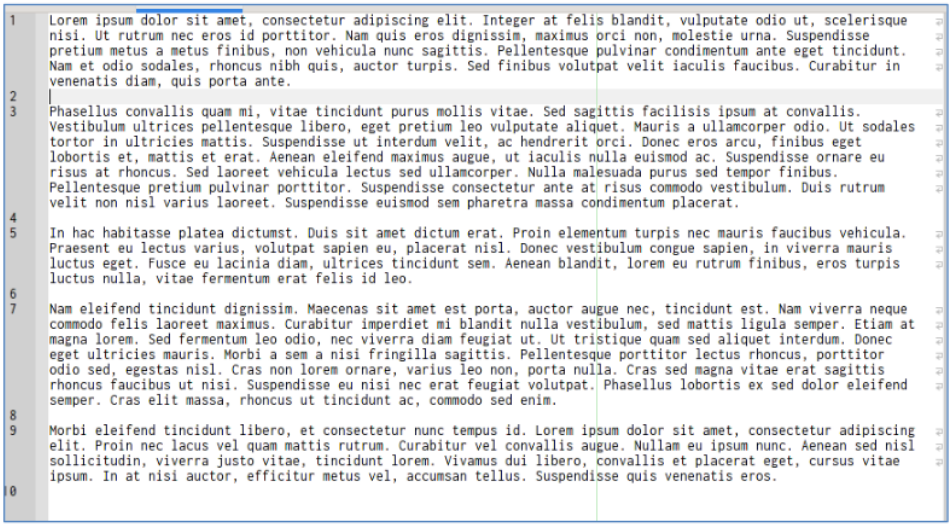
The matrix C_5x12 above indicates that the encryption process is complete. The next process is to decrypt, which is to multiply between the inverses of matrix K and C. Note that the unimodular matrix we use guarantees that every encryption process that is done can be decrypted again, because every unimodular matrix must have an inverse.

the Results of the Research and the Discussion

Here we used Intel® Xeon® CPU E5-2650 v2 @2.60GHz 2.60 GHz, RAM 32GB, Windows 10 64-bit. The propose algorithm was successfully implemented on text lorem ipsum.

```python
import numpy as np
from functools import reduce
import time

def factors(n):
    return set(reduce(list.__add__,
                ([i, n//i] for i in range(1, int(n**0.5) + 1) if n % i == 0)))


#OBE
def r_ij(m, baris_i, baris_j, r):
    return m[baris_i] + r*m[baris_j]

##########################

#barisan logistic map
def log(x0, banyak):
    x = x0
    for i in range(1000):
        x = 3.9 * x * (1 - x)     # 3.9 bisa diganti bil. pada [3.7, 4]

    barisan = np.zeros(banyak, dtype=np.uint16)
    for i in range(banyak):
        x = 3.9 * x * (1 - x)     # 3.9 bisa diganti bil. pada [3.7, 4]
        barisan[i] = x*1000%55000
    return barisan
###########################

def kunci(n, x0):
    # matriks segitiga atas
    banyak = int(n * (n - 1)/2)
    barisan = log(x0, banyak + n - 1)


    msa = np.eye(n)

    indeks = 0
    for i in range(n):
        for j in range(i+1,n):
            msa[i,j] = barisan[indeks]
            indeks += 1

```

The text we used as a test was dummy text taken from lipsum.com site that contained five paragraphs, which is as follows:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer at felis blandit, vulputate odio ut, scelerisque nisi. Ut rutrum nec eros id porttitor. Nam quis eros dignissim, maximus orci non, molestie urna. Suspendisse pretium metus a metus finibus, non vehicula nunc sagittis. Pellentesque pulvinar condimentum ante eget tincidunt. Nam et odio sodales, rhoncus nibh quis, auctor turpis. Sed finibus volutpat velit iaculis faucibus. Curabitur in venenatis diam, quis porta ante.

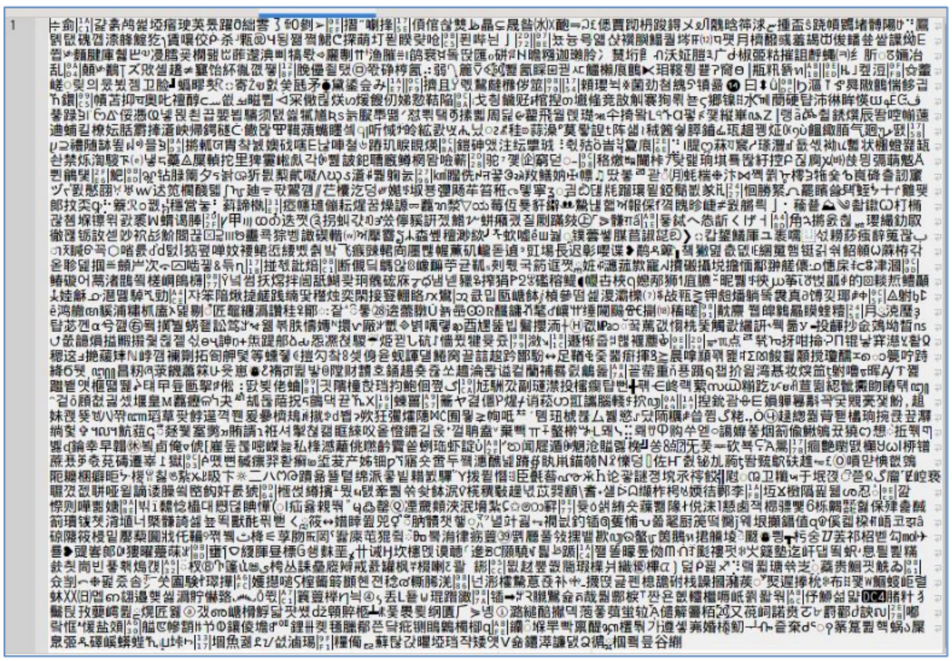
Phasellus convallis quam mi, vitae tincidunt purus mollis vitae. Sed sagittis facilisis ipsum at convallis. Vestibulum ultrices pellentesque libero, eget pretium leo vulputate aliquet. Mauris a ullamcorper odio. Ut sodales tortor in ultricies mattis. Suspendisse ut interdum velit, ac hendrerit orci. Donec eros arcu, finibus eget lobortis et, mattis et erat. Aenean eleifend maximus augue, ut iaculis nulla euismod ac. Suspendisse ornare eu risus at rhoncus. Sed laoreet vehicula lectus sed ullamcorper. Nulla malesuada purus sed tempor finibus. Pellentesque pretium pulvinar porttitor. Suspendisse consectetur ante at risus commodo vestibulum. Duis rutrum velit non nisl varius laoreet. Suspendisse euismod sem pharetra massa condimentum placerat.

In hac habitasse platea dictumst. Duis sit amet dictum erat. Proin elementum turpis nec mauris faucibus vehicula. Praesent eu lectus varius, volutpat sapien eu, placerat nisl. Donec vestibulum congue sapien, in viverra mauris luctus eget. Fusce eu lacinia diam, ultrices tincidunt sem. Aenean blandit, lorem eu rutrum finibus, eros turpis luctus nulla, vitae fermentum erat felis id leo.

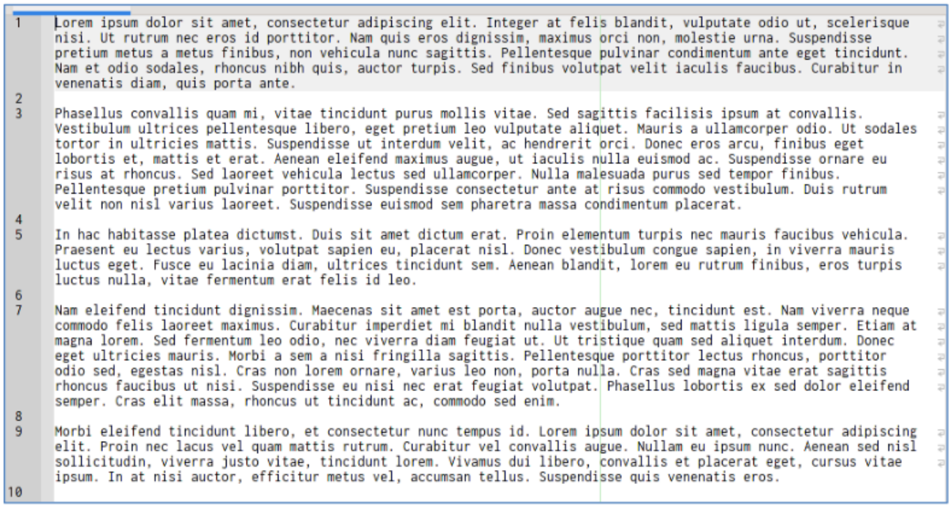
Nam eleifend tincidunt dignissim. Maecenas sit amet est porta, auctor augue nec, tincidunt est. Nam viverra neque commodo felis laoreet maximus. Curabitur imperdiet mi blandit nulla vestibulum, sed mattis ligula semper. Etiam at magna lorem. Sed fermentum leo odio, nec viverra diam feugiat ut. Ut tristique quam sed aliquet interdum. Donec eget ultricies mauris. Morbi a sem a nisi fringilla sagittis. Pellentesque porttitor lectus rhoncus, porttitor odio sed, egestas nisl. Cras non lorem ornare, varius leo non, porta nulla. Cras sed magna vitae erat sagittis rhoncus faucibus ut nisi. Suspendisse eu nisi nec erat feugiat volutpat. Phasellus lobortis ex sed dolor eleifend semper. Cras elit massa, rhoncus ut tincidunt ac, commodo sed enim.

Morbi eleifend tincidunt libero, et consectetur nunc tempus id. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nec lacus vel quam mattis rutrum. Curabitur vel convallis augue. Nullam eu ipsum nunc. Aenean sed nisl sollicitudin, viverra justo vitae, tincidunt lorem. Vivamus dui libero, convallis et placerat eget, cursus vitae ipsum. In at nisi auctor, efficitur metus vel, accumsan tellus. Suspendisse quis venenatis eros.

The encryption result of the text with key 1: 10 and key 2: 22021985 is as follows.



The characters are not well read. It's good for encryption. Then after decrypting the file is as follows.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer at felis blandit, vulputate odio ut, scelerisque nisi. Ut rutrum nec eros id porttitor. Nam quis eros dignissim, maximus orci non, molestie urna. Suspendisse pretium metus a metus finibus, non vehicula nunc sagittis. Pellentesque pulvinar condimentum ante eget tincidunt. Nam et odio sodales, rhoncus nibh quis, auctor turpis. Sed finibus volutpat velit iaculis faucibus. Curabitur in venenatis diam, quis porta ante.

Phasellus convallis quam mi, vitae tincidunt purus mollis vitae. Sed sagittis facilisis ipsum at convallis. Vestibulum ultrices pellentesque libero, eget pretium leo vulputate aliquet. Mauris a ullamcorper odio. Ut sodales tortor in ultricies mattis. Suspendisse ut interdum velit, ac hendrerit orci. Donec eros arcu, finibus eget lobortis et, mattis et erat. Aenean eleifend maximus augue, ut iaculis nulla euismod ac. Suspendisse ornare eu risus at rhoncus. Sed laoreet vehicula lectus sed ullamcorper. Nulla malesuada purus sed tempor finibus. Pellentesque pretium pulvinar porttitor. Suspendisse consectetur ante at risus commodo vestibulum. Duis rutrum velit non nisl varius laoreet. Suspendisse euismod sem pharetra massa condimentum placerat.

In hac habitasse platea dictumst. Duis sit amet dictum erat. Proin elementum turpis nec mauris faucibus vehicula. Praesent eu lectus varius, volutpat sapien eu, placerat nisl. Donec vestibulum congue sapien, in viverra mauris luctus eget. Fusce eu lacinia diam, ultrices tincidunt sem. Aenean blandit, lorem eu rutrum finibus, eros turpis luctus nulla, vitae fermentum erat felis id leo.

Nam eleifend tincidunt dignissim. Maecenas sit amet est porta, auctor augue nec, tincidunt est. Nam viverra neque commodo felis laoreet maximus. Curabitur imperdiet mi blandit nulla vestibulum, sed mattis ligula semper. Etiam at magna lorem. Sed fermentum leo odio, nec viverra diam feugiat ut. Ut tristique quam sed aliquet interdum. Donec eget ultricies mauris. Morbi a sem a nisi fringilla sagittis. Pellentesque porttitor lectus rhoncus, porttitor odio sed, egestas nisl. Cras non lorem ornare, varius leo non, porta nulla. Cras sed magna vitae erat sagittis rhoncus faucibus ut nisi. Suspendisse eu nisi nec erat feugiat volutpat. Phasellus lobortis ex sed dolor eleifend semper. Cras elit massa, rhoncus ut tincidunt ac, commodo sed enim.

Morbi eleifend tincidunt libero, et consectetur nunc tempus id. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nec lacus vel quam mattis rutrum. Curabitur vel convallis augue. Nullam eu ipsum nunc. Aenean sed nisl sollicitudin, viverra justo vitae, tincidunt lorem. Vivamus dui libero, convallis et placerat eget, cursus vitae ipsum. In at nisi auctor, efficitur metus vel, accumsan tellus. Suspendisse quis venenatis eros.

The full program code and example can be accessed at the following link: https://github.com/muktyas/text-encryption-chaos-unimodular

Conclusion and Suggestion

The conventional Hill cipher uses a small key matrix Kn, n = 4. Furthermore, using the usual Hill cipher, if n > 4, the entire matrix K n is used as the key. In this research, we use a Bernoulli Map as a key to build a Unimodular matrix that can solve the problem. That is, n > 4, yet only two arguments are required (password 1 and password 2). The encrypted text messages we tested were difficult to decipher in normal languages, according to the results of our experiments.

Acknowledgement

The authors would also want to express their gratitude to the reviewers for their insightful comments, recommendations, and ideas, which have helped to shape this article into something that deserves to be published.

References

[1] A. P. U. Siahaan, "Application of Hill Cipher Algorithm in Securing Text Messages," pp. 55–59, 2018, doi: 10.31227/osf.io/n2kdb.

[2] S. Arifin and I. B. Muktyas, "Membangkitkan Suatu Matriks Unimodular Dengan Python," *J. Deriv. J. Mat. dan Pendidik. Mat.*, vol. 5, no. 2, pp. 1–10, 2018.

[3] T. E. Oliphant, "Python for scientific computing," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 10–20, 2007.

[4] S. Arifin, I. Bayu Muktyas, and K. Iswara Sukmawati, "Product of two groups integers modulo m,n and their factor groups using python," in *Journal of Physics: Conference Series*, Mar. 2021, vol. 1778, no. 1, doi: 10.1088/1742-6596/1778/1/012026.

[5] S. W, *Cryptography and Network Security: Principles and Practices*, 4th ed. Upper Saddle River: Pearson Prentice Hall, 2006.

[6] I. B. Muktyas, Sulistiawati, and S. Arifin, "Digital image encryption algorithm through unimodular matrix and logistic map using Python," in *AIP Conference Proceedings*, Apr. 2021, vol. 2331, doi: 10.1063/5.0041653.

[7] H. Anton and C. Rorres, *Elementary linear algebra: applications version*. John Wiley & Sons, 2013.

[8] R. Hanson, "Integer matrices whose inverses contain only integers," *Two-Year Coll. Math. J.*, vol. 13, no. 1, pp. 18–21, 1982.

[9] S. Arifin and I. B. Muktyas, "Generate a system of linear equation through unimodular matrix using Python and Latex," in *AIP Conference Proceedings*, Apr. 2021, vol. 2331, doi: 10.1063/5.0041651.

[10] K. H. Rosen, *Discrete Mathematics and its Applications*, 8th ed. New York, USA: McGraw-Hill Education, 2019.

[11] A. Delmi, S. Suryadi, and Y. Satria, "Digital image steganography by using edge adaptive based chaos cryptography," *J. Phys. Conf. Ser.*, vol. 1442, no. 1, 2020, doi: 10.1088/1742-6596/1442/1/012041.

[12] R. Devaney, *An Introduction To Chaotic Dynamical Systems*. CRC Press, 2018.

[13] S. Maulida, "Pengamanan Citra Biner dengan Algoritma Data Encryption Standard (DES) dan Bernoulli Map," UNIVERSITAS JEMBER, 2018.

[14] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, 2006, doi: https://doi.org/10.1016/j.imavis.2006.02.021.

[15] R. Ye and Y. Ma, "A Secure and Robust Image Encryption Scheme Based on Mixture of Multiple Generalized Bernoulli Shift Maps and Arnold Maps," *Int. J. Comput. Netw. Inf. Secur.*, vol. 5, no. 7, pp. 21–33, 2013, doi: 10.5815/ijcnis.2013.07.03.

Paper Al-Jabar-[29Nov2021]b

ORIGINALITY REPORT

20%	4%	17%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Indra Bayu Muktyas, Sulistiawati, Samsul Arifin. "Digital image encryption algorithm through unimodular matrix and logistic map using Python", AIP Publishing, 2021 Publication	5%
2	Submitted to University of Muhammadiyah Malang Student Paper	5%
3	Nurhayati, Abdul Meizar, Frinto Tambunan, Erwin Ginting. "Optimizing the Complexity of Time in the Process of Multiplying Matrices in the Hill Cipher Algorithm Using the Strassen Algorithm", 2019 7th International Conference on Cyber and IT Service Management (CITSM), 2019 Publication	3%
4	Samsul Arifin, Indra Bayu Muktyas. "Generate a system of linear equation through unimodular matrix using Python and Latex", AIP Publishing, 2021 Publication	3%

5	Anvit Negi, Devansh Saxena, Kriti Suneja. "High Level Synthesis of Chaos based Text Encryption Using Modified Hill Cipher Algorithm", 2020 IEEE 17th India Council International Conference (INDICON), 2020 Publication	2%
6	Shengming Jiang. "On Securing Underwater Acoustic Networks: A Survey", IEEE Communications Surveys & Tutorials, 2019 Publication	1%
7	Arif Mandangan, Hailiza Kamarulhaili, Muhammad Asyraf Asbullah. "Square Integer Matrix with a Single Non-Integer Entry in Its Inverse", Mathematics, 2021 Publication	1%
8	www.innoarchitech.com Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		

Paper Al-Jabar-[29Nov2021]b

PAGE 1

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Hyph. You may need to add a hyphen between these two words.

Article Error You may need to remove this article.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Article Error You may need to use an article before this word.

PAGE 2

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Missing "," You may need to place a comma after this word.

Missing "," You may need to place a comma after this word.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Confused You have used **a** in this sentence. You may need to use **an** instead.

P/V You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

Article Error You may need to use an article before this word.

Article Error You may need to use an article before this word.

P/V You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

Article Error You may need to use an article before this word.

Prep. You may be using the wrong preposition.

PAGE 3

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Article Error You may need to use an article before this word.

Proofread This part of the sentence contains a grammatical error or misspelled word that makes your meaning unclear.

P/V You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

Confused You have used **a** in this sentence. You may need to use **an** instead.

P/V You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

Confused You have used **A** in this sentence. You may need to use **an** instead.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Confused You have used **A** in this sentence. You may need to use **an** instead.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Proper Noun If this word is a proper noun, you need to capitalize it.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Confused You have used **A** in this sentence. You may need to use **an** instead.

Proofread This part of the sentence contains a grammatical error or misspelled word that makes your meaning unclear.

Confused You have used **ð´ð** in this sentence. You may need to use **a** instead.

Proper Noun If this word is a proper noun, you need to capitalize it.

Confused You have used **a** in this sentence. You may need to use **an** instead.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

P/V You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Article Error You may need to use an article before this word. Consider using the article **the**.

P/V You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

Confused You have used **a** in this sentence. You may need to use **an** instead.

PAGE 5

Verb This verb may be incorrect. Proofread the sentence to make sure you have used the correct form of the verb.

Prep. You may be using the wrong preposition.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Proper Noun If this word is a proper noun, you need to capitalize it.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Run-on This sentence may be a run-on sentence. Proofread it to see if it contains too many independent clauses or contains independent clauses that have been combined without conjunctions or punctuation. Look at the "Writer's Handbook" for advice about correcting run-on sentences.

Missing "," You may need to place a comma after this word.

Article Error You may need to remove this article.

Missing "," You may need to place a comma after this word.

Article Error You may need to remove this article.

Article Error You may need to remove this article.

Word Error Did you type "**the**" instead of "**they**," or have you left out a word?

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Article Error You may need to use an article before this word. Consider using the article **a**.
