Bulletin of Applied Mathematics and Mathematics Education
Volume 1, Number 2, pp. 75-92
https://dx.doi.org/10.12928/bamme.v1i2.4252
e-ISSN 2776-1029, p-ISSN 2776-1002

An implementation of Hill Cipher and $3 \times 3 \times 3$ rubik's cube to enhance communication security

Nana, Puguh Wahyu Prasetyo*

Universitas Ahmad Dahlan, Jl. Ahmad Yani, Tamanan, Bantul, DIY 55711, Indonesia

*Corresponding e-mail: puguh.prasetyo@pmat.uad.ac.id

ARTICLE INFO

Article History
Received 16 June 2021
Revised 22 November 2021
Accepted 22 November 2021

Keywords
Applied mathematics;
Data analysis;
Mathematics literacy.

How to cite this article:
Nana, & Prasetyo, P. W. (2021). An implementation of Hill cipher and $3 \times 3 \times 3$ rubik's cube to enhance communication security. *Bulletin of Applied Mathematics and Mathematics Education*, *1*(2), 75-92.

ABSTRACT

Message security is something that must be taken seriously. Therefore, to maintain the confidentiality of any message, cryptography is needed. Cryptography is a science that uses mathematics to encrypt and decrypt messages. Cryptography is used as a tool to protect messages, for example, national secrets and strategies. The method of this research is a qualitative research with a literature review. This research implements a hybrid cryptographic algorithm by combining Hill cipher and $3 \times 3 \times 3$ Rubik's cube methods with Python software simulation.

Introduction

Information is a very significant asset for an organization because it becomes an essential resource in increasing business value. This causes information security to be essential for every organization. Information security refers to all aspects that are used to protect various types of threats or affect the organization's sustainability. It follows from (Indrajit, 2011) that information technology is like a double-edged sword because, on the other hand, it benefits from applications such as e-government, e-commerce, e-society, and e-education. On the other hand, information technology also has weaknesses such as commercial crimes, character assassinations, fraud, covert wiretapping. Today, human life is integrated with a network that is transmitted through the internet network every moment. In sharing this information, security is needed to ensure the safety of information from various attacks that can spread information that should be confidential (Kalaichelvi et al., 2017). Moreover, cryptography is designed as a science that uses mathematics to encrypt and decrypt messages to enhance the security of the messages. Furthermore, it follows from (Schneier, 1996) that cryptography can be seen as the art and science of keeping messages secure. One of the cryptosystems of cryptography is the Hill cipher. Lester S. Hill invented the Hill cipher in 1929. Hill cipher is a well-known symmetric cryptosystem that multiplies the plaintext vector by the key matrix to get the ciphertext. Hill ciphers are resistant to brute-force and statistical attacks. However, it can be hacked by plaintext-ciphertext attacks known as known plaintext-ciphertext attacks (KCPA).

A Hill cipher modification based on Pseudo-Random Eigenvalues (HCM-PRE) has been proposed in a previous study, which is still resistant to brute-force and statistical attacks and is also resistant to KPCA because it produces a dynamic encryption key matrix (Mahmoud & Chefranov, 2014). Furthermore, research conducted by Kalaichelvi (Kalaichelvi et al., 2017) also proposed a new variation of the Hill cipher encryption algorithm to provide data security by using Radix 64 to overcome the weakness when encrypting identical plaintext blocks to identical ciphertext blocks. Meanwhile, (Hraoui et al., 2019) proposed an improvement to the Hill cipher algorithm in the case of Affine transformation. In this study, the authors propose a cryptographic algorithm that combines the Hill cipher method with the Rubik's cube. Rubik's cube is used because it is a three-dimensional mechanical puzzle game with 43 quintillion different configurations (Raymond, 2005). The Rubik's cube was invented in 1974 by a Hungarian sculptor and professor of architecture named Erno Rubik. In 2016, Abitha and Pradeep proposed communication security based on the Rubik's cube algorithm. One of the steps used is using the Rubik's Cube principle in encrypting an image (Abitha & Bharathan, 2016). In this study, the authors use Python software in the simulation and implementation process. Hence, this study aims to figure out how to combine the encryption and decryption processes from the Hill cipher method and the Rubik's cube.

Hill Cipher

Hill cipher is a polyalphabetic cryptosystem. This means that each character of the alphabet can be mapped to more than one-type of character. Now let m be a positive integer and suppose $\mathcal{P} = \mathcal{C} = (\mathrm{Z}_{26})^m$. The idea the Hill Cipher is derived from m linear combination formed by m alphabet character in one-element of plaintext (Stinson, 2018).

Let $m = 2$. Then it can be written a plaintext, namely $x = (x_1, x_2)$ and a ciphertext $y = (y_1, y_2)$, where y_1, y_2 are linear combinations of x_1 and x_2. Suppose we have the following linear combinations

$$y_1 = 11x_1 + 3x_2 \tag{1}$$
$$y_2 = 8x_1 + 7x_2 \tag{2}$$

We therefore have the following matrix representation of the equation (1) and (2)

$$(y_1, y_2) = (x_1, x_2)\begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \tag{3}$$

In general, by using a $m \times m$ matrix, namely $\mathcal{K}$, as a key. Now suppose that the element of i −row and j −coloumn of the matrx $\mathcal{K}$ be $k_{i,j}$, then the matix can be written as $\mathcal{K} = (k_{i,j})$. Let $x = (x_1, \cdots, x_m) \in \mathcal{P}$ and $K \in \mathcal{K}$, the ciphertext $y = e_k(x) = (y_1, \cdots, y_m)$ is described as follow:

$$(y_1, y_2, \cdots, y_m) = (x_1, x_2, \cdots, x_m)\begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{bmatrix} \tag{4}$$

In other words, $y = x\mathcal{K}$. Hence, it can be concluded that in Hill cipher process the ciphertexts derived from plaintext can be formed by using linear transformation. Thus, this condition motivates us to calculate how the decryption process works. By using the inverse matrix $\mathcal{K}^{-1}$ of $\mathcal{K}$, the

10.12928/bamme.v1i2.4252

ciphertext can be formed into plaintext by decryption process with the formula $x = y\mathcal{K}^{-1}$ (Stinson, 2018).

Rubik's Cube

Rubik's Cube is a three-dimensional mechanical puzzle game invented in 1974 by a Hungarian sculptor and professor of architecture named Erno Rubik. The standard $3 \times 3 \times 3$ Rubik's cube is one of the combinatorial problems that is well known for its complexity. There are $43{,}2 \times 10^{18}$ different configurations that may be generated in the randomization process (Raymond, 2005). The illustration of the $3 \times 3 \times 3$ Rubik's cube can be seen in the Figure 1.

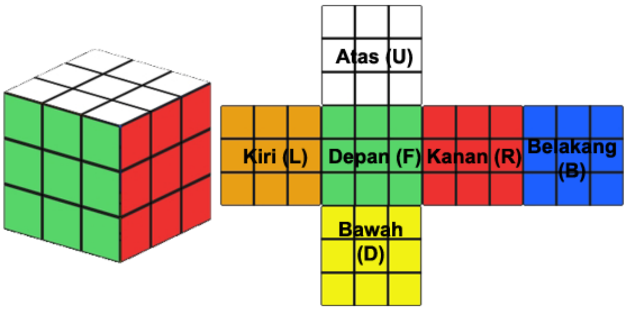


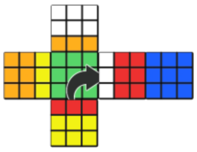
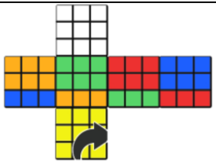
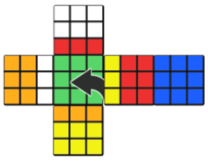
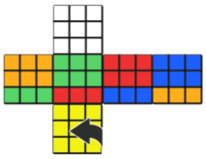
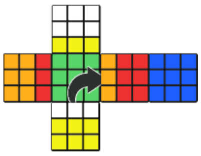
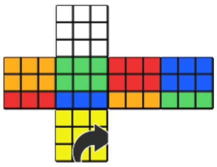
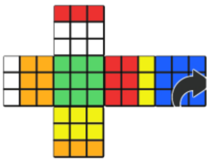
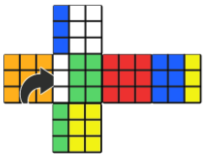
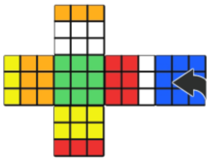
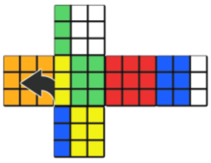
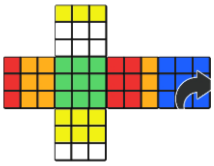
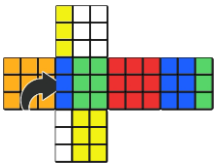
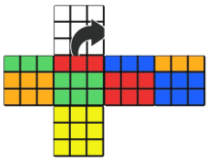
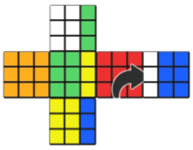
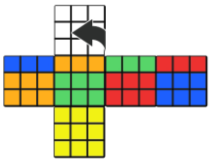
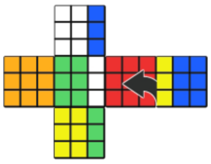
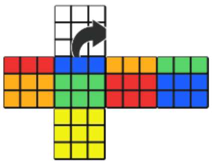
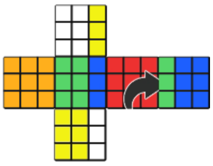
Figure 1. The 3×3×3 Rubik's cube

Notation on the Rubik's cube is essential stuff since it helps to demonstrate the algorithm to be more clear. There are various method to show the side of the Rubik's cube. The British mathematician, David Singmaster, invented some adapted notation to show the order of the Rubik's cube movement. The capital will denote the Rubik's cube side. The notation are described as F (*depan*/ front*)*, U (*atas*/ up), D (*bawah*/ down), L (*kiri*/ left), R (*kanan*/ right) (Khuyen, 2016). Furthermore, the Rubik's cube movements are illustrated in the Table 1.

Phyton

Python is a multipurpose interpretive programming language with a design philosophy that focuses on code readability. Python is considered as a language that combines capabilities with a very clear code structure, and it is equipped with an extensive and comprehensive standard library functionality. Python primarily supports multiple programming paradigms but is not restricted in object-oriented programming, imperative programming, and functional programming. One of the features available in Python is as a dynamic programming language that is equipped with automatic memory management. As with other dynamic programming languages, Python is generally used as a scripting language, although in practice, this language is used more broadly to include contexts of use that are usually not done by using scripting languages. Python can be used for various software development purposes, and it can be run on various operating systems. Python is distributed under several different licenses from several versions. But in principle, Python can be obtained and used freely, even for commercial purposes. The Python license does not contradict either the definition of the open-source or the General Public License (GPL) (Sinaga, 2017).

Table 1. The Movement Notation of 3 × 3 × 3 Rubik's Cube

Notation	Figure	Note	Notation	Figure	Note
F		Turn the entire front side by 90^0 clockwise	D		Turn the entire front side by 90^0 clockwise
F'		Turn the entire front side by 90^0 counterclockwise	D'		Turn the entire front side by 90^0 counterclockwise
F2		Rotate the entire front side by 180^0	D2		Rotate the entire downside by 180^0
B		Rotate the entire backside by 90^0 clockwise	L		Rotate the entire left side by 90^0 clockwise
B'		Rotate the entire backside by 90^0 counterclockwise	L'		Rotate the entire left side by 90^0 counterclockwise
B2		Rotate the entire backside by 180^0	L2		Rotate the entire backside by 180^0
U		Rotate the entire upside by 90^0 clockwise	R		Rotate the entire right side by 90^0 clockwise
U'		Rotate the entire upside by 90^0 counterclockwise	R'		Rotate the entire right side by 90^0 counterclockwise
U2		Rotate the entire upside by 180^0	R2		Rotate the entire right side by 180^0

Method

This research is qualitative research with the literature review resources to gain some mathematical aspects and concepts. We start the research by reviewing some resources related to basic cryptography, Hill cipher, $3 \times 3 \times 3$ Rubik's cube, and its properties. Furthermore, the fundamental concept gained from the previous resources is managed to be a tool to solve the formulated problem, that is, the combined Hill cipher and $3 \times 3 \times 3$ Rubik's cube. Moreover, we use this result to implement the combined Hill cipher and $3 \times 3 \times 3$ Rubik's cube by using Phyton, an open-source software license. It follows from (Danandjaja, 2014) that Library searches are not only the first step in preparing a research design but also at the same time utilizing library resources to obtain research data.

Results and Discussion

Encryption Process

In this research, we will use the following steps to encrypt the plaintext in the Hill cipher process.

a. Considering the plaintext,
b. Separating the plaintext into blocks,
c. Determining the key, which is relatively prime to 54,
d. Converting the characters of the plaintext using the part of ASCII symbol described in Table 2, which was started previously by adding %,
e. Encryption using Hill cipher method,
f. Converting the numbers into alphabet or character,
g. Gaining the ciphertext.

Before the encryption process, the plaintext should be converted into a character from Table 2.

Table 2. Convertion Table into ASCII character

ASCII Character and its representation																	
%	&	‘	(	)	*	+	,	-	.	/	0	1	2	3	4	5	6
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
7	8	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G	H
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53

The key used in this method is an invertible matrix so that the ciphertext can be returned into its plaintext. Since the space used in this research is $Z_{54} = \{0,1,2,\dots,53\}$, the matrix $\mathcal{K}$ used in this process should be an invertible matrix modulo 54.

Encryption Result

Encryption using Hill cipher

In this part, we will simulate the encryption result by using a plaintext: UNIVERSITAS AHMAD DAHLAN. We consider the key as JH67 since it is invertible modulo 54. It follows from the character representation illustrated in Table 2. We, therefore, have Table 3.

Table 3. Representation of the plaintext "UNIVERSITAS AHMAD DAHLAN"

Plaintext Character and Its Representation																					
U	N	I	V	E	R	S	I	T	A	S	A	H	M	A	D	D	A	H	L	A	N
48	41	36	49	32	45	46	36	47	28	46	28	35	40	28	31	31	28	35	39	28	41

Furthermore, the representation of the key character of JH67 is 37 35 17 18. So the matrix representation of the key is $\begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}$. Since the key used in this process is a 2 × 2 matrix, the converted text will be separated into some part, namely, blocks which consist of 2 alphabets, so we have Table 4.

Table 4. Separation of the converted plaintext

Block I	Block II	Block III	Block IV	Block V	Block VI	Block VII	Block VIII	Block IX	Block X	Block XI
48 41	36 49	32 45	46 36	47 28	46 28	35 40	28 31	31 28	35 39	28 41

Moreover, the encryption prosesses are done one by one of each block as follows

$$\begin{aligned}
&\text{Block I} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}48\\ 41\end{bmatrix} = \begin{bmatrix}1776+697\\ 1680+738\end{bmatrix} = \begin{bmatrix}2473\\ 2418\end{bmatrix} mod\ 54 = \begin{bmatrix}43\\ 42\end{bmatrix} = \begin{bmatrix}P\\ O\end{bmatrix}\\
&\text{Block II} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}36\\ 49\end{bmatrix} = \begin{bmatrix}1332+833\\ 1280+882\end{bmatrix} = \begin{bmatrix}2165\\ 2142\end{bmatrix} mod\ 54 = \begin{bmatrix}5\\ 36\end{bmatrix} = \begin{bmatrix}*\\ I\end{bmatrix}\\
&\text{Block III} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}32\\ 45\end{bmatrix} = \begin{bmatrix}1184+765\\ 1120+810\end{bmatrix} = \begin{bmatrix}1949\\ 1930\end{bmatrix} mod\ 54 = \begin{bmatrix}5\\ 40\end{bmatrix} = \begin{bmatrix}*\\ M\end{bmatrix}\\
&\text{Block IV} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}46\\ 36\end{bmatrix} = \begin{bmatrix}1702+612\\ 1610+648\end{bmatrix} = \begin{bmatrix}2314\\ 2258\end{bmatrix} mod\ 54 = \begin{bmatrix}46\\ 44\end{bmatrix} = \begin{bmatrix}S\\ Q\end{bmatrix}\\
&\text{Block V} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}47\\ 28\end{bmatrix} = \begin{bmatrix}1739+476\\ 1645+504\end{bmatrix} = \begin{bmatrix}2215\\ 2149\end{bmatrix} mod\ 54 = \begin{bmatrix}1\\ 43\end{bmatrix} = \begin{bmatrix}\&\\ P\end{bmatrix}\\
&\text{Block VI} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}46\\ 28\end{bmatrix} = \begin{bmatrix}1702+476\\ 1610+504\end{bmatrix} = \begin{bmatrix}2178\\ 2114\end{bmatrix} mod\ 54 = \begin{bmatrix}18\\ 8\end{bmatrix} = \begin{bmatrix}7\\ -\end{bmatrix}\\
&\text{Block VII} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}35\\ 40\end{bmatrix} = \begin{bmatrix}1295+680\\ 1225+720\end{bmatrix} = \begin{bmatrix}1975\\ 1945\end{bmatrix} mod\ 54 = \begin{bmatrix}31\\ 1\end{bmatrix} = \begin{bmatrix}D\\ \&\end{bmatrix}\\
&\text{Block VIII} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}28\\ 31\end{bmatrix} = \begin{bmatrix}1036+527\\ 980+558\end{bmatrix} = \begin{bmatrix}1563\\ 1538\end{bmatrix} mod\ 54 = \begin{bmatrix}51\\ 26\end{bmatrix} = \begin{bmatrix}X\\ ?\end{bmatrix}\\
&\text{Block IX} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}31\\ 28\end{bmatrix} = \begin{bmatrix}1147+476\\ 1085+504\end{bmatrix} = \begin{bmatrix}1623\\ 1589\end{bmatrix} mod\ 54 = \begin{bmatrix}3\\ 23\end{bmatrix} = \begin{bmatrix}(\\ <\end{bmatrix}\\
&\text{Block X} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}35\\ 39\end{bmatrix} = \begin{bmatrix}1295+663\\ 1225+702\end{bmatrix} = \begin{bmatrix}1958\\ 1927\end{bmatrix} mod\ 54 = \begin{bmatrix}14\\ 37\end{bmatrix} = \begin{bmatrix}3\\ J\end{bmatrix}\\
&\text{Block XI} && \begin{bmatrix}37 & 17\\ 35 & 18\end{bmatrix}\begin{bmatrix}28\\ 41\end{bmatrix} = \begin{bmatrix}1036+697\\ 980+738\end{bmatrix} = \begin{bmatrix}1733\\ 1718\end{bmatrix} mod\ 54 = \begin{bmatrix}5\\ 44\end{bmatrix} = \begin{bmatrix}*\\ Q\end{bmatrix}
\end{aligned}$$

It follows from the encryption processes starting from the block I to block XI, we have the following ciphertext

PO*I*MSQ&P7-D&X?(<3J*Q

Encryption using 3 × 3 × 3 *Rubik's Cube*

The process of encoding or encryption on the Rubik's cube is carried out in the following stages:

a. Determine the position or location of the plaintext to be encrypted in the Rubik's cube blocks. The existing plaintext characters are then placed on the Rubik's cube blocks in the order shown in the Figure 2.

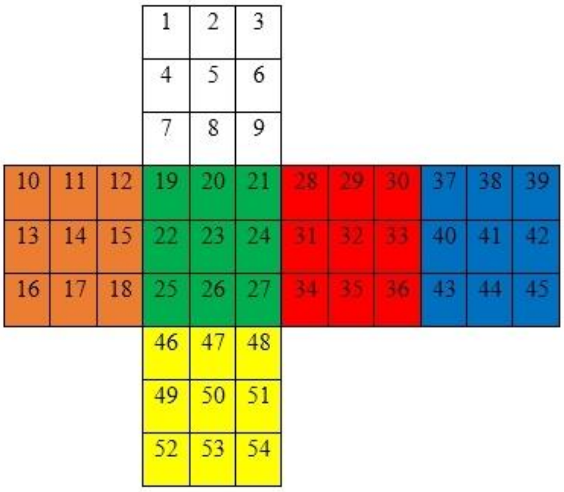


Figure 2. Position of laying plaintext on Rubik

b. Fills the specified block with plaintext. If the plaintext has a total of 40 characters, then characters 41 to 54 are filled with the "%" sign. With only 54 Rubik's Cube blocks, only 54 characters can be encrypted. The illustration of this process can be seen in Figure 3.

c. Specifiy the desired key. The key used is obtained from the notation of the movement of the Rubik's cube can be 10 steps or more, so that the resulting password is more complicated.

d. Randomize the Rubik's Cube with a predetermined key.

e. Write down the ciphertext that has been generated.

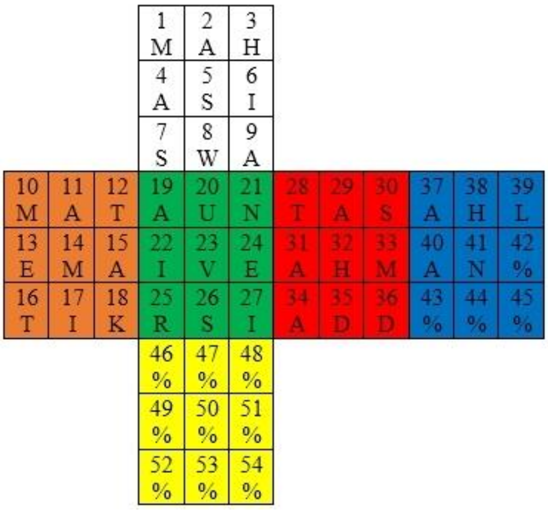

1 M 2 A 3 H
4 A 5 S 6 I
7 S 8 W 9 A
10 M 11 A 12 T 19 A 20 U 21 N 28 T 29 A 30 S 37 A 38 H 39 L
13 E 14 M 15 A 22 I 23 V 24 E 31 A 32 H 33 M 40 A 41 N 42 %
16 T 17 I 18 K 25 R 26 S 27 I 34 A 35 D 36 D 43 % 44 % 45 %
46 % 47 % 48 %
49 % 50 % 51 %
52 % 53 % 54 %

Figure 3. Example of laying plaintext on rubik

The ciphertext derived from the Hill cipher process will be considered as a plaintext for the encryption by Rubik's cube. Hence, we have the following message as the plaintext and the chosen key.

Plaintext: PO*I*MSQ&P7-D&X?(<3J*Q

Key: R U' L U R' B D L' R' F'

The initial process of the Rubik's cube can be seen in Figure 4.

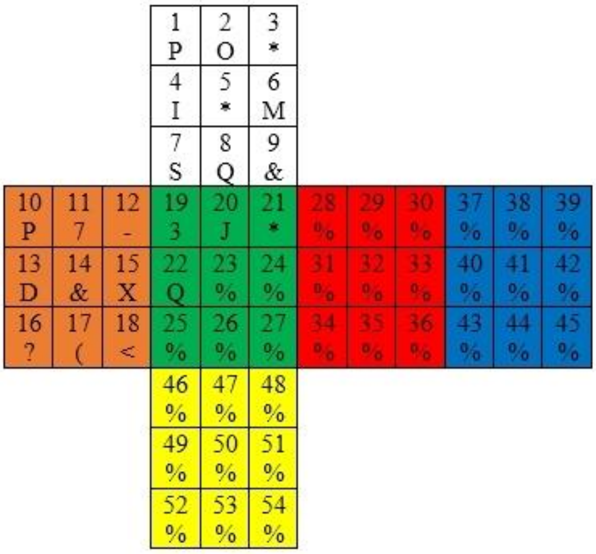


Figure 4. Initialize the encryption process on rubik

We use several steps to gain the final ciphertext as follows

a. Step 1

At the initialization step, the Rubik's cube is rotated on the right by 90^0 clockwise (R). Hence, the position of the white Rubik's block fills the blue position, blue block fills the yellow block position, yellow fills block the green block position and green block fills the white block position so that the resulting rubik is as described in Figure 5.

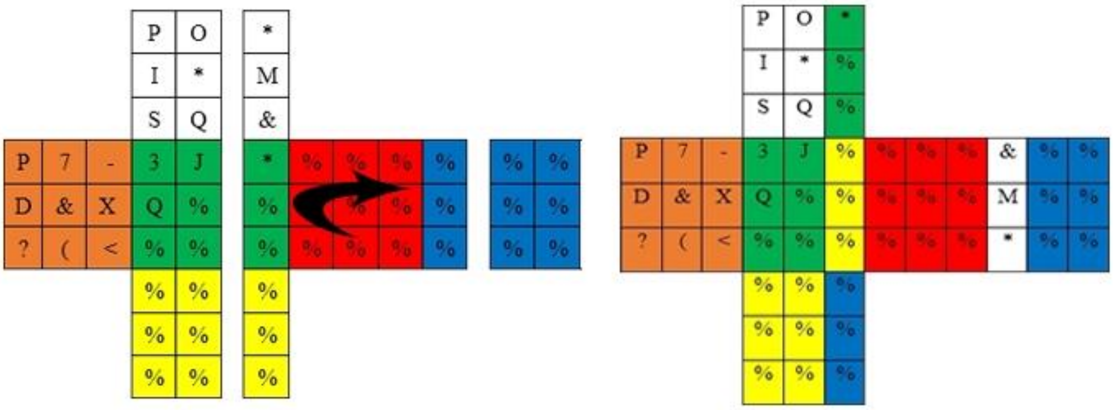


Figure 5. Illustration of the step 1

b. Step 2

The results in the first step, the resulting Rubik is rotated at the top by 90^0 counterclockwise (U'). Hence, we therefore the rubik as described in Figure 6.

10.12928/bamme.v1i2.4252

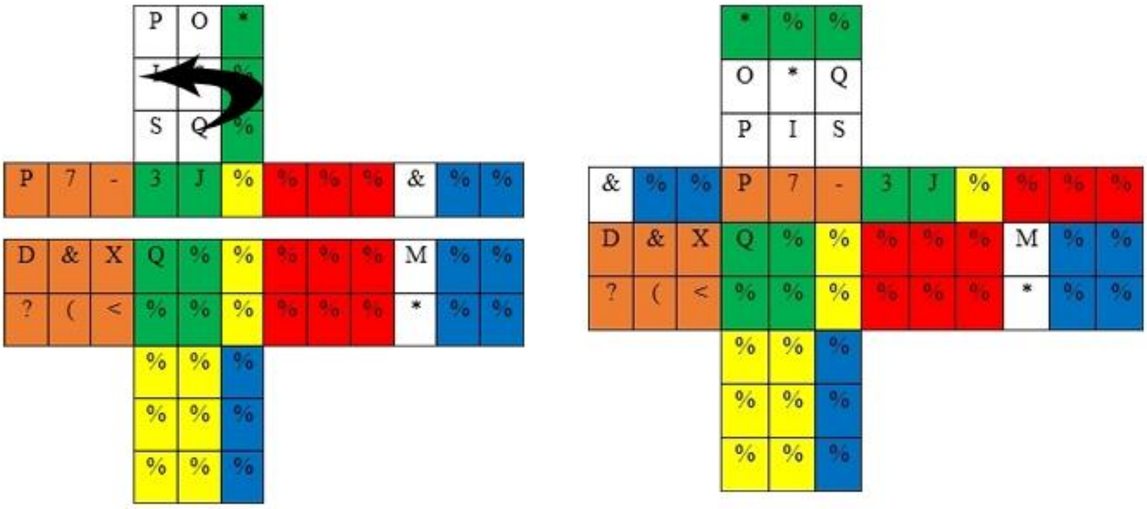


Figure 6. Illustration of the step 2

c. Step 3

The Rubik generated in the second step is rotated on the left by 90^0 clockwise (L). We have the Rubik's cube as described in Figure 7.

d. Step 4

The Rubik generated in the third step is rotated at the top by 90^0 clockwise (U). We have the Rubik's cube as described in Figure 8.

e. Step 5

The Rubik generated in the fourth step is rotated to the right by 90^0 counterclockwise (R'). We have the Rubik's cube as described in Figure 9.

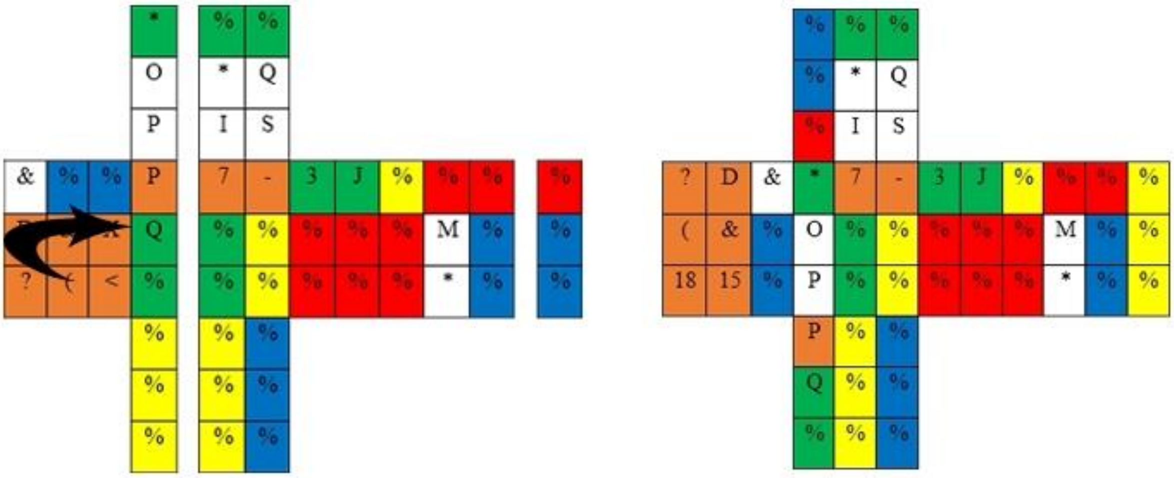


Figure 7. Illustration of the step 3

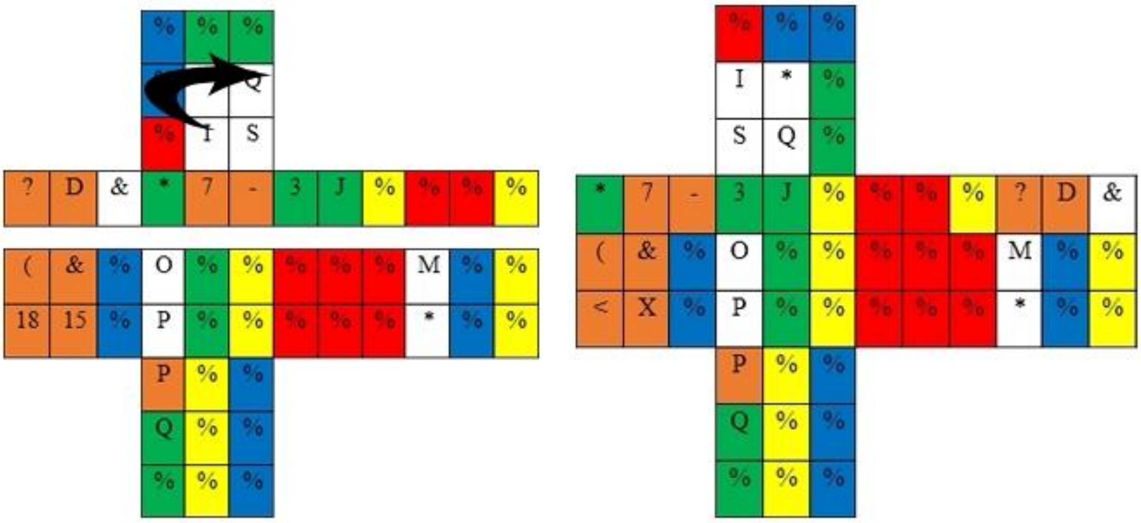


Figure 8. Illustration of the step 4

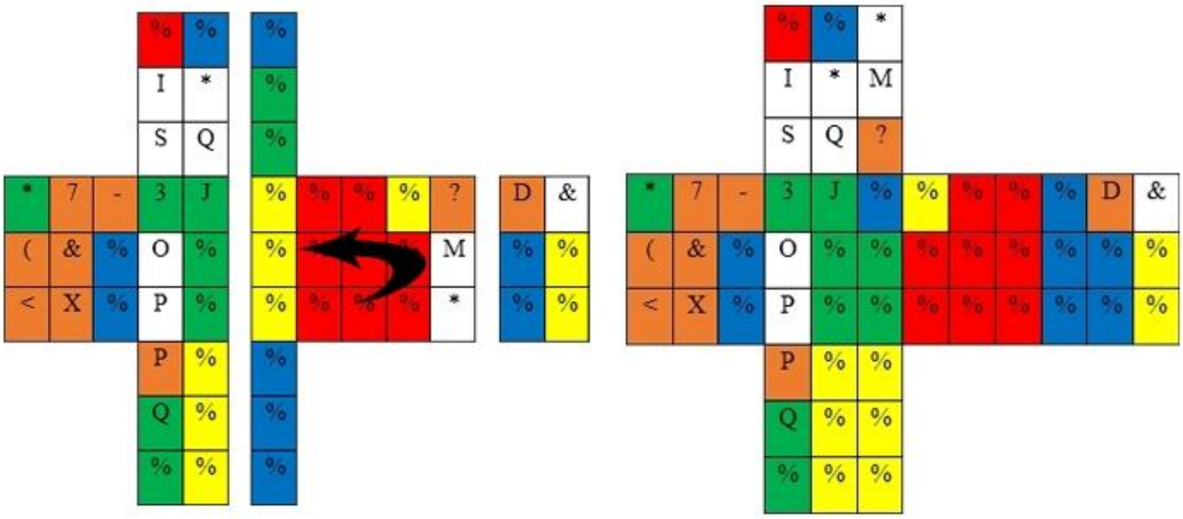


Figure 9. Illustration of the step 5

f. Step 6

The Rubik produced in the fifth step is rotated on the back by 90^0 clockwise (B) as illustrated in Figure 10.

10.12928/bamme.v1i2.4252

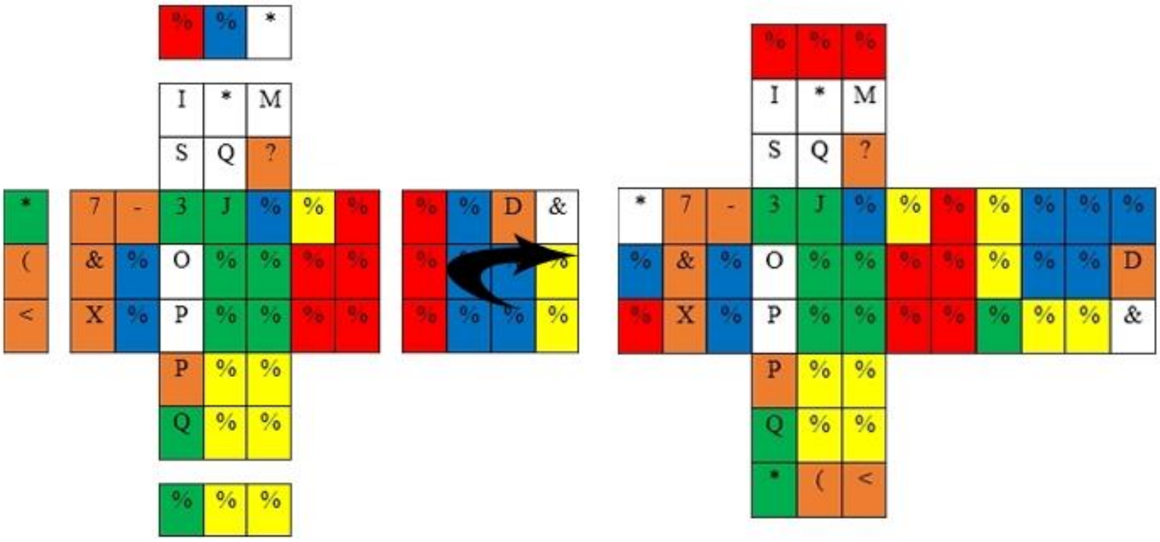


Figure 10. Illustration of the step 6

g. Step 7

The Rubik generated in the sixth step is rotated on the bottom by 90^0 clockwise (D) as illustrated in Figure 11.

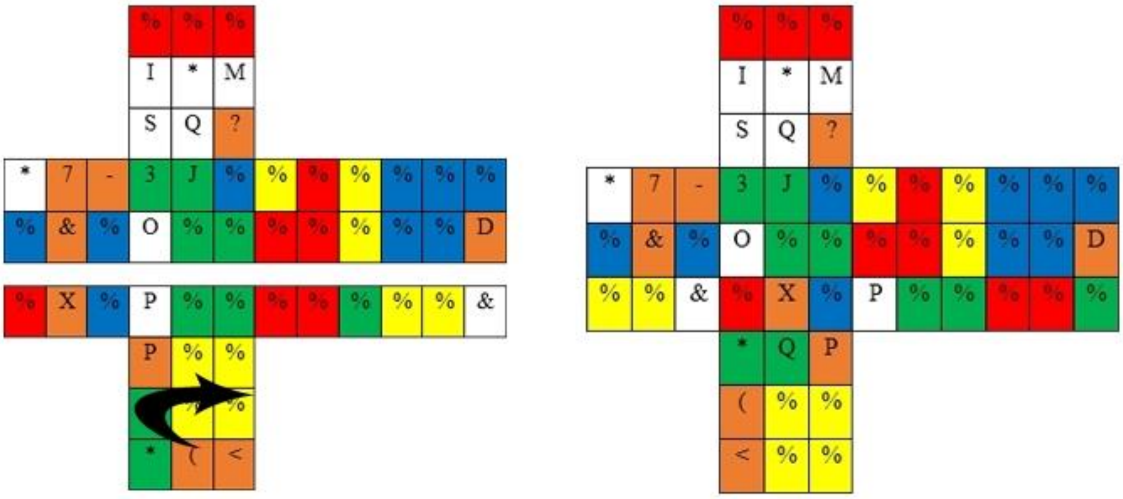


Figure 11. Illustration of the step 7

h. Step 8

The Rubik produced in the seventh step is rotated on the left by 90^0 counterclockwise (L') as illustrated in Figure 12.

i. Step 9

The Rubik's cube in the eighth step is rotated on the right by 90^0 counterclockwise (R') as illustrated in Figure 13.

j. Step 10

The Rubik generated in the ninth stage is rotated on the front by 90^0 counterclockwise (F') so that the encryption or ciphertext results are obtained as illustrated in Figure 14.

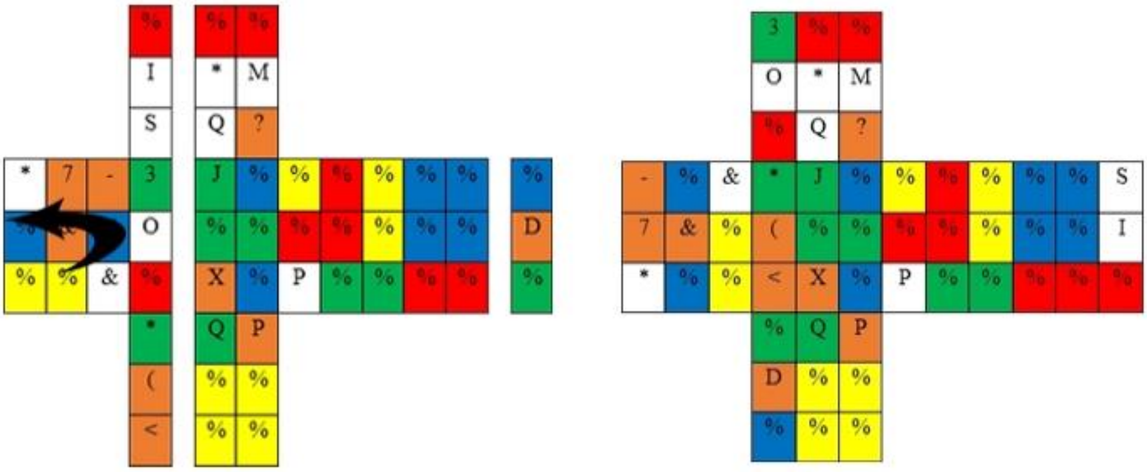


Figure 12. Illustration of the step 8

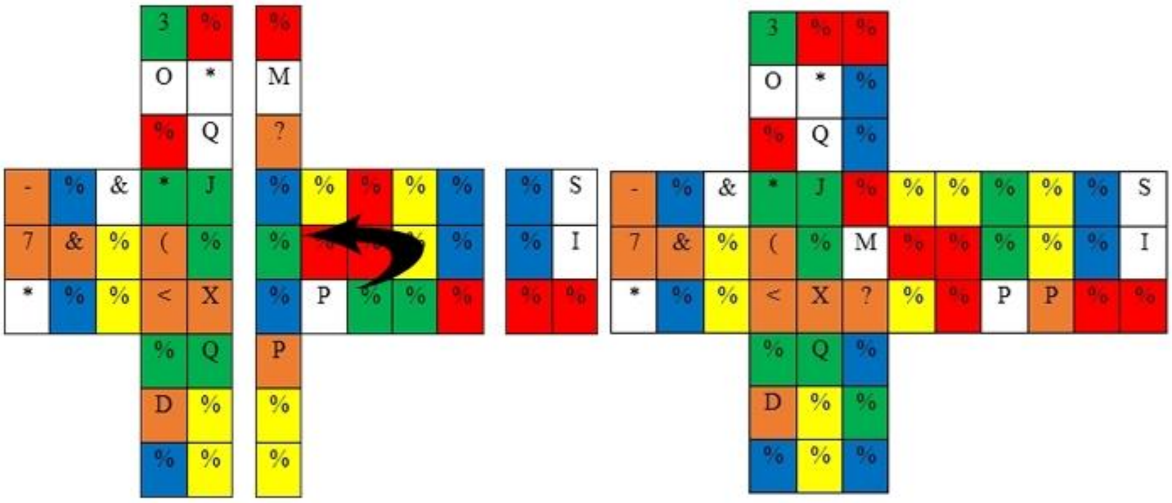


Figure 13. Illustration of the step 9

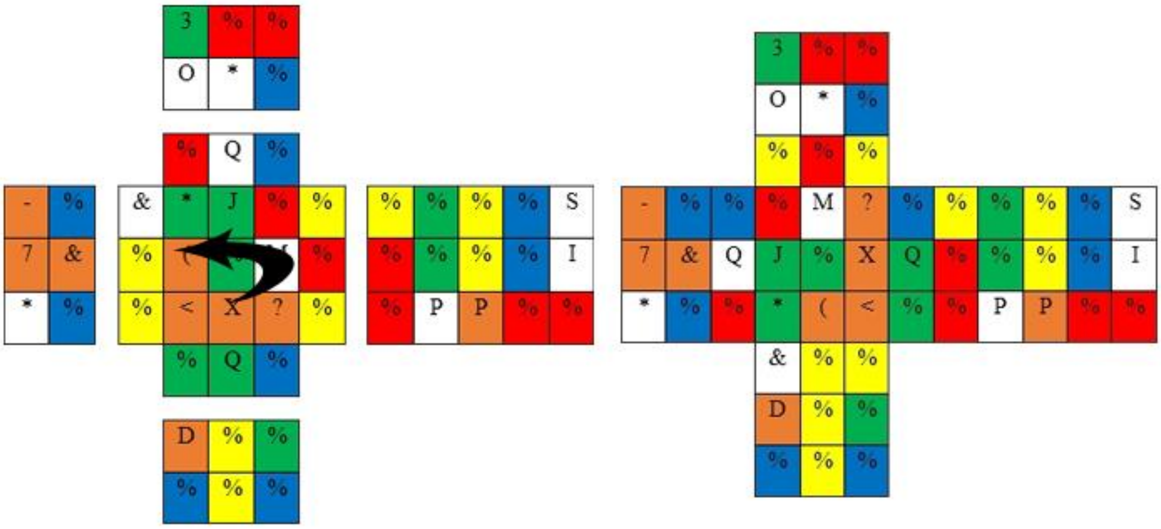


Figure 14. Illustration of the step 10

10.12928/bamme.v1i2.4252

It follows from the step 1 to 10 that the final chipertext of the initial plaintext “UNIVERSITAS AHMAD DAHLAN” is

“3\%\%O*\%\%\%\%\%\%7\&Q*\%\%\%M?J\%X*(<\%\%\%Q\%\%\%\%P\%\%S \%\%IP\%\%\&\%\%D\%\%\%\%\%”

Decryption Process

Decryption using Rubik’s Cube

The process of the decryption of the ciphertext using Rubik's cube is carried out in the following steps:

a. Form a rubik based on the determined key,
b. Put the ciphertext sequentially based on the determined order,
c. Determine the inverse of the key (the key is in the matix form),
d. Decryption process using the inverse of the key
e. Gain the plaintext (encrypted by the Hill chipher)

Key: R U’ L U R’ B D L’R’ F’

Inverse of the Key: F R L D’ B’ R U’ L’ U R’

Take the process analogously as illustrated in the encryption process so we have the following Rubik’s cube as described in Figure 15.

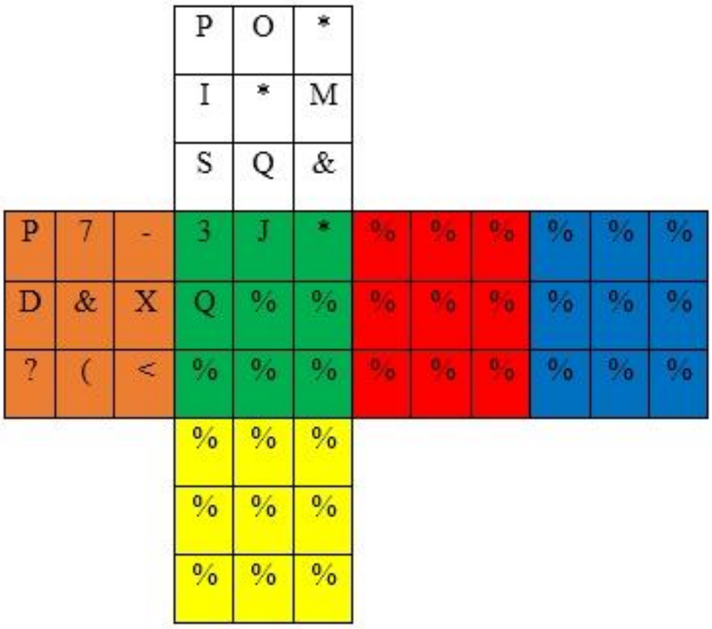


Figure 15. Illustration of the decryption process of Rubik’s cube

It follows from the decryption process that we will have the following plaintext

PO*I*MSQ\&P7-
D\&X?(<3J*Q\%
\%\%\%\%\%\%\%

Decryption process using the Hill cipher

The process of decryption is will be done analogously like the encryption process. However, the key used is the inverse of the key in the encryption process. Therefore, before we are going to decrypt the text, we have to determine the decrypted key first as follows

In fact, we have the matrix $K = \begin{bmatrix} 37 & 17 \\ 35 & 18 \end{bmatrix}$. Then the determinant of the matrix K is

$$\begin{vmatrix} 37 & 17 \\ 35 & 18 \end{vmatrix} = (37 \times 18) - (17 \times 35) = 666 - 595 = 71$$

Moreover, the inverse modulo 54 of the matrix K is

$$(71 \times a)\ mod\ 54 = 1$$

$$71^{-1}\ mod\ 54 = 35$$

$$K^{-1} = 35 \times \begin{bmatrix} 18 & -17 \\ -35 & 37 \end{bmatrix} mod\ 54 = \begin{bmatrix} 630 & -595 \\ -1225 & 1295 \end{bmatrix} mod\ 54 = \begin{bmatrix} 36 & 53 \\ 17 & 53 \end{bmatrix}$$

We therefore have the key for decryption processes : $K^{-1} = \begin{bmatrix} 36 & 53 \\ 17 & 53 \end{bmatrix}$.

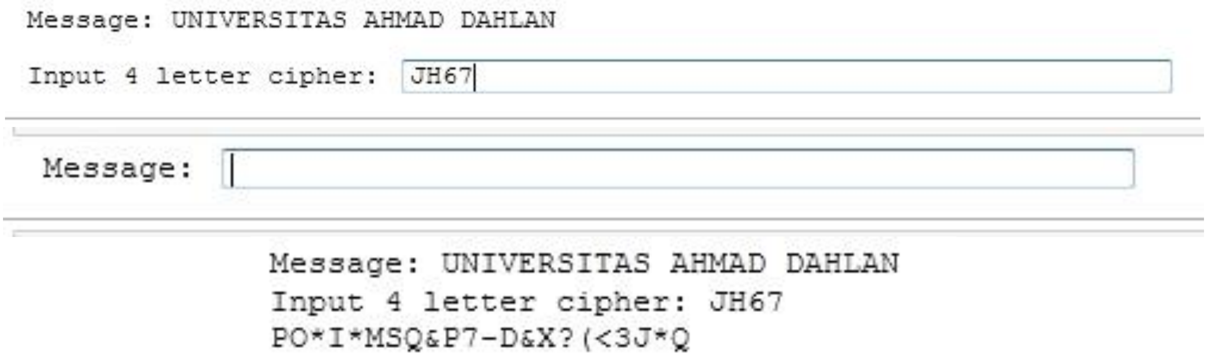
Decryption result

It follows from the decryption process that we will have the initial plaintext as follows "UNIVERSITASAHMADDAHLAN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%". Moreover, we can simply erase all of the character %, then we have "UNIVERSITAS AHMAD DAHLAN"

Encryption and Decryption Process Using Phyton

Encryption

The encryption process in Python is divided into 2 stages, namely encryption with Hill cipher and Rubik's cube. The encoding process with the Hill cipher method in Python is as shown in Figure 16.



```
Message: UNIVERSITAS AHMAD DAHLAN

Input 4 letter cipher: JH67

Message:

Message: UNIVERSITAS AHMAD DAHLAN
Input 4 letter cipher: JH67
PO*I*MSQ&P7-D&X?(<3J*Q
```

Figure 16. Encryption using Hill cipher using Phyton

In Figure 16, the initial plaintext "UNIVERSITAS AHMAD DAHLAN" will be encrypted. The encryption process begins by entering the sentence to be encrypted in the "Message" column, then entering a key that is relatively prime to 54 in the "Input 4 letter cipher" column, then we have the encrypted text or ciphertext:

PO*I*MSQ&P7-D&X?(< 3J*Q.

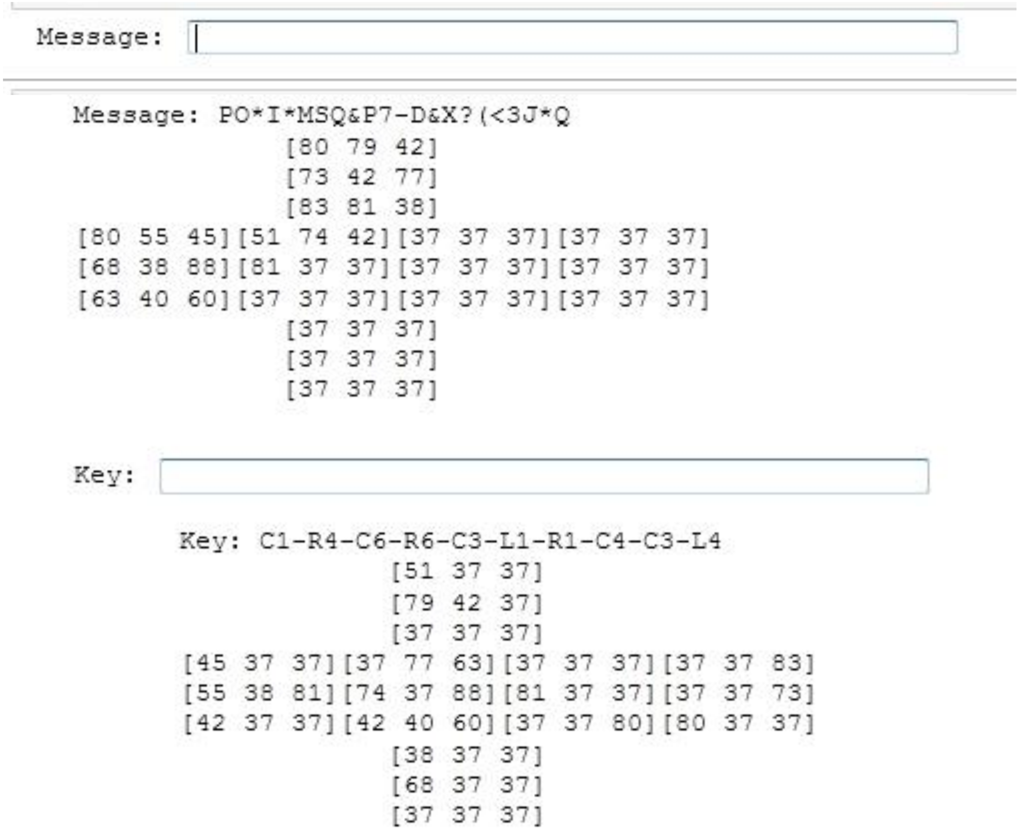
After the encryption process is fugured out using the Hill cipher method, then it will be encrypted with a Rubik's cube. In the process, adjustments are made to the rotation key used, as illustrated in Table 5.

10.12928/bamme.v1i2.4252

Table 5. The key rotation of rubik using Phyton

Key Rotation					
R = C1	L = C6	U = R6	D = R1	F = L6	B = L1
R’ = C3	L’ = C4	U’ = R4	D’ = R3	F’ = L4	B’ = L3
R2 =C2	L2 = C5	U2 = R5	D2 = R2	F2 = L5	B2 = L2

The key to be used is "R U' L U R' B D L' R' F'", then based on table 3, the key to be used in Python is "C1 R4 C6 R6 C3 L1 R1 C4 C3 L4". Hence, we will have the following appearance shown in Figure 17.



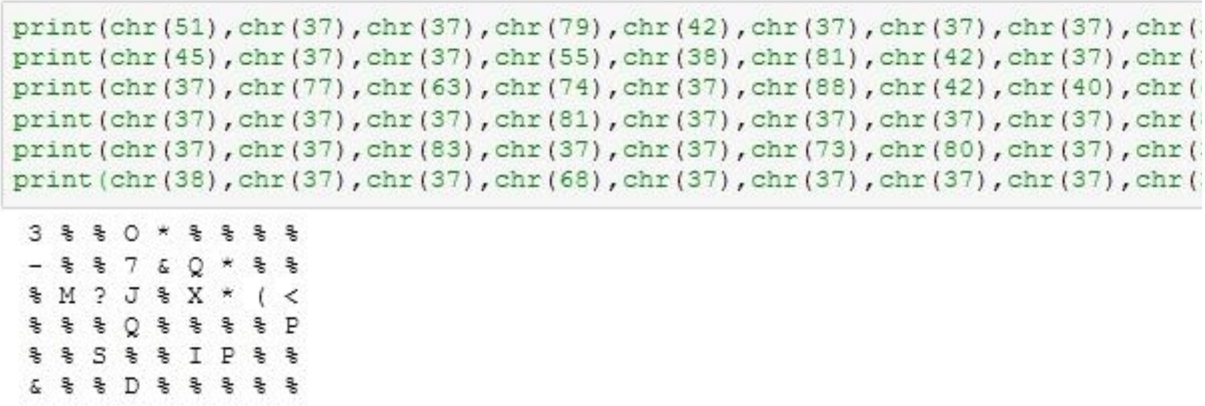
```
Message: |

Message: PO*I*MSQ&P7-D&X?(<3J*Q
                [80 79 42]
                [73 42 77]
                [83 81 38]
[80 55 45][51 74 42][37 37 37][37 37 37]
[68 38 88][81 37 37][37 37 37][37 37 37]
[63 40 60][37 37 37][37 37 37][37 37 37]
                [37 37 37]
                [37 37 37]
                [37 37 37]

Key:

Key: C1-R4-C6-R6-C3-L1-R1-C4-C3-L4
                [51 37 37]
                [79 42 37]
                [37 37 37]
[45 37 37][37 77 63][37 37 37][37 37 83]
[55 38 81][74 37 88][81 37 37][37 37 73]
[42 37 37][42 40 60][37 37 80][80 37 37]
                [38 37 37]
                [68 37 37]
                [37 37 37]
```

Figure 17. Encryption using Rubik’s Cube using Phyton

It follows from the encryption process that we have the following text

51 37 37 79 42 37 37 37 37 45 37 37 55 38 81 42 37 37 37 77 63 74 37 88 42 40 60 37 37 37 81 37 37 37 37 80 37 37 83 37 37 73 80 37 37 38 37 37 68 37 37 37 37 37

Moreover, the encryption result is converted using ASCII character standard. We therefore have the following as illustrated in Figure 18.



```
print(chr(51),chr(37),chr(37),chr(79),chr(42),chr(37),chr(37),chr(37),chr(
print(chr(45),chr(37),chr(37),chr(55),chr(38),chr(81),chr(42),chr(37),chr(
print(chr(37),chr(77),chr(63),chr(74),chr(37),chr(88),chr(42),chr(40),chr(
print(chr(37),chr(37),chr(37),chr(81),chr(37),chr(37),chr(37),chr(37),chr(
print(chr(37),chr(37),chr(83),chr(37),chr(37),chr(73),chr(80),chr(37),chr(
print(chr(38),chr(37),chr(37),chr(68),chr(37),chr(37),chr(37),chr(37),chr(
```

```
3 % % O * % % % %
- % % 7 & Q * % %
% M ? J % X * ( <
% % % Q % % % % P
% % S % % I P % %
& % % D % % % % %
```

Figure 18. ASCII character convertion

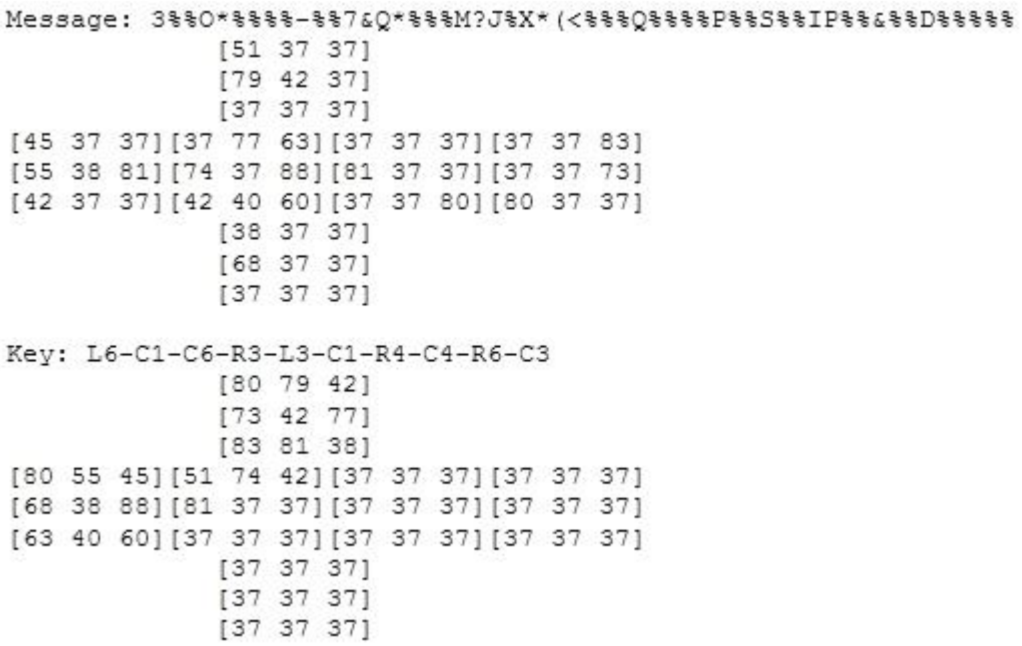
Finally, we have the following ciphertext
3%%O*%%%%%%%7&Q*%%%M?J%X*(<%%%Q%%%%P%%S%%IP%%&%%D%%%%%

Decryption

In the decryption process, the first thing to do is decrypt it with a Rubik's cube. The following plaintext,

3%%O*%%%%-%%7&Q*%%%M?J%X*(<%%%Q%%%%P%%
S%%IP%%&%%D %%%%%

will be decrypted. Then the key that we use is the inverse of the key in the previous encryption process. The inverse of the key is obtained as follows "FRLD' B' RU' L' U R'" which is then converted based on Table 3, we have "L6 C1 C6 R3 L3 C1 R4 C4 R6 C3". The process can be seen in the Figure 19.



```
Message: 3%%O*%%%%-%%7&Q*%%%M?J%X*(<%%%Q%%%%P%%S%%IP%%&%%D%%%%%
                [51 37 37]
                [79 42 37]
                [37 37 37]
[45 37 37][37 77 63][37 37 37][37 37 83]
[55 38 81][74 37 88][81 37 37][37 37 73]
[42 37 37][42 40 60][37 37 80][80 37 37]
                [38 37 37]
                [68 37 37]
                [37 37 37]

Key: L6-C1-C6-R3-L3-C1-R4-C4-R6-C3
                [80 79 42]
                [73 42 77]
                [83 81 38]
[80 55 45][51 74 42][37 37 37][37 37 37]
[68 38 88][81 37 37][37 37 37][37 37 37]
[63 40 60][37 37 37][37 37 37][37 37 37]
                [37 37 37]
                [37 37 37]
                [37 37 37]
```

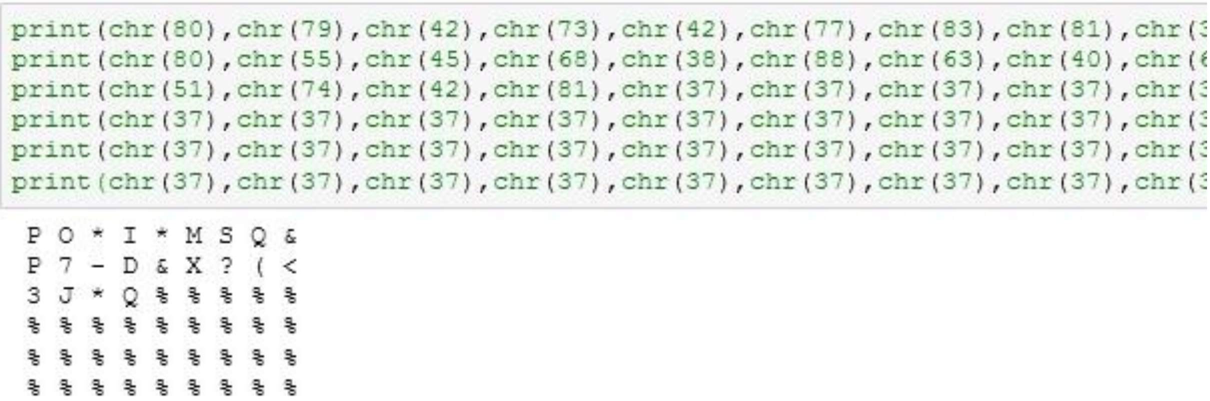
Figure 19. The decryption process using Rubik's cube with Phyton

10.12928/bamme.v1i2.4252

It follows from the decryption process that we have the following text

80 79 42 73 42 77 83 81 38 80 55 45 68 38 88 63 40 60 51 74 42 81 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37 37

Moreover, the text will be converted using the ASCII standard character. We have the following result shown in Figure 20.



```
print(chr(80),chr(79),chr(42),chr(73),chr(42),chr(77),chr(83),chr(81),chr(3
print(chr(80),chr(55),chr(45),chr(68),chr(38),chr(88),chr(63),chr(40),chr(6
print(chr(51),chr(74),chr(42),chr(81),chr(37),chr(37),chr(37),chr(37),chr(3
print(chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(3
print(chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(3
print(chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(37),chr(3
```

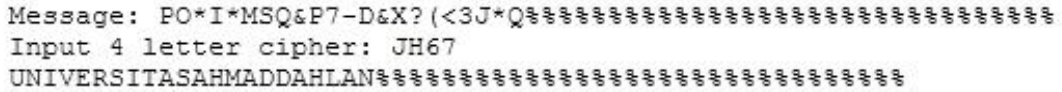
```
 P O * I * M S Q &
 P 7 - D & X ? ( <
 3 J * Q % % % % %
 % % % % % % % % %
 % % % % % % % % %
 % % % % % % % % %
```

Figure 20. Character convertion using ASCII strandard character

We therefore have

PO*I*MSQ&P7-D&X?(<3J*Q%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

After the decryption process using Rubik, then we will implement the Hill cipher method to decrypt the text. Hence, we have the following result shown in Figure 21.



```
Message: PO*I*MSQ&P7-D&X?(<3J*Q%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Input 4 letter cipher: JH67
UNIVERSITASAHMADDAHLAN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 21. Decryption process using Hill Cipher with Phyton

Conclusion

Based on the results and discussion in this paper, it can be concluded that the encryption process by combining the Hill cipher method and the Rubik's cube algorithm produces a more random and complicated ciphertext than if we implement only the one of them. The color arrangement of the Rubik's cube in the encryption results is also one of the keys so that if we want to decrypt the ciphertext with the general formula of the Rubik's cube without using the inverse key in the encryption process, we must previously know the position or color arrangement in order to put the ciphertext to be decrypted. As for using Python, it makes it easier and minimizes the processing steps or errors that can be obtained when using a manual process.

Acknowledgement

Authors express a gratitude to anypart who have helped to finish this research and paper.

References

Abitha, K. A., & Bharathan, P. K. (2016). Secure communication based on Rubik's cube algorithm and chaotic baker map. *Procedia Technology*, *24*, 782-789. https://doi.org/10.1016/j.protcy.2016.05.089

Danandjaja, J. (2014). Metode Penelitian Kepustakaan. *Antropologi Indonesia*, *0*(52) https://doi.org/10.7454/ai.v0i52.3318

Hraoui, S., Gmira, F., Abbou, M. F., Oulidi, A. J., & Jarjar, A. (2019). A New Cryptosystem of Color Image Using a Dinamic-Chaos Hill Cipher Algorithm. *Procedia Computer Science*, *148*, 399–408.

Indrajit, R. E. (2011). *Manajemen Keamanan Informasi dan Internet*. Kementerian Komunikasi dan Informatika RI.

Kalaichelvi, V., Manimozhi, K., Meenakshi, P., Rajakumar, B., & Vimaladevi, P. (2017). A new variant of Hill cipher algorithm for data security. *International Journal of Pure and Applied Mathematics*, *117*(15), 581-588.

Khuyen, D. (2016). *3x3x3 Rubik's Cube Simulator and Group Theory*. Centria University of Applied Sciences.

Mahmoud, A., & Chefranov, A. (2014). Hill cipher modification based on pseudo-random eigenvalues. *Applied Mathematics & Information Sciences*, *8*(2), 505. https://doi.org/10.12785/amis/080208

Raymond, T. (2005). *A Mathematical Approach to Solving Rubik's Cube*.

Schneier, B. (1996). Applied Cryptography. *Electrical Engineering*. https://doi.org/10.1.1.99.2838

Sinaga, M. C. (2017). *Kriptografi dan Python*.

Stinson, D. R. (2018). *Cryptography: Theory and Practice* (4th ed.). Chapman and Hall/CRC.