Modified Key Using Multi-Cycle Key In Vigenere Cipher

Ansar Rizal, Didi Susilo Budi Utomo, Rihartanto Rihartanto, Marselina Endah Hiswati, Haviluddin Haviluddin

Abstract— Encryption-Decryption is one form of securing text data, images and sound in order to minimize data stealing, attack, etc. The purpose of this study is to modified encryption-decryption keys of Vigenere cipher.Based on experiment show that multi-cycle keyencryption-decryption have been implemented. Meanwhile, the best MAE values of final encryption process of 56.837 and final decryption process of 0. In other words, the modification of key is more efficient and effective for data security. Furthermore, the multi-cycle key encryption-decryption optimizing using metaheuristics as one of the future works that can be conducted in order to investigate the best encryption-decryption especially using Vigenere cipher.

Keywords: Vigenere; encryption; decryption; multi-cycle; metaheuristics.

I. INTRODUCTION

Data security techniques are developed in order to minimize data stealing. Encryption is one form of securing text data, images and sound [1]. Then, data encryption is a process of changing the data into unreadable. So, the result of data encryption is cipher text. Meanwhile, in order to obtain information called decryption. Hence, a key is a main factor in encryption. In other words, the key is a heart of data security for decryption and encryption processes. Furthermore, an algorithm in order to process of encryption and decryption are divided into two types, namely symmetric and asymmetric keys. Symmetric is encryption and decryption process algorithm using the same key "one-key". Meanwhile, asymmetric key is an algorithm that uses two keys "different keys": private and public keys. The public key is the key for encryption. Then, private key is the key to decrypt cipher text. There are several types of symmetric algorithms including cipher (permutation and Substitution), cipher Hill, Caesar, Vigenere, OTP, RC6, Twofish, Magenta, FEAL, SAFER, LOCI, CAST, Rijndael (AES), Blowfish, GOST, A5, Kasumi, DES and IDEA, etc. [2]–[7].

Many studies using symmetric algorithm have been implemented. In order to improve its encryption and decryption performances especially in an improvement standard process of symmetric algorithm have also been carried out. For example, in Vigenere algorithm, by adding random bits (diffusion) before encryption on each byte [8], extend the Vigenere matrix [9], encryption repetitions [10], adding value to the plaintext[4], combining the LFSR key with Vigenere cipher key [6], using random table [7], varying the key [2] and more. Some researchers, such as [11] have been implemented Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Cuckoo Search (CS) algorithm as a crypt-analysis of Vigenere cipher using 26 characters. The results showed that GA and PSO techniques have recovered the entire key correctly for keys with small lengths. Then, CS technique has recovered more than 90% of key. In other words, CS showed that fast convergence and accuracy over PSO and GA.

In this study, Vigenere cipher as a development of the Caesar cipher algorithm especially on the filtering process will be applied. Meanwhile, the principle of Vigenere cipher algorithm is utilizing the Vigenere square for encryption. Classically, Vigenere cipher uses only 26 capital letters which starts from A to Z for encryption and decryption using "one-key". In other words, all characters like punctuations and numbers are unreadable. If the key has that characters (i.e., punctuations and numbers) so that it must be converted into capital letters. This is a weakness of Vigenere cipher algorithm which only use capital letters A-Z [5], [8], [9].

The aim of this study is to adjust the encryption-decryption keys in order to reduce the risk of weakness of Vigenere cipher algorithm using 128 ASCII characters. This paper is organized as follows. Research methodology is summarized in Section 2. Experimental results and discussions are given in Section 3, and Section 4 draws conclusions.

II. MATERIAL AND METHODS

2.1. Vigenere Cipher

In principle, a cryptographic algorithms can be classified into two types: classical and modern cryptographic [12]. Meanwhile, a Vigenere cipher method is a part of simple classical cryptographic. The Vigenere cipher is named after Blaise de Vigenere, a 16th century Frenchman Diplomat. Later, first described by Giovan Batista Belaso in 1553. The Vigenere cipher is a polyalphabetic cipher (letter transformed to different letters), depending upon its position in the plaintext, when encoded [3], [13].

Revised Version Manuscript Received on 10 September, 2019.

Ansar Rizal, Department of Information Technology, State Polytechnic of Samarinda, Indonesia

Didi Susilo Budi Utomo, Department of Information Technology, State Polytechnic of Samarinda, Indonesia

Rihartanto Rihartanto, Department of Information Technology, State Polytechnic of Samarinda, Indonesia (E-Mail: rihart.c@gmail.com)

Marselina Endah Hiswati, Faculty of Science and Technology, Respati University of Yogyakarta, Indonesia

Haviluddin Haviluddin, Faculty of Computer and Information Technology, Mulawarman University, Indonesia

2.2. ASCII Character

ASCII characters stands for American Standard Codes for International Interchange which is a collection of codes that used to interaction between user and computer. ASCII is universal international standard code (i.e., codes, letters, and symbols) with 0-127 range values which have 7 bits of binary composition. In this study, the implementation of Vigenere cipher algorithm using 128 ASCII Character. It aims to prevent the loss of some data on plaintext such as null, beep, Del, backspace, enter and others because of the filtering process. The ASCII characters can be seen in Table 1. Where, 0-31 and 127 are control characters, and 32-126 are printable characters.

Table 1. ASCII Table

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	|
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

asciichars.com

Source: http://www.asciichars.com

In this paper, short key modifications are used in order to be retained in the encrypted text. In this study, the encryption process using Eq. 3, and the decryption process using Eq. 4 are presented. These formulas are Vigenere cipher classical algorithm modifications, Eq. 1, 2.

$$C_i = (P_i + K_i) - 26 \quad (1)$$

$$P_i = (C_i - K_i) + 26 \quad (2)$$

$$C_i = (P_i + K_i)\ mod\ 128 \quad (3)$$

$$P_i = (C_i - K_i)\ mod\ 128 \quad (4)$$

As an illustrated. The original plaintextdata name is applied "Polytechnic" with key of "ab1d". Firstly, plaintext and key is changed into ASCII character using Eq. 3 and 4. The encryption-decryption processes of Vigenere cipher as an illustrated bellow.

Decryption

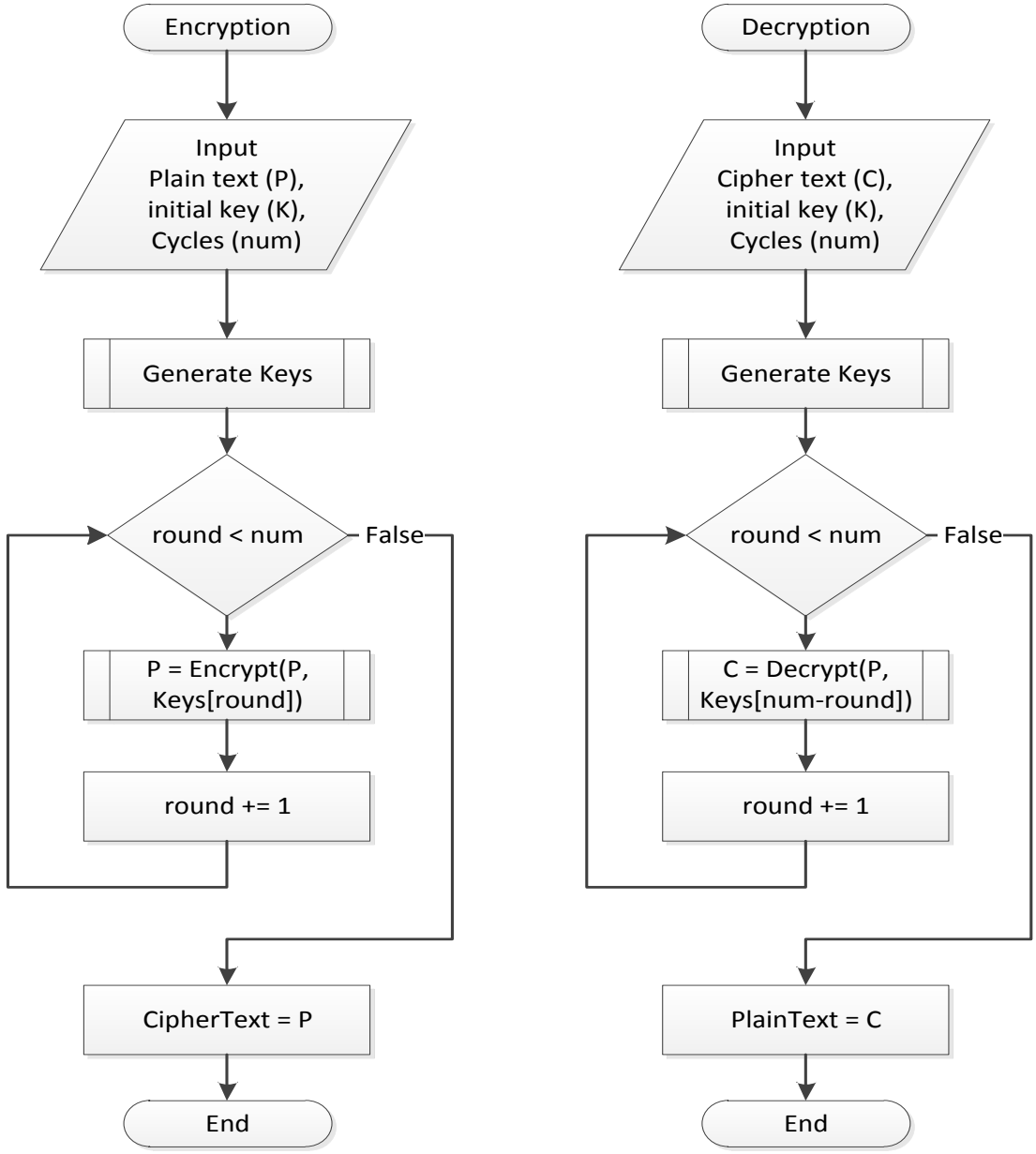
Ciphertext (C)	1	Q		]	U	G		L	O	K	
	49	81	29	93	85	71	20	76	79	75	20
Key (K)	a	b	1	d	a	b	1	d	a	b	1
	97	98	49	100	97	98	49	100	97	98	49
C-K mod 128	80	111	108	121	116	101	99	104	110	105	99
Plaintext (P)	P	o	l	y	t	e	c	h	n	i	c

Encryption

Plaintext (P)	P	o	l	y	t	e	c	h	n	i	c
	80	111	108	121	116	101	99	104	110	105	99
Key (K)	a	b	1	d	a	b	1	d	a	b	1
	97	98	49	100	97	98	49	100	97	98	49
P+K mod 128	49	81	29	93	85	71	20	76	79	75	20
Ciphertext (C)	1	Q		]	U	G		L	O	K	

2.3. Multi Cycle Encryption and Decryption Processes

In this study, multi-cycle encryption has been implemented. The goal is to improve the performance of Vigenere cipher without additional algorithm. Where, the keys for multi-cycle process have been created from the initial key using permutation. The flow of encryption-decryption multi-cycle process can be seen in Figure 1.



.

Figure 1. Flowchart of encryption-decryption multi-cycle in Vigenere cipher

In the encryption process, the ciphertext as output from the first cycle will become the plaintext for the second cycle, and so on until the last cycle. The same manner is applied to the decryption process.

2.4. Performance Measurement

In this study, the performance of Vigenere cipher algorithm is measured using Mean Absolute Error (MAE). Where, the MAE value is obtained from the difference original character and substitution distance positions. In this case, it is plaintext to ciphertext changed. The MAE formula using Eq. 5.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \qquad (4)$$

Where, n is the number of characters; y_i is the cipher character; x_i is the plain character.

III. RESULTS

In this experiment, the encryption-decryption program created using Pythonprogramming in order to run and analyze the Vigenere cipher algorithm was explored.The*itertools* module of python language for the permutation has been exploited. The keys from the initial key using permutation in order to produce unique keys which have the same length with the initial key then selected randomly have been generated. In order to ensure that the selected keys were exactly the same for the encryption and decryption process, a seed random was specified based on the given initial key. The seed random value was derived from the sum of the ASCII values multiplied by its position for each character contained in the initial key. It is assumed that the minimum initial key length was six characters

consisting of at least three different characters, the goal some combination of keys that can be selected for use in the encryption process were obtained.

The tests on five different plaintext using the same initial key were carried out. The initial key was “abc123” and the plaintext as shown in Table 2.

Table 2. The Plaintexts

Plaintext	# Chars	Description
Cryptography is about securing information.	43	Regular sentence
abcdef 123456 abcdef 123456 abcdef 123456	41	Two word segment,repeated for three times
abc123abc123abc123abc123abc123abc123	36	The initial key, repeated for six times
def4567def4567def4567def4567def4567	35	“def4567” one character longer than the initial key, repeated for five times
greengreengreengreengreengreengreen	35	“green” one character lesser than the initial key, repeated for seven times

For the processes encryption-decryption, three processes cycles of Vigenere cipher encryption-decryption have been presented. From the given initial key “abc123’ then the selected generated keys were “a123cb”, “1a2c3b” and “2bc31a”

Table 3. Result of Encryption

Cycle	Key	Plaintext	Ciphertext	MeanAbsolute Error (MAE)
The first plaintext				
1	a123cb	Cryptography is about securing information.	$#+#WQH##K[%SDDP&&SVG D&$QI RTN&RP	52.256
2	1a2c3b	$#+#WQH##K[%SDDP&&SVGD&$QI RTN&RP	U] 3yE~=2{W6w&X6)uV□+2{R|6□sX□2@	52.140
3	2bc31a	U] 3yE~=2{W6w&X6)uV□+2{R|6□sX□2@	f@9;+f(9/-d]:i(3i;i: 'i925d]5/61U;26r	56.837
The second plaintext				
1	a123cb	abcdef 123456 abcdef 123456 abcdef 123456	BHHbdfQFFFRdfhSDDDdfh	46.390
2	1a2c3b	BHHbdfQFFFRdfhSDDDdfh	stGz{*2CIJyH2Exy(wxGHwFG6 w&uvI|6uDEKL	27.805
3	2bc31a	stGz{*2CIJyH2Exy(wxGHwFG6w&uvI|6uDE KL	%V*-,d%y|{Zz(+*) ZgzyXx)}i('X,/gVv'{~}	55.780
The third plaintext				
1	a123cb	abc123abc123abc123abc123abc123abc123	BdBdBdBdBdBd	49.667
2	1a2c3b	BdBdBdBdBdBd	stGGHwstGGHwstGGHwstGGH wstGGHwstGGHw	29.333
3	2bc31a	stGGHwstGGHwstGGHwstGGHwstGGHwst GGHw	%V*zyX%V*zyX%V*zyX%V*z yX%V*zyX%V*zyX	51.667
The fourth plaintext				
1	a123cb	def4567def4567def4567def4567def4567	EghIgiHHfhjGGGegi	46.629
2	1a2c3b	EghIgiHHfhjGGGegi	vwJJKzIvI|JyHIH{|xGHz{*FGMz)xFLM	27.771
3	2bc31a	vwJJKzIvI|JyHIH{|xGHz{*FGMz)xFLM	(Y-}|[{X,/{Zz++.-Yy*~-, x)}+ *(|□~	54.543
The fifth plaintext				
1	a123cb	greengreengreengreengreengreengreen	H#QIS!JTF UGF%HGO$HPH#Q	54.686
2	1a2c3b	H#QIS!JTF UGF%HGO$HPH#Q	yI{+wI}6wwR})wK{)yV{{2yI{ +f,.56Y,7.)Y509	52.229
3	2bc31a	yI{+wI}6wwR})wK{)yV{{2yI{	)b.;, 2[9.,+f,.5	54.314

Table 4. Result of Decryption

Cycle	Key	Ciphertext	Deciphertext	Mean Absolute Error (MAE)
The first plaintext				
1	2bc31a	f@9;+f(9/-d]:i(3i;i: 'i925d]5/61U;26r	U] 3yE~=2{W6w&X6)uV□+2{R|6□sX□2@	52.140
2	1a2c3b	U] 3yE~=2{W6w&X6)uV□+2{R|6□sX□2@	$#+#WQH##K[%SDDP&&SVG D&$QI RTN&RP	52.256
3	a123cb	$#+#WQH##K[%SDDP&&SVGD&$QI RTN&RP	Cryptography is about securing information.	0.000
The second plaintext				
1	2bc31a	%V*-, d%y|{Zz(+*)ZgzyXx)}i('X,/gVv'{~}	stGz{*2CIJyH2Exy(wxGHwFG6 w&uvI|6uDEKL	27.805
2	1a2c3b	stGz{*2CIJyH2Exy(wxGHwFG6w&uvI|6uDE KL	BHHbdfQFFFRdfhSDDDdfh	46.390
3	a123cb	BHHbdfQFFFRdfhSDDDdfh	abcdef 123456 abcdef 123456 abcdef 123456	0.000
The third plaintext				
1	2bc31a	%V*zyX%V*zyX%V*zyX%V*zyX%V*zyX %V*zyX	stGGHwstGGHwstGGHwstGGH wstGGHwstGGHw	29.333
2	1a2c3b	stGGHwstGGHwstGGHwstGGHwstGGHwst GGHw	BdBdBdBdBdBd	49.667
3	a123cb	BdBdBdBdBdBd	abc123abc123abc123abc123abc12 3abc123	0.000
The fourth plaintext				
1	2bc31a	(Y-}|[{X,/{Zz++.-Yy*~-, x)}+ *(|□~	vwJJKzIvI|JyHIH{|xGHz{*FGMz)xFLM	27.771
2	1a2c3b	vwJJKzIvI|JyHIH{|xGHz{*FGMz)xFLM	EghIgiHHfhjGGGegi	46.629
3	a123cb	EghIgiHHfhjGGGegi	def4567def4567def4567def4567d ef4567	0.000
The fifth plaintext				
1	2bc31a	+f,.56Y,7.)Y509)b.;, 2[9.,+f,.5	yI{+wI}6wwR})wK{)yV{{2yI{	52.229
2	1a2c3b	yI{+wI}6wwR})wK{)yV{{2yI{	H#QIS!JTF UGF%HGO$HPH#Q	54.686
3	a123cb	H#QIS!JTF UGF%HGO$HPH#Q	greengreengreengreengreengreeng reen	0.000

In this experiment the performance of encryption-decryption using mean absolute error (MAE) have been used.The MAE value of encryption process vary depend on the plaintext. These MAE values from its original plaintext were measured. All MAE values were greater than 0, it mean that all cipher texts were different from its original. Then, the MAE value of final decryption of 0, means that the text decryption was the same as the original.

In this experiment, characteristic checking for cryptanalysis has also been performed. Then, the character check results that there were no duplicated characters that have same as key lengths have been showed. Therefore, the original character was unpredictable especially using the Kasiski or Friedman Test methods. The results of character repeating can be seen in Table 5 for the first plaintext which is a regular sentence. Since there weretoo many repeating characters for the second to the fifth plaintext, thus the summary its occurrence, maximum and minimum character in each cycle is shown in Table 6. It is interesting, for example in the second and the fourth plaintext, while in the beginning there were so many repeating characters in the original plaintext, then extremely reduce in the first cycle and no repeating at all in the second of the third cycle.

Table 5. Results of Characters Repeating Positionsfor the First Plaintext

Text	Characters	Length	Occurrences
Plaintext	"in"	2	2
	" i"	2	2
CipherText1	""	2	2
CipherText2	"2{"	2	2
CipherText3	"d]"	2	2

Table 6. Number of Occurrence Characters in Each Cycle

		2nd plain text	3rd plain text	4th plain text	5th plain text
Original text	# of words	182	87	91	70
	max chars	14	18	14	15
	min chars	2	2	2	2
1st Ciphertext (first cycle)	# of words	4	87	2	10
	max chars	2	18	2	5
	min chars	2	2	2	2
2nd Ciphertext (second cycle)	# of words	-	87	-	10
	max chars	-	18	-	5
	min chars	-	2	-	2
3rd Ciphertext (third cycle)	# of words	-	87	-	10
	max chars	-	18	-	5
	min chars	-	2	-	2

IV. DISCUSSION

In this experiment, the first plaintext using a regular expression have been used, while the other was a text constructed from a sequence of repetitive characters. In addition, for the same keys were used successively on each encryption cycle, and in descending order of each decryption cycle, Table 2.

Tables 3 and 4 show that every keys were used successively in each encryption cycle, and in reverse order in each decryption cycle. The first cycle encrypts the original plaintext. Then, in the second cycle was the first cycle output as plaintext has been used.

Then, the third cycle was the second cycle output as plaintext has been used, and so forth, while there were more cycles have been applied.

In contrast to [9] suggested that a one-time pad on a Vigenere cipher, where its length is as long as the key is true. Nevertheless, in this experiment, using short keys and which may require minimal cryptanalysis in each ciphertext of each cycle can be overcome by key modification.

V. CONCLUSION

This paper has presented the performance of classic Vigenerecipher implementing 128 ASCII characters with multi-cycle key encryption-decryption. The mean absolute error (MAE) are computed for each key encryption-decryption model and compared. Based on the results obtained, the multi-cycle key is found to be more efficient in encryption-decryption using Vigenere cipher algorithm. Optimizing the multi-cycle key encryption-decryption such as calculating of multi-cycles and key length as one of the future works that can be conducted in order to investigate the best encryption-decryption in Vigenere cipher

REFERENCES

1. A. Jawahir and H. Haviluddin, “An Audio Encryption Using Transposition Method,” Int. J. Adv. Intell. Informatics, vol. 1, no. 2, July 2015, pp. 98–106, 2015.
2. Q.-A. Kester, “A cryptosystem based on Vigenère cipher with varying key,” Int. J. Adv. Res. Comput. Eng. Technol., vol. 1, no. 10, pp. 108–113, 2012.
3. Q.-A. Kester, “A Hybrid Cryptosystem Based on Vigenère Cipher and Columnar Transposition Cipher,” Int. J. Adv. Technol. Eng. Res., vol. 3, no. 1, pp. 141–147, 2013.
4. A. A. Soofi, I. Riaz, and U. Rasheed, “An Enhanced Vigenere Cipher For Data Security,” Int. J. Sci. Technol. Res., vol. 5, no. 3, pp. 141–145, 2016.
5. R. S. Kartha and V. Paul, “Survey : Recent Modifications in Vigenere Cipher,” IOSR J. Comput. Eng., vol. 16, no. 2, pp. 49–53, 2014.
6. A. Razzaq, Y. Mahmood, F. Ahmed, and A. Hur, “Strong Key Machanism Generated by LFSR based Vigenère Cipher,” Int. Arab Conf. Inf. Technol., pp. 554–548, 2012.
7. Y. K. Singh, “Generalization of Vigenere cipher,” ARPN J. Eng. Appl. Sci., vol. 7, no. 1, pp. 39–44, 2012.
8. P. Wilson and M. Garcia, “A Modified Version of the Vigenère Algorithm,” Int. J. Comput. Sci. Netw. Secur., vol. 6, no. 3, pp. 140–143, 2006.

9. P. Ravindra, B. Kallam, S. U. Kumar, A. Vinaya, and V. Shravan, "A Contemporary Polyalphabetic Cipher using Comprehensive Vigenere Table," World Comput. Sci. Inf. Technol. J., vol. 1, no. 4, pp. 167–171, 2011.
10. R. S. Kartha and V. Paul, "A New Cryptosystem Based On Polyalphabetic Substitution Scheme With Multiple Number Of Cipher," in 6th IRF International Conference, 2014, pp. 40–44.
11. A. K. Bhateja, A. Bhateja, S. Chaudhury, and P. K. Saxena, "Cryptanalysis of Vigenere cipher using Cuckoo Search," Appl. Soft Comput. J., 2015.
12. H. Delfs, K. Paterson, and R. Cramer, Introduction to Cryptography: Principles and Application, Third Edit. Berlin: Springer-Verlag GmnH, 2015.
13. K. Senthil, K. Prasanthi, and R. Rajaram, "A modern avatar of Julius Ceasar and Vigenere cipher," 2013 IEEE Int. Conf. Comput. Intell. Comput. Res. IEEE ICCIC 2013, pp. 13–15, 2013.