International Journal of Security and Its Applications
Vol. 10, No. 8 (2016), pp.173-180
http://dx.doi.org/10.14257/ijsia.2016.10.8.15

Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm

Andysah Putera, Utama Siahaan[1] and Robbi Rahim[2]

[1]Faculty of Computer Science, [2]Faculty of Information System
[1]Universitas Pembangunan Panca Budi, [2]Universitas Prima Indonesia
[1]Jl. Jend. Gatot Subroto Km. 4,5 Sei Sikambing, 20122, Medan, Sumatera Utara, Indonesia
[2]Jl. Sekip Simpang Sikambing, Medan, Sumatera Utara, Indonesia
[1]andiesiahaan@gmail.com, [2]usurobbi85@zoho.com

Abstract

Genetic algorithms can solve complex problems, including the problems of cryptography. What problems often occur on the Hill Cipher is the waste of time to determine the numbers that are used in the encryption process. In the encryption process, it is not a problem if the key is derived from any number. However, the problem is ciphertext cannot be returned to the original message. The key that is used must have the determinant is 1. To find the value of it is something that takes time if it must be done manually. Due to the entered value to the Hill Cipher is random, Genetic algorithms can be used to optimize the search time. By using this algorithm, the determinant calculation will be more accurate and faster. The result achieved is the program can specify some combination of numbers that can be used as the encryption key Hill Cipher and it can reject the unnecessary numbers.

Keywords*: Cryptography, Genetic Algorithm, Hill Cipher*

1. Introduction

In the digital network, multimedia is a way to transfer information [2]. The confidential information must be secure using a cryptography method. Hill Cipher is a symmetric key cryptography algorithm using a n x n matrix [4-7]. It is used a set of numbers inserted into the matrix as a key. In this key, there are nine pieces utilized random integers that set a matrix of 3x3 [1,20]. Each number will be associated with each other to generate the ciphertext, but not all the number can be used to restore the original messages. The numbers must have the exact value of the determinant. Before the numbers could be utilized, there is a test to ensure it has the correct determinant [8]. The test itself takes time meanwhile these numbers which make up the determinant correct is not necessarily obtained. If the result is wrong, the search of random integers has to be repeated, and it takes a long time. The problem that arises is an inefficient time if the key on Hill Cipher algorithm is performed manually. Generating keys on Hill Cipher algorithm by combining Genetic algorithms are supposed to speed up the search for the suitable key for the Hill Cipher encryption [17].

2. Related Work

In the earlier research, Genetic algorithms use an image to perform a secret key encryption. A new technique of this algorithm is to utilize the crossover and mutation method to produce the encryption [3]. This method is implemented to data images. Some results were applied in 1993. Spillman did the cryptanalysis technique based on a genetic approach. It was implemented on the substitution cipher [14]. It tried to discover the key by using possible random search. It is done to a simple plaintext. In 1993, Mathew did the

ISSN: 1738-9976 IJSIA

same way using genetic algorithms to a transposition cipher. At that time, Spillman also applied the genetic evolution to a knapsack problem. Garg also described the Genetic algorithm efficiency in attacking knapsack [15]. It was told that the method could be developed with an improvisation of the parameters. The statement was declared in 2006. Garg stated that using the memetic mutation can break a simplified data encryption. Nalini also compared the difference of heuristic and Genetic technique. It concluded the Genetic algorithm reduced the time complexity [16].

The previous research was proposed by Dr. Geetha. It is based on Knapsack problem. The NP-hard problem attack is various to the performance optimization. It offered the robustness against the attack of the known plain text. It is used an 8-bit ASCII character encoding of plaintext. Super increasing sequence is converted into non-super increasing using the values W, M. Non-super increasing sequence of knapsack cipher is more difficult to break and hence it is used as the public key. The public key is only available for attack. The ciphertext is calculated from the non-super increasing sequence. If the non-super increasing sequence is {21033, 63094, 16375, 11711, 23422, 58557, 16665, 54322, 64252, 39720, 32718, 63106, 63119, 18753, 21135, 42270} then the ciphertext for the plaintext "QUICK" can be calculated as shown in Table 1 [18].

Table 1. Ciphertext Calculation

Plaintext	Unicode (in binary)	Ciphertext
Q	0000000001010001	145096
U	0000000001010101	163849
I	0000000001001001	145109
C	0000000001000011	103125
K	0000000001001011	166244

Genetic algorithm attacks the ciphertext. It attacks the Knapsack cipher using the genetic algorithm. It showed the analysis on the effect of various genetic control and initial population size. It also analyzed the operator probabilities and selection process. In following subsections Genetic Algorithm process, Individual representation, Initial population, Fitness evaluation, Termination condition, Selection methods, Crossover and Mutation are explained [18].

In 2015, Ali said that a Genetic algorithm is a search tool that's used to ensure the high probability of finding a solution by decreasing the amount of time in key space searching. The focus was on the cryptanalysis of a Hill cipher by using Genetic algorithms and study the algorithm factors that affect finding solution taking into consideration the time and efficiency of the cryptanalysis process, the different type of crossover, population size and mutation rate that used to find the optimal solution [19].

3. Proposed Work

At this section, it describes the role of Genetic algorithms in Hill Cipher encryption. In Hill Cipher of a 3 x 3 matrix, there are three rows and three columns which expanded into nine chromosomes in Genetic algorithms. It is filled with a random integer number respectively. It is in range of 0 to 255 (ASCII Code).

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Figure 1. Hill Cipher Key Matrix

$$\begin{matrix}[a & b & c & d & e & f & g & h & i] \\ [c1 & c2 & c3 & c4 & c5 & c6 & c7 & c8 & c9]\end{matrix}$$

Figure 2. Genetic Chromosome

Figure 1 and Figure 2 are identical. The first picture is the Hill Cipher matrix. It still consists of rows and columns. The process of Genetic algorithm needs one dimension vector for its data. The matrix must be converted into a single line. The caption "a to i" are the chromosome numbers. Each chromosome has one gene which value is random. "a" = "c1". It means the number represented by "a" refers to the chromosome "c1" and so further until "i" = "c9".

$$F = \big((\mathrm{a} * \mathrm{e} * \mathrm{i}) + (\mathrm{b} * \mathrm{f} * \mathrm{g}) + (\mathrm{c} * \mathrm{d} * \mathrm{h})\big) - \big((\mathrm{a} * \mathrm{f} * \mathrm{h}) + (\mathrm{b} * \mathrm{d} * \mathrm{i}) + (\mathrm{c} * \mathrm{e} * \mathrm{g})\big) \quad (1)$$

There is an objective function to calculate the fitness. It determines whether the numbers can be inserted into the matrix. The determinant of Hill Cipher works as a condition when it stops calculating. The value desired is 1 but sometimes it cannot reach and must be re-calculated again. Formula 1 is how to obtain the fitness (F) in Genetic algorithm.

Genetic algorithms try to find the appropriate fitness [9, 10]. It forms the initial population with the specific parameters. Generations and populations take a role in the speed of searching. Those parameters are adjustable according to users, and the result will be variant in every single process [11-13]. It calculates the fitness and compares the result with the value of 1. After such process is accomplished, it generates all the key pair which value is 1. There will be a probability that the numbers obtained more than one key pair. Selection, Crossover, and Mutation are also applied to next generation to remove the unnecessary results. After the last repetition, Genetic algorithms show several 3 x 3 matrix numbers for further use.

4. Evaluation

The initial population needs to be generated by the program before calculating the fitness. It is derived from any numbers in a range of 0 to 255 (ASCII Code). There are nine chromosomes to be filled from "a" to "i". This initial population sets Population = 20. The generation will stop when the fitness is reached. There are two types of termination, loop until the condition is met or loop according to the generation determined earlier. Tabel 1 shows the population generated by the computer program.

Table 1. Generated Population

Pop	a	b	c	d	e	f	g	h	i
1	65	109	34	95	40	136	100	44	173
2	14	191	220	112	179	122	87	156	177
3	85	150	192	247	206	50	237	91	1
4	86	213	205	7	4	149	95	199	156
5	218	98	118	156	37	57	92	140	22
6	174	68	146	86	17	24	144	186	252
7	58	73	158	239	113	204	136	15	85
8	223	8	188	72	17	235	220	123	58

9	60	184	113	136	155	66	29	64	23
10	110	254	196	181	118	72	151	38	127
11	253	234	240	169	20	22	226	123	133
12	89	102	52	166	120	148	221	84	3
13	140	177	172	93	210	49	181	27	16
14	143	42	144	116	175	234	109	222	143
15	73	146	164	91	164	32	187	194	216
16	226	199	135	49	215	34	69	12	116
17	209	160	24	191	242	248	7	83	82
18	243	227	182	72	58	152	5	148	193
19	94	49	13	140	160	17	88	210	82
20	135	95	82	111	48	164	119	154	63

Every chromosome must be calculated. Fitness, Probability, and Cumulative Probability are the parameters. Fitness it to determine the numbers for the final decision. In Table 2, there are many numbers showed in F, P, and CP columns but none of the fitness values demonstrate the value of 1. The further process needs to be performed. It continues to the next generation. This process are looped until it reach the correct fitness (Fitness = 1).

$$
\begin{aligned}
F(1) &= ((65 * 40 * 173) + (109 * 136 * 100) + (34 * 95 * 44)) - \\
&\quad ((65 * 136 * 44) + (109 * 95 * 173) + (34 * 40 * 100)) \\
&= -242055 \\
&= 256 - (\mathrm{ABS}(-240255) \bmod 256) \\
&= 256 - 135 \\
&= 121
\end{aligned}
$$

Table 2. Fitness, Probability, and Cumulative Probability

Pop	F	P	CP
1	121	0,048477564	0,048477564
2	72	0,028846154	0,077323718
3	138	0,055288462	0,132612179
4	194	0,077724359	0,210336538
5	52	0,020833333	0,231169872
6	160	0,064102564	0,295272436
7	245	0,098157051	0,393429487
8	31	0,012419872	0,405849359
9	221	0,088541667	0,494391026
10	98	0,039262821	0,533653846
11	48	0,019230769	0,552884615
12	68	0,02724359	0,580128205
13	93	0,037259615	0,617387821
14	231	0,092548077	0,709935897
15	184	0,073717949	0,783653846
16	81	0,032451923	0,816105769
17	132	0,052884615	0,868990385

18	146	0,05849359	0,927483974
19	156	0,0625	0,989983974
20	25	0,010016026	1

The value obtained from population number 1 is 121. The number does not fit the fitness desired. Population 1 to 20 is resulting the incorrect number. However, the next generation needs to start to recalculate the fitness again. The genetic process will stop after it finds the appropriate key. Table 3 illustrates the keys generated by the genetic process.

Table 3. Key Pairs

No.	a	b	c	d	e	f	g	h	i
1	45	45	192	202	65	159	45	103	45
2	6	153	11	225	225	83	153	176	229
3	19	20	6	20	19	20	11	41	83

Key(1) = ((45 * 65 * 45) + (45 * 159 * 45) + (192 * 202 * 103)) –
((45 * 159 * 103) + (45 * 202 * 45) + (192 * 65 * 45))
= 2740737
= 256 – (ABS(2740737) MOD 256)
= 256 – 255
= 1

Key(2) = ((6 * 225 * 229) + (153 * 83 * 153) + (11 * 225 * 176)) –
((6 * 83 * 176) + (153 * 225 * 229) + (11 * 225 * 153))
= - 5661951
= 256 – (ABS(-5661951) MOD 256)
= 256 – 255
= 1

Key(3) = ((19 * 19 * 83) + (20 * 20 * 11) + (6 * 20 * 41)) –
((19 * 20 * 41) + (20 * 20 * 83) + (6 * 19 * 11))
= - 10751
= 256 – (ABS(-10751) MOD 256)
= 256 – 255
= 1

Key 1, 2 and 3 are the results of the correct fitness. The calculations show the determinant is 1. Since it is correct, it can be used for the encryption and decryption. Sometimes it produces either positive or negative value, but it is must be normalized by using the modular expression. The determinant in Table 3 are in negative value, and those values have been normalized.

$$\begin{bmatrix} 45 & 45 & 192 \\ 202 & 65 & 159 \\ 45 & 103 & 45 \end{bmatrix} \begin{bmatrix} 6 & 153 & 11 \\ 225 & 225 & 83 \\ 153 & 176 & 229 \end{bmatrix} \begin{bmatrix} 19 & 20 & 6 \\ 20 & 19 & 20 \\ 11 & 41 & 83 \end{bmatrix}$$

(a) (b) (c)

Figure 3. Key Produced by Genetic Algorithms (a) Key 1, (bv) Key 2, (c) Key 3

In Figure 3, (a), (b), and (c) can be used for the Hill Cipher. When using the manual process, it is a very hard way to obtain these three key pairs. It is difficult to calculate the numbers one by one. Genetic algorithms take role in this situation. It generates the number quicker than the conventional way.

The following calculation will prove the correctness of key number 1. Assume the plaintext is "ABC" where the ASCII Code is [65-67].

Encryption Process:

$$\begin{bmatrix} 45 & 45 & 192 \\ 202 & 65 & 159 \\ 45 & 103 & 45 \end{bmatrix} * \begin{bmatrix} 65 \\ 66 \\ 67 \end{bmatrix}$$

$$\begin{aligned} C1 &= \big((45 * 65) + (45 * 66) + (192 * 67)\big)\ mod\ 256 \\ &= 18759\ mod\ 256 \\ &= 71 \end{aligned}$$

$$\begin{aligned} C2 &= \big((202 * 65) + (65 * 66) + (159 * 67)\big) mod\ 256 \\ &= 28073\ mod\ 256 \\ &= 169 \end{aligned}$$

$$\begin{aligned} C3 &= \big((45 * 65) + (103 * 66) + (45 * 67)\big)\ mod\ 256 \\ &= 12738\ mod\ 256 \\ &= 194 \end{aligned}$$

Decryption Process:

$$\begin{bmatrix} 116 & 87 & 51 \\ 113 & 41 & 141 \\ 217 & 206 & 235 \end{bmatrix} * \begin{bmatrix} 71 \\ 169 \\ 194 \end{bmatrix}$$

$$\begin{aligned} P1 &= (116 * 71) + (87 * 169) + (51 * 194)\ mod\ 256 \\ &= 32833\ mod\ 256 \\ &= 65 \end{aligned}$$

$$\begin{aligned} P2 &= (113 * 71) + (41 * 169) + (141 * 194)\ mod\ 256 \\ &= 42306\ mod\ 256 \\ &= 66 \end{aligned}$$

$$\begin{aligned} P3 &= (217 * 71) + (206 * 169) + (235 * 194)\ mod\ 256 \\ &= 95811\ mod\ 256 \\ &= 67 \end{aligned}$$

The key is implemented in the encryption process while in the decryption uses the inverted key. Using the correct key pair will produce the correct result as well. C1, C2, and C3 described the process of the encryption using the key selected. The numbers [65-67] are processed into cipher numbers [71, 169, 94]. The decryption process will return the numbers back to [65-67].

6. Conclusion

Hill Cipher 3 x 3 uses nine cells to build the key matrix. The key can be derived from any number from 0 to 255. The key is unique. It should have the determinant value is 1. However, to find the correct determinant is hard. The searching of the key pairs in Hill

Cipher is wasting time if done manually. Genetic algorithms can help the working time. It minimizes the cost of time from being calculating the correct fitness. It can provide the alternative numbers by selecting the total amount desired.

References

[1] A. P. U. Siahaan, "Genetic Algorithm in Hill Cipher Encryption", International Association of Scientific Innovation and Research (IASIR), vol. 15, no. 1, **(2016)**.
[2] A. P. U. Siahaan, "RC4 Technique in Visual Cryptography RGB Image Encryption", International Journal of Computer Science and Engineering (IJCSE), vol. 3, no. 7, **(2016)**, pp. 1-6.
[3] A. A. Abdullah, R. Khalaf dan and M. Riza, "A Realizable Quantum Three-Pass Protocol Authentication", Mathematical Problems in Engineering, **(2015)**.
[4] R. Kumar dan R. C., "Analysis of Diffie Hellman Key Exchange Algorithm with Proposed Key Exchange Algorithm", International Journal of Emerging Trends & Technology in Computer Science, vol. 4, no. 1, **(2015)**, pp. 40-43.
[5] M. Ahmed, B. Sanja, D. Aldiaz, A. Rezaei dan and H. Omotunde, "Diffie-Hellman and Its Application in Security Protocols", International Journal of Engineering Science and Innovative Technology, vol. 1, no. 2, **(2008)**, pp. 69-73.
[6] M. N. A. Rahman, A. F. A. Abidin, M. K. Yusof dan and N. S. M. Usop, "Cryptography: A New Approach of Classical Hill Cipher", International Journal of Security and Its Applications, vol. 7, no. 2, **(2013)**, pp. 179-190.
[7] S. I. Chowdhury, S. A. M. Shohag dan and H. Sahid, "A Secured Message Transaction Approach by Dynamic Hill Cipher Generation and Digest Concatenation", International Journal of Computer Applications, vol. 23, no. 9, **(2011)**, pp. 25-31.
[8] A. A. Khalaf, M. S. A. El-karim dan and H. F. A. Hamed, "A Triple Hill Cipher Algorithm Proposed to Increase the Security of Encrypted Binary Data and its Implementation Using FPGA", ICACT Transactions on Advanced Communications Technology, vol. 5, no. 1, **(2016)**, pp. 752-757.
[9] E. Heidari dan and A. Movaghar, "An Efficient Method Based On Genetic Algorithms To Solve Sensor Network Optimization Problem", International journal on applications of graph theory in wireless ad hoc networks and sensor networks, vol. 3, no. 1, **(2011)**, pp. 18-33.
[10] A. G. Karegowda, A. Manjunath dan and M. Jayaram, "Application Of Genetic Algorithm Optimized Neural Network Connection Weights For Medical Diagnosis Of Pima Indians Diabetes", International Journal on Soft Computing, vol. 2, no. 2, **(2011)**, pp. 15-23.
[11] C. H. Lin, J. L. Yu, J. C. Liu, W. S. Lai dan and C. H. Ho, "Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems", International Journal of Hybrid Information Technology, vol. 2, no. 1, **(2009)**, pp. 21-30.
[12] A. A. Ghanbari, A. Broumandnia, H. Navidi dan and A. Ahmadi, "Brain Computer Interface with Genetic Algorithm", International Journal of Information and Communication Technology Research, vol. 2, no. 1, **(2012)**, pp. 79-86.
[13] S. Szénási dan and Z. Vámossy, "Implementation of a Distributed Genetic Algorithm for Parameter Optimization in a Cell Nuclei Detection Project", Acta Polytechnica Hungarica, vol. 10, no. 4, **(2013)**, pp. 59-86.
[14] S. R., J. M., N. B. dan K. N., "Use of Genetic Algorithm in Cryptanalysis of Simple Substituion Cipher", Cryptologia,, vol. 17, no. 4, **(1993)**, pp. 367-377.
[15] G. Poonam, "Genetic algorithm Attack on Simplified Data Encryption Standard Algorithm", International journal Research in Computing Science, **(2006)**.
[16] Nalini, "Cryptanalysis of Simplified Data Encryption Standard via Optimization Heuristics", International Journal of Computer Sciences and Network Security, vol. 6, no. 1, **(2006)**.
[17] A. Agarwal, "Secret Key Encryption Algorithm Using Genetic Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 4, **(2012)**, pp. 216-218.
[18] R. G. Ramani dan and L. Balasubramanian, "Genetic Algorithm solution for Cryptanalysis of Knapsack Cipher with Knapsack Sequence of Size 16", International Journal of Computer Applications, vol. 35, no. 11, **(2011)**, pp. 18-23.
[19] A. S. Alkhalid, "Cryptanalysis of a Hill Cipher Using Genetic Algorithm", dalam IEEE, Hammamet, **(2015)**.
[20] A. P. U. Siahaan, "Three-Pass Protocol Concept in Hill Cipher Encryption Technique", International Journal of Science and Research (IJSR), vol. 5, no. 7, **(2016)**, pp. 1149-1152.

Authors

Andysah Putera Utama Siahaan, He was born in 1980, Medan, Indonesia. He received the S.Kom. degree in computer science from Universitas Pembangunan Panca Budi, Medan, Indonesia, in 2010, and in 2012, he obtained M.Kom. from the University of Sumatera Utara, Medan, Indonesia. In 2010, he joined as a lecturer at the Department of Engineering, Universitas Pembangunan Panca Budi. He has been a researcher since 2012. He has studied his Ph. D. degree from 2016. He is now active in writing international journals and conferences.

Robbi Rahim, He was born in Medan, Indonesia, in 1985. He received the S.Kom. degree in computer science from STMIK Budi Darma, Medan, Indonesia, in 2007, and the M.Kom. degree in computer science as well from the University of Sumatera Utara, Medan, Indonesia, in 2016. In 2014, he joined the Department of Information System, Universitas Prima Indonesia, as a Lecturer, and in 2016 became a junior researcher. He is applying for his Ph. D. degree in the middle of 2016. He has written in several international journal and conference.