Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 2, February 2019, pp. 643~648
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i2.pp643-648

An improved fitness function for automated cryptanalysis using genetic algorithm

Md. Shafiul Alam Forhad[1], Md. Sabir Hossain[2], Mohammad Obaidur Rahman[3], Md. Mostafizur Rahaman[4], Md. Mokammel Haque[5], Md. Kamrul Hossain[6]
[1,2,3,4,5]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Bangladesh.
[6]Institute of Information and Communication Technology, Chittagong University of Engineering and Technology, Bangladesh.

Article Info

Article history:

Received Aug 6, 2018
Revised Nov 24, 2018
Accepted Dec 4, 2018

Keywords:

Automated cryptanalysis
Cryptanalysis
Fitness function
Genetic algorithm

ABSTRACT

Genetic Algorithm (GA) is a popular desire for the researchers for creating an automated cryptanalysis system. GA strategy is useful for many problems. Genetic Algorithms try to solve problems by using genetic processes. Different techniques for deciding on fitness function relying on the ciphers have proposed by different researchers. The most necessary component is to set such a fitness function that can evaluate different types of ciphers on the identical scale. In this paper, we have proposed a combined fitness function that is valid for great sorts of ciphers. We use GA to select the fitness function. We have bought the higher result after imposing our proposed method.

Corresponding Author:

Md. Sabir Hossain,
Department of Computer Science & Engineering,
Chittagong University of Engineering & Technology,
Chittagong-4349, Bangladesh.
Email: sabir.cse@cuet.ac.bd

1. INTRODUCTION

The approach toward improving the plaintext from a cipher in order that they can be improved for giving higher security is called Cryptanalysis. Many cryptographic systems are inclined to an exhaustive key search attack. However, a large range of ciphers still remains separate. One of the vital issues of integrating the range of fitness function is being used for different cipher.

In this paper, we have tried to develop a fitness function that can furnish proper endpoint for a wide verity of ciphertext. We have given more importance on retrieving the key of the length longer than the required time. So there is a trade-off between time and correctness. The columnar transposition cipher is used for experimental purpose. Our proposed combined fitness feature valid for great types of ciphers. We incorporate GA to pick the fitness function and get a better result after extensive experiments.

Albassall determines the possibility of breaking a cipher by using a random type search [1]. Most classical ciphers fall under definitely one of two large categories. One category is the substitution cipher and the other one is the transposition cipher. Together with its corresponding cipher-text or the ciphertext alone known plaintext attack can make use of some plaintext [2].

In [3], Hameed, Asif, et al. presents a methodical survey of Genetic Algorithm connected to the cryptanalysis. To comprehend the extent of GAs in cryptanalysis, it gives an efficient audit. The first technique to break the transposition cipher was proposed by Matthews [4]. Bigram and trigram model used by Mathews for fitness scoring. His approach successfully broke the key measurement up to 7. A subset of the most frequent

ones is chosen by using all possible bigrams and trigrams. By using the technique proposed by Matthews, they assign a weight to each of them by listing a range of the most frequent bigrams and trigrams. As E is very common in English, a plaintext message would possibly incorporate a notably excessive number of E's. Three E's would possibly take area simultaneously. However, the mutation and mating rate had little consequences on decryption success, with greater decryptions going on at huge populace sizes and/or greater generations.

Grundlingh and Vuuren [5] use a fitness equation for selecting the best population. But it needs more decryption for a massive population size. Toemeh and Argumugam [6] try to improve Mathews algorithm by using enhancing bigram and trigram fitness weight table and introducing new fitness equation. Their method efficaciously worked for key length 17. But the algorithm desires large ciphertext (2000 letters) for precise accuracy and their crossover and mutation rates are uncontrollable. Also, they used a constant number of populations for all combinations.

Singh Pooja and R.K. Chauhan [7] present a survey of completely different asymmetric algorithms e.g. RSA, DSA, ECC, Diffie-Hellman, etc. Based on attacks, potential countermeasures, key size, strength and weakness in the form of table these can be compared. Ahmad, Jasmin, et al presents some Public Key Cryptography (PKC) algorithms with the aim of view of researcher's exertion [8]. It had been created over the foremost recent four decades. The comparative trends of PKC algorithmic rules supported a variety of analysis for each algorithm in the last four decades. They were invented and so the foremost chosen algorithms among previous researchers.

Khalid, Omran, and Hammond [9] use two equations for fitness calculation. The first fitness function gives the best solution for key size 30 and 40. The second fitness function offers a better result than the first fitness function. It offers a better result for the different variety of populations. Alkathiry and Al-Mogren [10] try to limit the number of generation for every divisor and the key with the best score kept until a better key is found. They extensively minimize the time required from [6].

Dastanpour, Amin, and Raja Mahmood Raja Azlina [11] detects distinct varieties of network attacks by investigating the performances of genetic algorithm (GA) with support vector machine (SVM). All of the algorithms are successful in accomplishing about 99% detection rate at different varieties of decreased features. GA with SVM and LCFS require completely 21 features, whilst FFSA requires 31 aspects to discover the attacks successfully. Clark, Andrew, and Dawson proposed a parallel model [12] of the genetic algorithm for attacking the simple substitution cipher. Their approach permits communication between different types of parallel nodes each fixing a separate a part of the problem. For attacking the simple substitution cipher a new technique is proposed that utilizes a parallel version of the genetic algorithm. An appropriate method is devised that permits communication between ranges of parallel nodes each solving a separate part of the problem. An analysis of the fitness function is additionally carried out.

In [13], an improved genetic algorithm with a novel fitness function is used. And it offers a technique to the cryptanalysis of transposition ciphers. The proposed algorithm is unimaginable for cryptanalysis of transposition cipher with key lengths up to twenty-five. The fitness function is evaluated by means of common bigrams and trigrams.

Dureha presents a GA called DUREHA's [14]. The underlying purpose is to automatize the technique of cryptography so as to render salvage of time, resources out there, hold people diversity, decrease the convergence value and manage mutation rates. The three types of ciphers are effectively distinguished. It also manipulates the mutation and convergence costs and preserves people diversity.

In [15], the identity of certification users has usually verified by means of the digital certificate registration, certificate generation method, and verification. They use RSA and user's property does not seem to be infringed.

2. PRELIMINARIES

2.1. Genetic Algorithm

A genetic algorithm is an optimization approach which mimics the theory of natural selection of Darwin's which is completely based on the survival of the fittest. The genetic algorithm has many variants. But the core genetic algorithm is used for this approach.

In this paper, we viewed the keys as population. These keys are evaluated and serialized according to their fitness. Then the pool is modified using crossover and mutation. In a crossover, a random point is chosen and the part of the key after that point is rearranged randomly. In mutation, two random points are chosen and they are swapped for a new key. By these operations, we create a new generation of keys to fill the pool. This procedure repeats till the defined accuracy level is reached or the maximum generation is created. Then the chromosome (key) with the best fitness value is the solution (true key).

2.2. Cryptanalysis

Cryptanalysis entails with the discovering of the key used to encrypt an undeniable text. If we can analyze about a little piece of facts from the ciphertext then it is moreover be viewed as successful cryptanalysis. So from this aspect, it is less tough than creating an impenetrable encryption technique. But full decryption is a difficult task for cryptanalyst. For this purpose, a random search is a legitimate cryptanalysis technique. Giving route in a random search is a specialty of the genetic algorithm

In a genetic algorithm, the chromosome has a phase that suggests direction. But till now we cannot add a direct phase in the chromosome used for cryptanalysis using the genetic algorithm. The primary trouble is, there is no approach which can validate the trade as superb or negative. Moreover what change will cease end result in a profitable cracking quickly. Without this, the end result can be similar to brute force or worse for the chance of generating the same key higher than one case in a random key generation procedure.

3. RESEARCH METHOD

In our work, an improved fitness function and necessary pre-calculation have proposed for automated cryptanalysis. The columnar transposition cipher is used for experimental purpose.

3.1. Input

The input to the system will be ciphertext, list as a dictionary and a prefix list of those words. The word list is used to find words from the decrypted text. The prefix list is used to reduce the complexity of searching for the word.

3.2. Create Initial Pool

In the second step, an initial pool of keys will be generated as follow:

a. Get the length of ciphertext and generate all divisors of the length.
b. Generate keys of that length for each divisor. The number of keys will be ((l-1)*1.5) where l is the divisor. For example, if l=5 then one of the keys might be 1, 4, 3, 5, 2.

In this way, a pool of initial key will be generated [10].

3.3. Generate Wordgraph

For each key, the ciphertext will be decrypted. All possible words are generated in each position of the ciphertext. A word graph is generated where the node will be the starting position of the words and child will be the words. Then traverse the graph using the depth-first search (DFS). If we get the full path from start to end of the graph then it is a candidate sentence.

3.4. Check Sentence Validity

After generating all possible candidate sentences the validity of the sentences are checked by a rule-based parser referred to as link synchronic linguistics. It is developed by John Lafferty, Daniel Sleator, and Davy Temporally et al. at Carnegie Mellon University. The link grammar produces the parsed sentences if the sentence is grammatically correct otherwise no parse is generated. Rank of the sentences is calculated by the following fitness equation:

Fitness value = (15.58 * (no. of letters in parsed sentence / entire no. of letters sent in the word graph)) + (3.35 * (no. of ‘in' in the sentence / entire no. of words in the sentence)) + (2.92 * (no. of ‘pronoun' in the sentence / entire no. of words in the sentence)) + (2.72 * (no. of ‘noun' in the sentence / entire no. of words in the sentence)) + (2.30 * (no. of ‘verb' in the sentence / total no. of words in the sentence)) + (2.27 * (no. of ‘interjection' in the sentence / total no. of words in the sentence)).

If multiple candidate sentences are generated from the word graph, the highest fitness value of those sentences will be given priority. In Figure 1, the flowchart of breaking transposition cipher using the genetic algorithm is shown.

3.5. Store the Best Key

Then search for a fitness value greater than the previous iterations. If such value can be found, then the best fitness value will be updated and the key will be set as the best key.

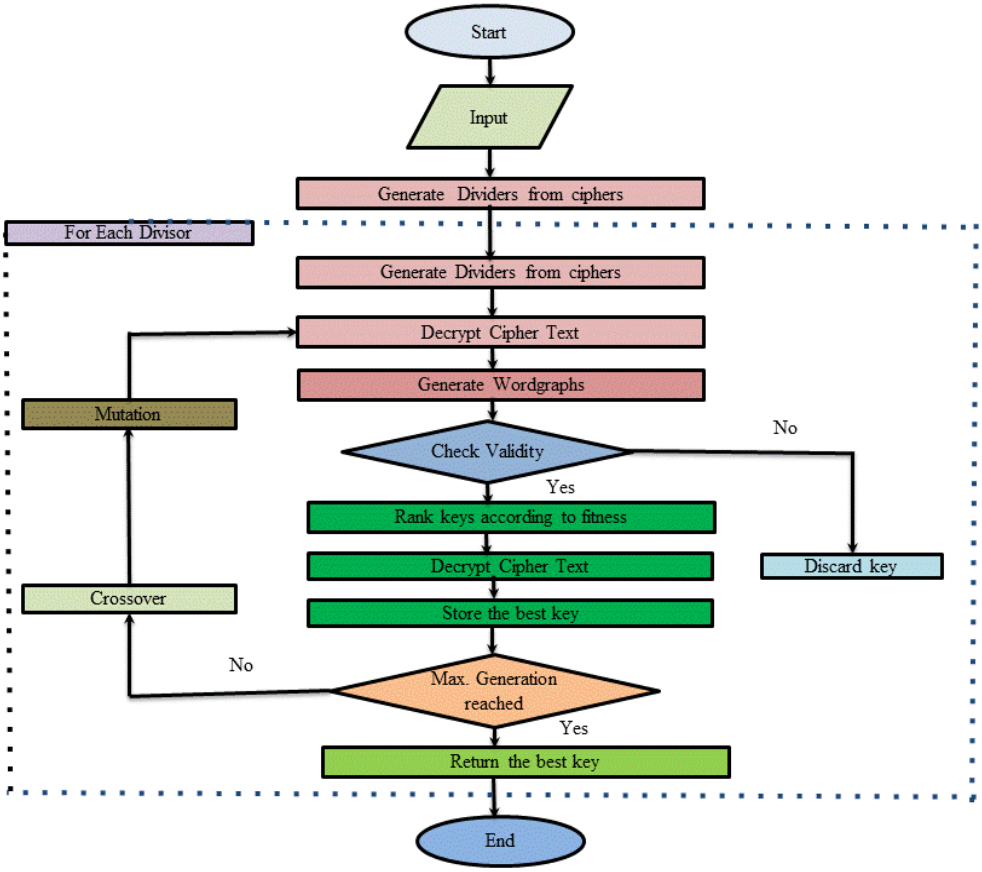

Figure 1. The flow chart of breaking transposition cipher using the GA

3.6. Crossover

Crossover is performed by selecting a point randomly and rearranging the left or right side of the elements of that point. For example: If a key is 1,6,5,7,10,3,2,4,8,9 and the crossover point is 5 then we randomly select the right side. After crossover operation the key becomes 1,6,5,7,10,3 || 4,9,2,8 [6].

3.7. Mutation

For mutation, two points are selected randomly and swapped the elements of those points. For example: If a key is 1,6,5,7,10,3,4,9,2,8 and the mutation points are 2 and 9 then after the mutation operation the key becomes 1,6,8||,7,10,3, 4,9,2,||5. The mutation rate used in our paper is 14% [6].

Steps 3.6 and 3.7 will be repeated till the most range of generation is reached. After maximum generation; the key is stored as the best possible key.

4. RESULTS AND ANALYSIS

There is no fitness function for combined algorithm to crack simple ciphers. Different cipher breaking algorithms use different fitness functions (i.e. for transposition cipher uses the bigram-trigram equation, for substitution cipher, uses monogram equation etc.). So, we have proposed this fitness function for a combined algorithm consisting of different simple cipher breaking techniques. In Figure 2, we have shown the relationship between key length and fitness value after 100 generations. We found the best result for key length 6, which was the correct key length.

For bigram-trigram, the time complexity for a text with length n will be O (n^2) where the sentence construction needs, O (nlogn) for the word graph and O (2^n) for generating all possible sentences. So, the total time complexity becomes O (nlogn) +O (2^n) = O (2^n) which is worse than O (n^2). But most of the time the algorithm does not produce any complete path from start to end of the word graph. Most of the time, there is no node at the starting position of the link. Generally, the algorithm requires only O (nlogn) which is better than the bigram-trigram algorithm.

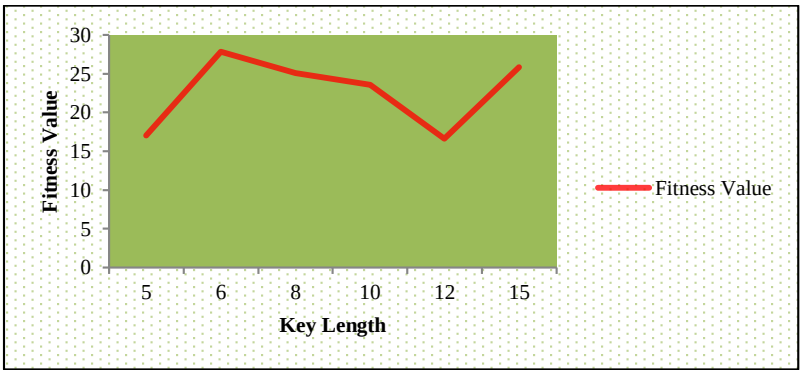

Figure 2. Key length vs. fitness value after 100 generations

We have compared our work with the work of R. Tomeh [6] and found that our proposed algorithm outperformed for cipher length 1000. In Figure 3, the comparison between R. Tomeh and our results for cipher length 1000 is shown.

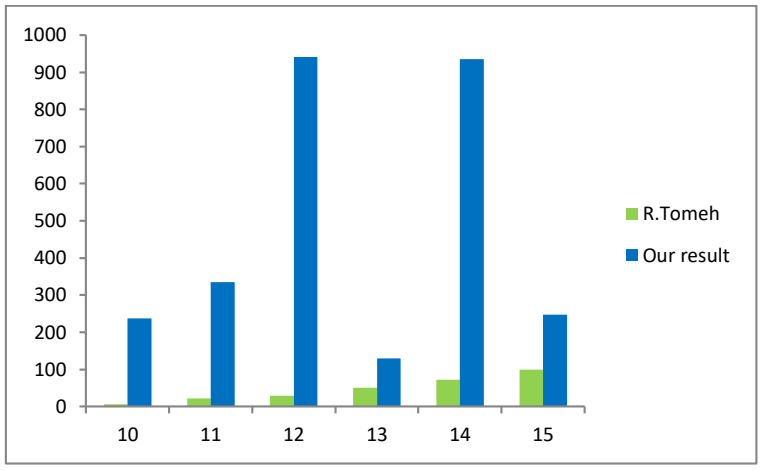

Figure 3. Comparison between R. Tomeh and our results for cipher length 1000

5. CONCLUSION

A fitness function is proposed in this paper that can be used for the combined cipher breaking algorithm. But if we can develop better algorithms for these problems using the concept of a random function generator, then the time complexity can be decreased significantly.

The random function generator is no longer specialized to generate unique key each time. A specialized random generator can be researched. The crossover and mutation operation has little or no impact on the performance of the algorithm. So, efficient crossover and mutation operations are very essential for the research in this field. An efficient technique to generate candidate sentences from a decrypted text is fundamental to the use of link grammar and fitness function. Also, the word graph does no longer reflect on consideration of the non-sense words appended deliberately with the aid of the sender. But it can be easily modified to accept the confined variety of non-sense words.

REFERENCES

[1] Albassall, A.M.B. "Genetic algorithm cryptanalysis of a feistel type block cipher." *International Conference on Electrical, Electronic and Computer Engineering, 2004. ICEEC '04*, doi:10.1109/iceec.2004.1374427.

[2] Clark, Andrew J. *Optimisation Heuristics for Cryptology*. 1998. Information Security Research Centre Faculty of Information Technology Queensland, PhD Dissertation. eprints.qut.edu.au/15777/1/Andrew_Clark_Thesis.pdf.

[3] Hameed, Asif, et al. "The Applicability of Genetic Algorithm in Cryptanalysis: A Survey." *International Journal of Computer Applications*, vol. 130, no. 9, 2015, pp. 42-46, doi:10.5120/ijca2015907118.

[4] Matthews, Robert A. "the use of genetic algorithms in cryptanalysis." *Cryptologia*, vol. 17, no. 2, 1993, pp. 187-201, doi:10.1080/0161-119391867863.
[5] Grundlingh, W.R. "Using Genetic Algorithms to break a simple cryptographic cipher." *Stellenbosch University*, 2003, dip.sun.ac.za/nvuuren/abstracts/abstr-genetic.htm.
[6] Toemeh, R., and S. Arumugam. "Breaking Transposition Cipher with Genetic Algorithm." *Electronics And Electrical Engineering*, vol. 79, no. 7, 2007, eejournal.ktu.lt/index.php/elt/article/view/10844.
[7] Singh, Pooja, and R.K. Chauhan. "A Survey on Comparisons of Cryptographic Algorithms Using Certain Parameters in WSN." *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 4, 2017, p. 2232, doi:10.11591/ijece.v7i4.pp2232-2240.
[8] Ahmad, Jasmin, et al. "Analysis Review on Public Key Cryptography Algorithms." *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, 2018, pp. 447-454, doi:10.11591/ijeecs.v12.i2.
[9] Al-Khalid, A. S. "Using Genetic Algorithms To Break a Simple Transposition Cipher." *The 6th International Conference on Information Technology (ICIT)*, 2013, pp. 20-25.
[10] Alkathiry, Omar, and Ahmad Al-Mogren. "A Powerful Genetic Algorithm to Crack a Transposition Cipher." *International Journal of Future Computer and Communication*, vol. 3, no. 6, 2014, pp. 395-399, doi:10.7763/ijfcc.2014.v3.335.
[11] Dastanpour, Amin, and Raja Mahmood Raja Azlina. "Feature Selection Based on Genetic Algorithm and Support Vector Machine for Intrusion Detection System." *Conference: The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013)*, 2013, doi:10.13140/2.1.4289.4721.
[12] Clark, Andrew, and Ed Dawson. "A parallel genetic algorithm for cryptanalysis of the polyalphabetic substitution cipher." *Cryptologia*, vol. 21, no. 2, 1997, pp. 129-138, doi:10.1080/0161-119791885850.
[13] Heydari, Morteza, et al. "Cryptanalysis of Transposition Ciphers with Long Key Lengths Using an Im-proved Genetic Algorithm." *World Applied Sciences Journal*, vol. 21, no. 8, 2013, pp. 1194-1199.
[14] Dureha, Anukriti, and Arashdeep Kaur. "A Generic Genetic Algorithm to Automate an Attack on Classical Ciphers." *International Journal of Computer Applications*, vol. 64, no. 12, 2013, pp. 20-25, doi:10.5120/10687-5588.
[15] Ming, Zhang Q. "Secure Digital Certificate Design Based on the Public Key Cryptography Algorithm." *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 12, 2013, pp. 7366-7372, doi:10.11591/telkomnika.v11i12.3824.