Cryptography Algorithm Based on Cohort Intelligence

Dipti Kapoor Sarmah and Ishaan R. Kale

Abstract Information security is very important in the current era as we share most of the information through digital media/Internet. Cryptography is a technique which converts the secret information into some other form which is referred to as ciphertext. In order to have a strong ciphertext, one should have a strong cryptography algorithm as well as the secured key. The performance of any cryptography algorithm can be measured through the secret text input file size (number of bytes), performance time to encrypt input file and how fast it can be retrieved the secret text through cryptanalysis attack. In this paper, a socio-inspired optimization algorithm referred as cohort intelligence (CI) is applied to the secret text to retrieve the optimized ciphertext. The efficiency of the algorithm is also analyzed in this paper with respect to the secret text capacity and time.

Keywords Cryptography · Cohort intelligence · Secret text capacity
Estimated time

1 Introduction

Nowadays, Internet dependency is widely increased for most of the applications such as email communication, ecommerce, social media. Due to this, all the respective tasks have become easier for human. At the same time, the security issues with respect to these applications have extensively increased. In order to protect the information during network communication, one of the important sciences is referred to as cryptography. Cryptography has been used to hide the secret text or to encrypt the secret text from readable to unintelligible form. In terms of cryptography, the secret text is named as plaintext and the unintelligible form is called as ciphertext.

D. K. Sarmah (✉) · I. R. Kale (✉)
Symbiosis Institute of Technology, Symbiosis International University, Pune 411042, India
e-mail: dipti.sarmah@sitpune.edu.in

I. R. Kale
e-mail: ishaan.kale@sitpune.edu.in

A. J. Kulkarni et al. (eds.), *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, Advances in Intelligent Systems and Computing 828, https://doi.org/10.1007/978-981-13-1610-4_44

To ensure the security of secret messages, the encryption process needs to be more complex so that the cryptanalysts are unable to decrypt the plaintext.

Cryptography techniques are formally divided into two different categories: (1) private key cryptography and (2) public-key cryptography. Various traditional cryptographic techniques were introduced such as Vernam cipher method, stream cipher cryptosystem, fast and secure stream cipher, RC4 stream cipher. However, these techniques are not very secure. There are some other known private and public-key cryptography algorithms such as data encryption standard (DES), triple DES, advanced encryption standard (AES), Diffie–Hellman key exchange algorithm, and RSA. Though the security level of this algorithm is high, however, the estimated time for converting plaintext to secret text is quite large due to its high complexity. In order to diminish these limitations, many researchers have introduced a nature and socio-based optimization techniques such as genetic algorithm (GA) [1–4], particle swarm optimization (PSO) [5, 6], ant colony optimization (ACO) [7] in various cipher methods [8, 9].

Formerly, GA was implemented by Coppersmith [1] using end-to-end security mechanism incorporated with threshold cryptography and Diffie–Hellman key exchange method to avoid the mobile network nodes. GA provides robust security to validate the nodes entering into the network. GA was also provided to break the mono-alphabetic substitution cipher [3] and to make the network more secure so that the cryptanalyst does not alter the original information/facts. Cryptanalyst can easily decode the information as they know well which ciphertext will be suited to decrypt the information. Therefore, Paul et al. [2] utilized GA incorporated with crossover and mutation which assist to make new ciphertext. Similar GA approach was implemented by Sindhuja and Devi [4] using symmetric key encryption technique for encryption and decryption. Sreelaja and Paib (2009) presented a distributed and decentralized swarm intelligence and ACO [10] approach for encryption. The data encryption standard (DES) was used in PSO to identify the plaintext from ciphertext [6]. This approach was based on two outputs such as particle best and error best, whereas the optimal solution is obtained by identifying the least error. Similar to these techniques, an emerging socio-inspired cohort intelligence (CI) algorithm incorporated with cipher method using the practice of mathematics and computer science is applied to encrypt the secret message. The CI algorithm was proposed by Kulkarni et al. [11] and successfully validated in health care and logistics [12], discrete and mixed variable from truss structure and design engineering domain [13], steganography [14], traveling salesman problem (TSP) [15], and various work domains. Though CI has been applied and validated in different engineering domain [16], however, it is untouched to information security domain yet. Thus, an effort has been put to apply and validate CI in this area which motivated us to develop a new cryptography algorithm.

There are totally four sections proposed in this paper. The organization of this paper is as follows: Proposed work is given in Sect. 2, Results and analysis are done in Sect. 3, conclusion and future scope are mentioned in Sect. 4, and references are written at the end.

2 Proposed Work

In this paper, a novel cryptography algorithm is proposed which is based on a socio-inspired optimization algorithm, i.e., cohort intelligence (CI). The main idea of this algorithm is based on a cohort. Cohort refers to the number of candidates competing with each other to improvise their own behavior. Each candidate is having their own abilities or potentials. During competition, each candidate tries to adapt the few qualities of other candidates or itself to make themselves as a better candidate which enables the cohort to improve its overall behavior. Though CI algorithm is validated to solve different test problems and real-world datasets, however, this algorithm is not applied yet in the information security domain especially in cryptography. Cryptography is one of the important sciences in information security which converts the secret text referred to as plaintext to the transformed text referred to as ciphertext. This is useful to make the secret text secure from different cryptanalysis attacks. There are many cryptographic algorithms available such as advanced encryption standard (AES), data encryption standard (DES), triple DES, Blowfish. AES is found more secure than the other mentioned methods; however, this algorithm is more complex and is dependent upon the number of rounds. Also, the computation time is very large to convert the plaintext to ciphertext. Due to this motivation, CI has been considered for this proposed work. The objective function of the proposed work is selected as the time function as:

$$f(t) = t \tag{1}$$

The flowchart of proposed work is presented in Fig. 1, and it is explained in the following steps.

Step 1: Accept the secret text. For example:

Secret text: “Cryptography”

Step 2: Each character is converted into binary digits such as if the first character of the secret text, i.e., “C” is considered, the binary digits are: 01000011

Step 3: Since there are 8 bits in every character, the following pair of bits are united to make a decimal number. For example:

(i) First and third pair bits: (00);
(ii) Second and fourth pair bits: (10);
(iii) Fifth and seventh pair bits: (01);
(iv) Sixth and eighth pair bits: (01);

Step 4: The respective pair of bits are converted into decimal numbers. Thus, totally four decimal numbers are generated. As referred to step 3, the decimal numbers are:

0 2 1 1.

Step 5: In this step, the CI algorithm is applied which considers the cohort as a group of four numbers of candidates. These numbers of candidates are represented

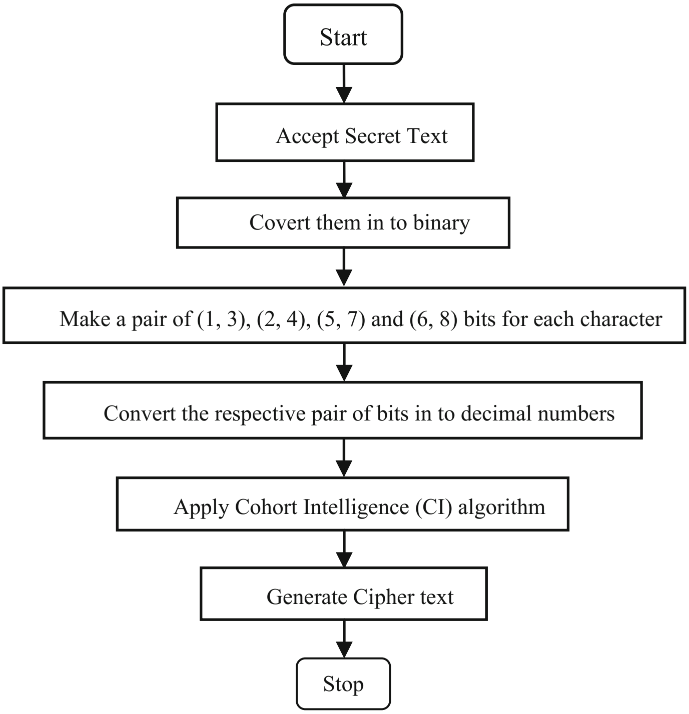
Start
Accept Secret Text
Covert them in to binary
Make a pair of (1, 3), (2, 4), (5, 7) and (6, 8) bits for each character
Convert the respective pair of bits in to decimal numbers
Apply Cohort Intelligence (CI) algorithm
Generate Cipher text
Stop

Fig. 1 Ciphertext generation using cohort intelligence algorithm

as an identity matrix and its different forms. As discussed in Step 4, the total evaluated decimal numbers are 4 for each character; thus, the dimension for each candidate is also considered as 4. The number of matrices by using 4×4 dimension by considering each pair of bits is $4!$ i.e.64. In order for experimental purpose, we have considered totally four candidates in this paper which are referred to as $B_1(i, j)$, $B_2(i, j)$, $B_3(i, j)$, $B_4(i, j)$ where i denotes the number of rows from 0 to 3 and j denotes the number of columns from 0 to 3. The quality of each candidate is considered as the row number where the matrix value is 1, and it is replaced with the column number to identify the cipher decimal value. For example: if $B(0, 2) = 1$, then 2 is replaced for the decimal number 0. Thus, each decimal number is having its corresponding value when it passes through different matrices. This helps to generate four different ciphertexts. In order to find the optimized ciphertext, the time function $f(t)$ is evaluated every time. One should always have less evaluation time to process this algorithm and to convert the plaintext into ciphertext. Since we have applied the same process for four different candidates, there will be four different evaluation time. To get the optimized one, the following points are considered:

a. Let us consider the computation time for the secret text to be converted it to ciphertext is $t1$, $t2$, $t3$ and $t4$.

b. Calculate the probability for computation time for each candidate; i.e.,

$$p_1 = \frac{t1}{t1 + t2 + t3 + t4}, p_2 = \frac{t2}{t1 + t2 + t3 + t4},$$
$$p_3 = \frac{t3}{t1 + t2 + t3 + t4}, \text{ and } p_4 = \frac{t4}{t1 + t2 + t3 + t4} \quad (2)$$

c. Calculate the cumulative probability.
d. Apply Roulette wheel approach to evaluate the candidate to be followed by other candidate.
e. To adapt the quality of the candidates, a random number from 1 to 4 is generated with respect to every candidate. This number designates the row of the following candidate to be replaced with the same row of the candidate being followed. This enables to generate a new cohort with 4 new candidates and each candidate may have different qualities.
f. The same process from step a. to step e. will be repeated till 100 times unless the saturation condition is reached. The total number of 20 runs is considered for experimental purpose.
g. The saturation condition exists where no further improvement in qualities of the candidate is identified.

Results and discussions are done in the next section. The size of the secret text and time is considered in the result section.

3 Results Analysis

Results are described in this section. The proposed work was coded in MATLAB (R2011b), and the simulations were run on Windows platform using an Intel(R) Core(TM)2Duo, 2.93 GHz processor speed, and 4 GB RAM. Time analysis with respect to the size of secret text is completed. Totally four cases are considered to capture the minimum time/optimize time taken by the candidate. Also, the total number of function evaluations and the total number of iterations of saturation condition are evaluated for a single run. Totally 20 runs are considered in the proposed work as described in the previous section. Based on the size of secret text, we have considered four cases, i.e., Case 1 for 48 bytes, Case 2 for 264 bytes, Case 3 for 480 bytes, and Case 4 for 3280 bytes. Table 1 depicted the analysis of the obtained solution with respect to evaluated parameters. We have considered five parameters for analyzing our solution. These parameters are standard deviation, minimum time, maximum time, average time, and standard deviation of function evaluations. We could see in Table 1 that the values are increased as we enlarge the secret text size for the different cases. The values against the parameters are considered with respect to 20 runs. These parameters are minimum time, iteration number on which the candidates are converged, and the number of function evaluations. As we could see from Table 1

Table 1 Result analysis of obtained solution

Parameters ↓ Cases →	Case 1	Case 2	Case 3	Case 4
Standard deviation	0.000227	0.00017	0.006735	0.000603
Min time	0.007598	0.035767	1.040855	0.092433
Max time	0.008545	0.036417	1.065062	0.094605
Average	0.008258	0.03603	1.049098	0.093233
Standard deviation function evaluations	94.72753	107.145	131.3829	124.5473

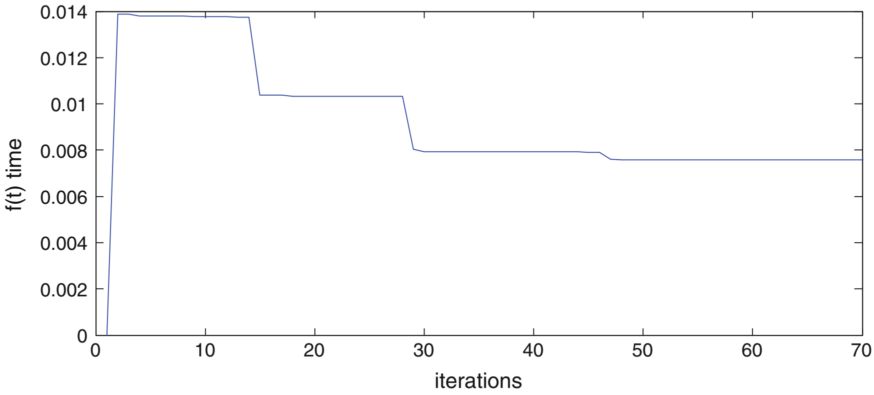

Fig. 2 Convergence plot for optimum solution 48 byte

that the standard deviation of time taken to convert 48 bytes of string (Case 1), i.e., plaintext to cipher text, using CI is quite less and the standard deviation of function evaluation is also 94.72753 which indicates that the proposed method is pretty fast. If we increase the size of secret text from 48 to 264 bytes, Case 2 is considered. We could observe that the standard deviation of time taken from Case 1 to Case 2 is decreased from 0.000227 to 0.00017. The calculated standard deviation for Case 3 is 0.000603, and the standard deviation of function evaluations is increased from Case 2 to 124.5473. Again, the secret text size is increased to 3280 bytes to see the change in the previous cases, and it is observed that the standard deviation of time taken and standard deviation of function evaluations are increased than the previous cases; however, this increment is not very substantial. Convergence plots are also considered and shown for each case as shown in the plot section.

Plots are also captured against each case as shown from Figs. 2, 3, 4, and 5. As shown in each plot, two dimensions are considered. X-axis describes the total number of iterations, and Y-axis describes the time taken. In every plot, the converged value of iteration and its corresponding captured time could be seen. In Figs. 2, 3, 4, and

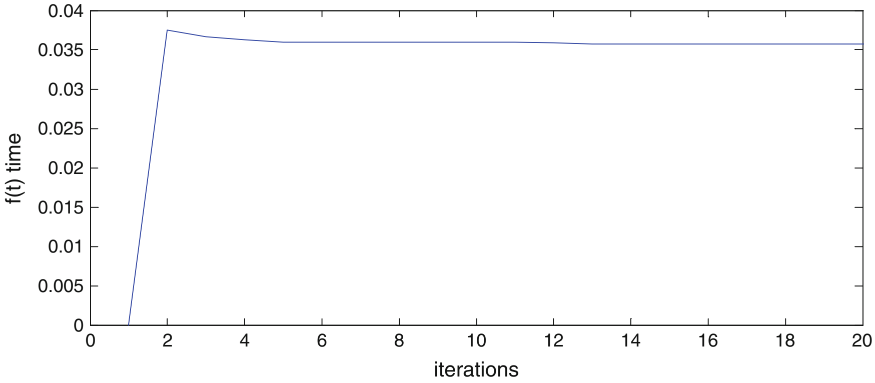

Fig. 3 Convergence plot for optimum solution 264 byte

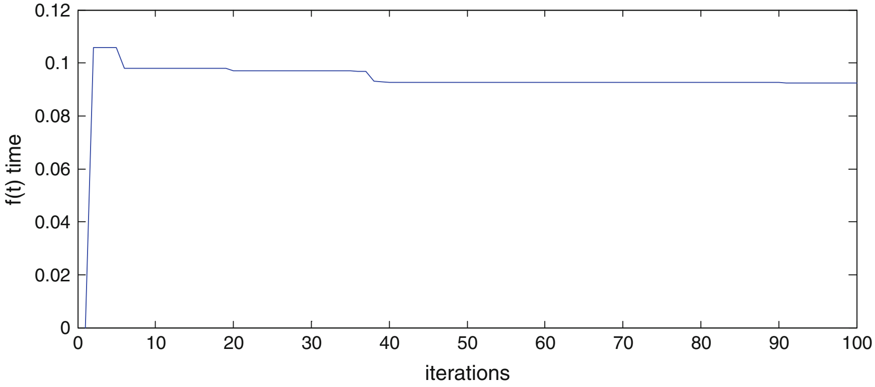

Fig. 4 Convergence plot for optimum solution 480 byte

5, one could observe that as we increase the size of the secret text, the encryption time also gets increased.

4 Conclusion and Future Scope

Due to its results as shown in the previous section, the computation time of the proposed work is very less. Even if we increase the size of the secret text to 3280 bytes, the evaluated parameters are found better than the other similar type of algorithms. Also, this method is found very secure because cryptanalyst needs to identify the optimized matrix as well as the optimized secret bits which makes the entire algorithm more secure. However, the combination of this algorithm and the other existing

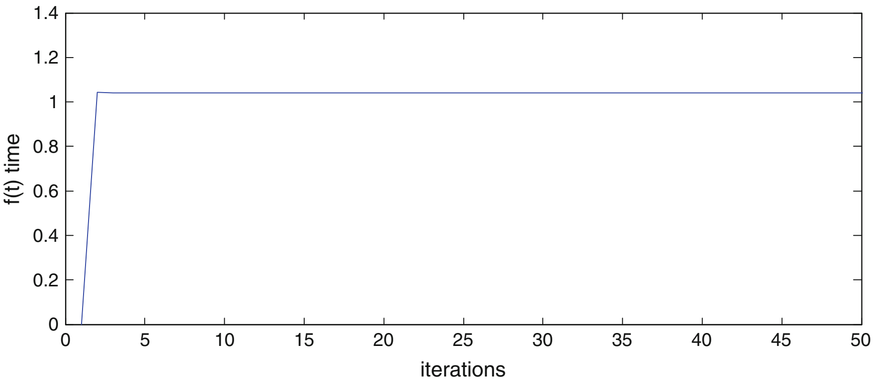

Fig. 5 Convergence plot for optimum solution 3280 byte

algorithms can also be tried to make the algorithm more secure and complex. This opens a new door for researchers to work in this direction.

References

1. Coppersmith D (1994) The data encryption standard (DES) and its strength against attacks. IBM J Res Dev 38(3):243–250
2. Paul S, Dutt I, Choudhri SN (2013) Design and implementation of network security using genetic algorithm. Int J Res Eng Technol 2(2):172–177
3. Omran SS, Al-Khalid AS, Al-Saady DM (2010) Using genetic algorithm to break a mono-alphabetic substitution cipher. In: 2010 IEEE conference open systems (ICOS), pp 63–67
4. Sindhuja K, Devi PS (2014) A symmetric key encryption technique using genetic algorithm. Int J Comput Sci Inf Technol 5(1):414–416
5. Abdul Halim MF, Bara'a, AA, Hameed SM (2008) May. a binary particle swarm optimization for attacking knapsacks cipher algorithm. In: International Conference Computer and communication engineering, 2008. ICCCE 2008, pp 77–81
6. Pandey S, Mishra M (2012) Particle swarm optimization in cryptanalysis of DES. Int J Adv Res Comput Eng Technol (IJARCET) 1(4):379
7. Khan S, Shahzad W, Khan FA (2010) Cryptanalysis of four-rounded DES using ant colony optimization. In: 2010 International conference information science and applications (ICISA), pp 1–7
8. Biham E, Seberry J (2005) Py (Roo): a fast and secure stream cipher using rolling arrays. IACR Cryptology ePrint Archive, pp 155
9. Kim H, Han J, Cho S (2007) An efficient implementation of RC4 cipher for encrypting multimedia files on mobile devices. In: Proceedings of the 2007 ACM symposium on applied computing pp 1171–1175
10. Sreelaja NK, Pai GV (2012) Stream cipher for binary image encryption using ant colony optimization based key generation. Appl Soft Comput 12(9):2879–2895
11. Kulkarni AJ, Durugkar IP, Kumar M (2013) Cohort intelligence: a self-supervised learning behavior. In: 2013 IEEE international conference, systems, man, and cybernetics (SMC), pp 1396–1400

12. Kulkarni AJ, Baki MF, Chaouch BA (2016) Application of the cohort-intelligence optimization method to three selected combinatorial optimization problems. Eur J Oper Res 250(2):427–447
13. Kale IR, Kulkarni AJ (2017) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parall Emergent Distributed Syst 1–36
14. Sarmah DK, Kulkarni AJ (2017) Image steganography capacity improvement using cohort intelligence and modified multi-random start local search methods. Arabian J Sci Eng 1–24
15. Kulkarni AJ, Krishnasamy G, Abraham A (2017) Cohort intelligence: a socio-inspired optimization method. Springer, Heidelberg, Germany
16. Dhavle SV, Kulkarni AJ, Shastri A, Kale IR (2016) Design and economic optimization of shell-and-tube heat exchanger using cohort intelligence algorithm. Neural Comput Appl 1–15
17. Sreelaja NK, Vijayalakshmi Pai GA (2011) Swarm intelligence based key generation for stream cipher. Security Commun Networks 4(2):181–194