Improving performance of the symmetrical evolutionary ciphering system SEC

Mohammed Bougrine*, Salima Trichni and Fouzia Omary

Faculty of Sciences of Rabat,
4 Avenue Ibn Batouta, B.P. 1014,
Rabat, 10106, Morocco
Email: mr.mohammed.bougrine@gmail.com
Email: tr.salima@gmail.com
Email: omary@fsr.ac.ma
*Corresponding author

Abstract: Nowadays computer security becomes, increasingly, an indispensable field. It represents a major challenge for all entities: economic, political, social … However, few are the cryptographic systems that ensure security and still resists the enormous growth in technology. Now, cryptanalysis tools are much more sophisticated and more powerful than before. Hence, the need to design new systems that can be competitive to the old ones. New cryptographic systems with new properties and the ability to support this technology watch. In this work we have focused on the SEC encryption system uses a new approach of ciphering based on evolutionary algorithms, hence it's called: symmetrical evolutionary ciphering. Since the strength of this type of algorithm will be very beneficial in cryptography, we will present in this paper, a new evaluation function that we have developed to improve the performance of this system and improve its resistance against all possible types of attacks.

Keywords: security; cryptography; cipher; symmetrical encryption; evolutionary algorithms; fitness function; appearance frequency; integrity; secret key; SEC.

Reference to this paper should be made as follows: Bougrine, M., Trichni, S. and Omary, F. (2021) 'Improving performance of the symmetrical evolutionary ciphering system SEC', *Int. J. High Performance Systems Architecture*, Vol. 10, No. 1, pp.12–19.

Biographical notes: Mohammed Bougrine got his Master Offshoring in Applied Computer Sciences 2009 and currently PhD Student in Department of Computer Sciences, Faculty of Sciences, Mohammed V University in Rabat, Morocco. He is member of "Intelligent Processing & Security of Systems" team, where she does her research in the field of cryptography and security. He held the position of Technical Architect in an international Computer Company.

Salima Trichni got her Master Offshoring in applied Computer Sciences 2008 and currently PhD Student in Department of Computer Sciences, Faculty of Sciences, Mohammed V University in Rabat, Morocco. She is member of "Intelligent Processing & Security of Systems" team, where she does her research in the field of cryptography and security. She held the position of BI Technical expert in an international Computer Company.

Fouzia Omary having obtained her Bachelor in applied mathematics and her DEA (Diploma of Advanced Studies) in Computer Science, she obtained her PhD in Computer Science in the field of Compilation to get the position of assistant professor in computer science in the Department of Mathematics of the Faculty of Sciences of Rabat. She became Professor at university by obtaining the hability to direct the research in Cryptography field in 2006. Also, she was Director of Laboratory LRI of Research in Computer Science. And from 2016 she is Head of Research Structure "Intelligent Processing & Security of Systems" (IPSS).

This paper is a revised and expanded version of a paper entitled 'Improving performance of the symmetrical evolutionary ciphering system SEC' presented at *International Conference on Modern Intelligent Systems Concepts (MISC'18)*, Rabat, Morocco, 12–13 December, 2018.

1 Introduction

The Evolutionary Algorithm (EA) is a stochastic search algorithm using mechanisms inspired by natural evolution. It can efficiently be used to solve large types of optimisation problems (Trichni et al., 2013). Moreover, it develops good approximate solutions to different types of problems in terms of finding good results by using a performing fitness

function. Given a group of genes, the fitness function serves as an effective tool to distinguish between good and worst individuals in the population and can also determine which solutions are better into the all candidate solutions to the problem. The majority of these methods are used to solve optimisation problems, namely combinatorial ones which are generally NP-complete or NP-hard. The EA uses also random processes like crossover and mutation operators, which are benefits for the field of cryptography.

In this paper, we present a new use of EA in cryptography to more secure information exchange. Indeed, the industrial revolution that we are currently experiencing is at the base of the enormous and rapid evolution of new technologies. However, data security and privacy is a critical constraint that strengthens the competitiveness and development of all sectors. The adoption of good security policy at the level of digital tools becomes a necessity and must be based mainly on the use of secure cryptographic tools, mathematically proven and able to ensure confidentiality, integrity, authentication, and availability of information. Our cryptosystem SEC, which is based on the EA, ensures data confidentiality. It is a symmetrical encryption system, which transforms the problem of encryption to a combinatorial optimisation one. Through this work, we will see how to improve its fitness function to assure two conditions, the first one is to maximise the difference of the character's appearance frequency, and the second one is to maximise the disturbance between the character positions in the cipher text. To present this work, the paper will be organised as follows: First, we will describe of our cryptosystem SEC. Then, we will describe the detail of our improving fitness function. Experimental results and discussion will be given at the end.

2 Description of SEC

SEC is the newest version of symmetrical encryption systems, which has contributed to transforming the encryption issue into a combinatorial optimisation problem, by using basically the EA (Omary, 2006). The plaintext to encrypt is represented by a group of disjoint lists. Each list contains the different positions of a character in a clear message (Omary et al., 2005, 2006). The goal is to maximise the exchange of positions and frequent occurrence of the different characters in the message. The ciphering with SEC begins by generating random solutions by using cryptography processes (Florin and Natkin, 2002).

- M is the plaintext to encrypt. It is formed by a sequence of n characters which can be only numbers, or can be a combination of numbers and alphabets.
- Let $c_1, c_2, \ldots, c_m$: different characters of M.
- Li $(1 \le i \le m)$; list of different positions of the character ci before encryption
- card(Li) : number of occurrences of ci in the message M
- $Li \cap Lj = \varnothing$ if $i \ne j$. $\forall i, j \in \{1, 2, \ldots, m\}$.
- $L_1, L_2 \ldots L_m$ is a partition of the set $\{1, 2, \ldots, m\}$
- M can be represented by the vector:

(c_1, L_1)	(c_2, L_2)	…	…	…	(c_m, L_m)

The aim of ciphering by the SEC is to change the frequency of the character's appearance in the clear message M and to make a maximum disorder between the character's positions (Goldberg, 1989; Omary et al., 2005). In fact, we thought that the distribution lists over the different characters of M must be iteratively changed. In particular, we must choose the permutation σ of 1, 2, …, m for which the difference between the cardinal of the new list $L\sigma(i)$ of the Ith character and Cardinal of the original list Li is maximal. By this, we will be confronted with a combinatorial optimisation problem that will be solved using evolutionary algorithms.

2.1 SEC algorithm

Coding:

- we use a chromosome as a vector of size m
- L_{pi} $(1 \le i \le m)$ is the lists of genes, which contains a news positions of the character c_i.

Initialisation:

- Creation of q individuals: $X_1, X_2, \ldots, X_q$ of the initial population P_0.
- Original-Ch is the chromosome composed by $L_1, L_2, \ldots, L_m$ (in this order) that represent the message before encryption. A simple modification of Original-Ch is able to change the appearance frequency of the characters.

Evaluation of individuals: Xj an individual of Pi has L_{j1}, $L_{j2}, \ldots, L_{jm}$ as genes. The fitness function F is determined for the group of individuals Xj by:

$$F(Xj) = \sum_{i=1}^{n} |\,\mathrm{card}(Lji) - \mathrm{card}(Li)\,|$$

Selection of the best individuals: The conventional method of the roulette wheel takes only the strongest individuals. The fitness function is introduced to discriminate individuals representing a minority of genes that have changed from the original chromosome. This problem can be reduced to a problem of permutations with constraints; the genetic operators are best applied to solve them (Goldberg, 1989).

Crossover MPX (maximal preservative X): This Crossover MPX is applied to chosen individuals with specific rates. The best rate is between 60% and 100%.

Transposition mutation: The choice of the mutation is a random permutation of two genes on a chromosome. This is applied to individuals with a suitable crossover speed, preferably about 0.1% to 5%. Place the new offspring population to a new P_{i+1}.

Repeat steps 2, 3, and 4 until a stop condition (Grefenstette, 1986).

Stopping condition: For all X, the function F is defined between $0 \leq F(X) \leq 2*m$. Since it is bounded, the function F has a maximum. According to some research results, the convergence of the fitness function is provided but may be close to the maximum. Final-Ch represents our final solution given by the evolutionary algorithm. The symmetric key called the genetic key is constructed from the Original-Ch to Final-Ch (Khan, 1988).

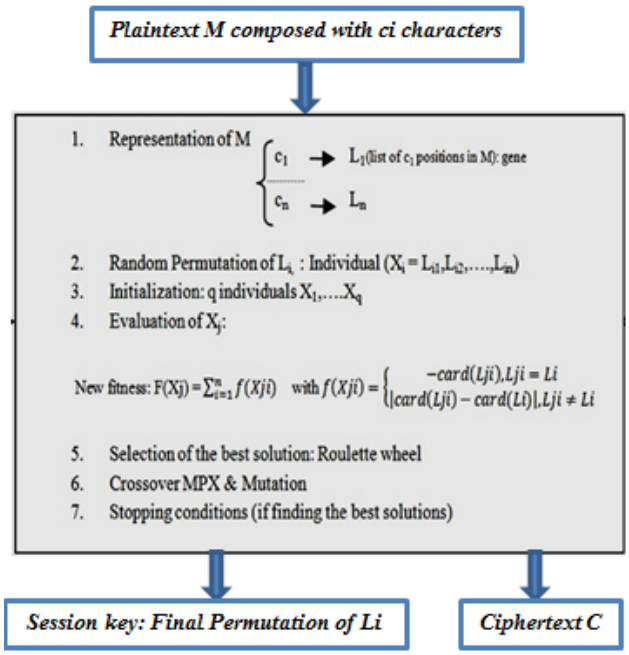
3 Description of the new fitness function

In the evolutionary algorithm, the determination of the fitness function is a very critical phase to find the best solution. In our situation, the properties that we are trying to evolve in this solution concern essentially the maximisation of both following properties:

- the appearance frequency of characters
- the difference between the initial solution and the final solution.

Going back to the initial version of our fitness function, we find that maximising the difference between the appearance frequencies of characters is a good condition for evaluating our solutions. However, we cannot eliminate the possibility to have a better solution with characters occupy the same positions as in the original text, which is unacceptable in a ciphertext. So, in this work we have ensured that the fitness function developed in SEC must satisfy the following constraint: "No plaintext characters must keep its list of occurrences in the ciphertext" (Figure 1).

Figure 1 Algorithm of our new version of SEC (see online version for colours)



Our new function aims to maximise the difference between the appearance frequencies characters without neglecting the fact when the initial lists are assigned to their original characters.

3.1 Mathematical formulation

M is the plaintext to encrypt.

- X_0 is the original chromosome (Original-CH) representing M with $X_0 = \{L_1, L_2, L_3, \ldots, L_n\}$.
- $X_j = \{L_{j1}, L_{j2}, \ldots, L_{jn}\}$ is the new chromosome to estimate.

The new formula of the fitness function puts us in front of two different situations:

- the gene *Lji* coincides with *Li*
- the genes *Lji*, *Li* are different.

From this new formula, we try to push aside the first situation and minimise the fitness value of the solution that contains this case, so it cannot be selected. Consequently, the new objective function that we propose is expressed as follows:

$$F(Xj) = \sum_{i=1}^{n} f(Xji)$$

With

$$f(Xij) = \begin{cases} -\text{card}(Lji), & \text{if } Lij = Li \\ |\,\text{card}(Lji) - \text{card}(Li)\,|, & \text{if } Lij \neq Li \end{cases}$$

Therefore, the permutation representing the individual *Xji* is evaluated according to the position of each list *Lji* in this permutation. It takes two values:

- A positive value that represents the difference between the new and the old list of character cj, aims to maximise.

A negative value of Card(*Lji*) when *Lji* coincides with the original list of *cj*. It has a negative effect on the sum of this function for all the lists that construct this permutation. As a result, it will undo its chance to be selected as a solution for our system.

3.2 Application of the new fitness function to the SEC

Theoretically, the application of the new formula for the evaluation function must have a positive impact on the following three components:

- the quality of the solution obtained
- the convergence of application
- the runtime system.

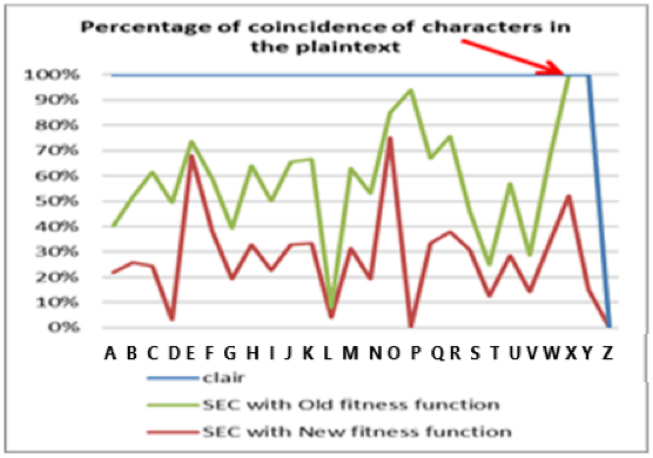
The various realised experiments, join this context and consist of comparing the results obtained with the old evaluation function of system SEC and the new function

integrated into the same system. Below an overview of some results from these experiments:

The quality of the optimal solution: To ensure the quality of the solution obtained by our system, we decided to work on with the same initial randomly generated population for the first time. The obtained results are not very different compared to the new solution because, in practice, the probability of having one of the characters in the clear message appearing in the same initial positions is weak even with the old fitness function. Figure 2 represents a successful experiment of this scenario. It shows the percentage of coincidence of each character in the plaintext and the ciphertext once with the new fitness function and once with the old formula of this function.

As you can notice in the Figure 2, the percentage of coincidence of characters position in the plaintext and in the ciphertext using the new fitness function of the SEC system is, the most, less than 40%. However, in the first ciphertext, this value is bigger, almost between 40% and 60% or sometimes it could exceed widely the 60%. Furthermore, with the new formula, we cannot have the same list of characters positions as of the initial text. Something we were able to have only once and for this text, by using the old fitness function.

Figure 2 Percentage of characters coincidence compared with plaintext (see online version for colours)



Study of convergence:

The convergence of our encryption system is always assured either with the old or the new fitness function. However, our goal through this experiment is to compare:

- the number of iterations at which this convergence was completed
- the maximal value of the optimal solution held by the system
- and also, the difference between population from one to another iteration.

Figures 3 and 4 show the finding results of these three points: The number of iterations and the convergence value of the two fitness functions are shown in Figure 3, while the results of the difference between populations are shown in Figure 4.

Figure 3 Convergence of the fitness function (see online version for colours)

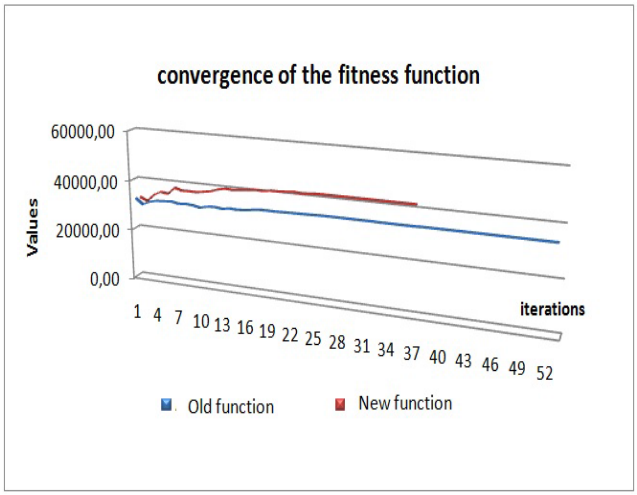
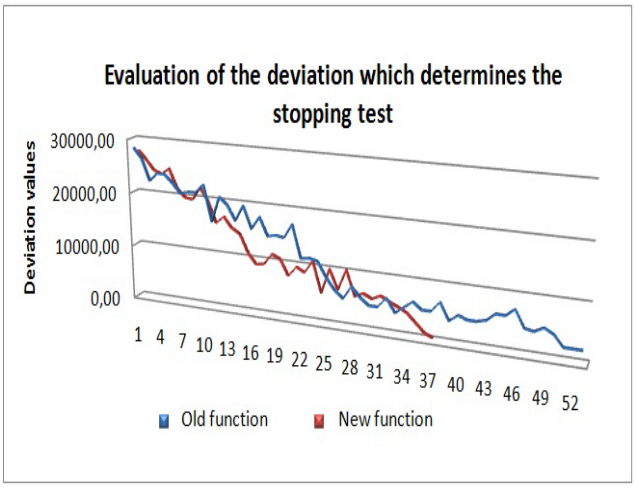


Figure 4 Evaluation of the deviation which determines the stopping test (see online version for colours)



By reading the result given in Figure 3, we can make a comparison between the values of the fitness function for each iteration, and to deduce the convergence of the SEC system using the old and the new objective function. For the old version of the system (blue curve), the convergence is reached after the 52nd iteration. While the convergence in the new system (red curve) is reached earlier than in the latter, and this after the 37th iteration. The difference is clear and the system execution performance becomes faster and powerful than before.

Comparing the show results in Figure 4, we remark that the two curves of gap value are decreasing from each iteration. However, what makes the difference between these two curves; is the value of the gap where the execution of the system ends. We remark also that this value equals to 0 at the end of the execution of SEC using the new

fitness function. It means that the convergence of the population is total and tends to the same solution. On the other side, the old function ends with a value equal to 967 (for this example).

In this experience, we search to measure the performance of our system and the impact of applying the new fitness on terms of the execution time.

As disclosed in Table 1, we measure the execution time of our system using texts of different sizes. Since we use a random process in the steps of our algorithm, we choose to repeat each experiment several times to get a more detailed view on the average time consumed for each text.

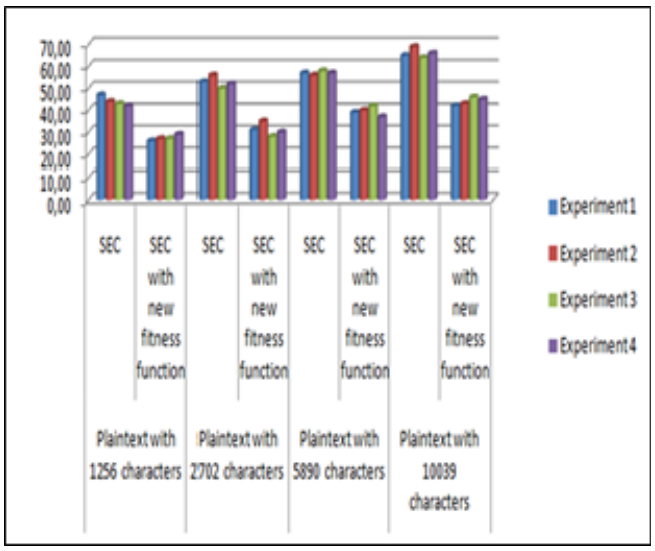
Table 1 Impact on execution time by applying the new fitness to SEC

Plaintext size	*Ciphering system*	*Experiments/Executing time* (ms)			
		1	*2*	*3*	*4*
1256 characters	SEC	47.00	44.00	43.00	42.00
	SEC with new fitness function	26.00	27.00	27.00	29.00
2702 characters	SEC	53.00	56.00	50.00	52.00
	SEC with new fitness function	31.00	35.00	28.00	30.00
5890 characters	SEC	57.00	56.00	58.00	57.00
	SEC with new fitness function	39.00	40.00	42.00	37.00
10039 characters	SEC	65.00	69.00	64.00	66.00
	SEC with new fitness function	42.00	43.00	46.00	45.00

Figure 5 represented the results found in Table 1.

This graph shows the improvement of the execution time of our system. The new function has optimised almost half time the period to have a better solution.

Figure 5 Graphical representation of the found results in Table 1 (see online version for colours)



Impact on the extended version of SEC

In the previous experiments, we have shown the advantages of integrating the new fitness function in the SEC system. Given that the latter has been extended (Bougrine et al., 2012) combined with other over-ciphering processes such as "Subdivision or fragmentation Process in Omary (2006), or used for other security goals as SEC-CMAC for message authentication in Castroa et al. (2005) or other uses with a complex problem (Even and Mansour, 1997). Then, improving this basic system will be very beneficial for all these systems. For example, in this section, we present the impact of this function of the combination of ciphering SEC and the Subdivision Process SEC-SP.

Table 2 shows the values of the execution time for this system, both with and without integrating our new evaluation function.

Table 2 Impact execution time by applying the new fitness to SEC–SP

Plaintext size	*Ciphering system*	*Experiments/Executing time* (ms)				
		1	*2*	*3*	*4*	*Average time*
1256 characters	SEC	51.00	48.00	48.00	47.00	48.50
	SEC with new fitness function	31.00	37.00	31.00	33.00	33.00
2702 characters	SEC	60.00	59.00	61.00	64.00	61.00
	SEC with new fitness function	37.00	39.00	40.00	38.00	38.50
5890 characters	SEC	65.00	63.00	66.00	66.00	65.00
	SEC with new fitness function	44.00	47.00	42.00	37.00	42.50
10039 characters	SEC	73.00	69.00	71.00	70.00	70.75
	SEC with new fitness function	53.00	53.00	52.00	55.00	53.25

Figure 6 Graph of results found in Table 2 (Impact execution time by applying the new fitness to SEC–SP) (see online version for colours)

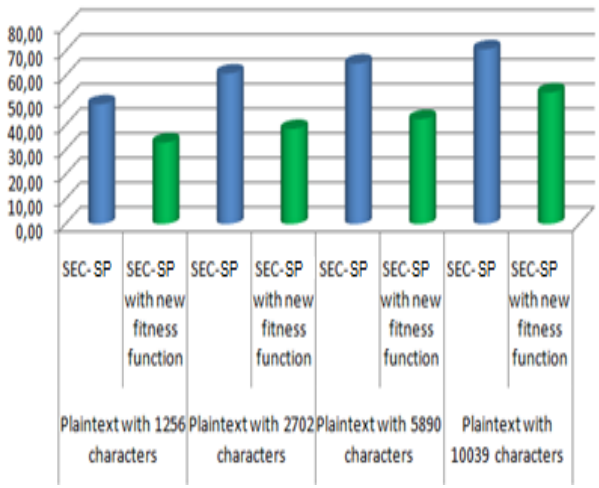


Figure 6 represented the results found by applying the new fitness function to SEC-SP system which represented in Table 2.

Through this graphical representation, we can generalise the observation occurring following the previous experiment and conclude that the use of our new function optimises the run time systems using SEC up to 50% throughout the encryption process.

4 Algorithm and security

As we have presented above, the SEC encryption algorithm is an evolutionary algorithm that has several characteristics helping in the security of the system, namely:

- it is a non-deterministic algorithm which is based on choices and probabilities in order to decide the final solution
- it is based on random processes
- it is adaptable and flexible: we can change the parameterisation of the algorithm according to types of texts
- it generates in each execution different solutions even for the same clear text, especially if the design of the initial population changes from one encryption to another.

5 Brute force attack

The brute force attack consists of trying and checking all possible Keys of the symmetrical system. In our case, the size of the key is of the order of 8 * n where n is the number of different characters in the plain text. So if we assume that our plain text contains only 26 different characters than the key size is about 208 bit, which will allow us to cover an interesting level of security compared to most of the known symmetrical ciphering algorithms. Indeed, to attack this key we will need to browse 2208 possibilities to find the correct key thing that is not possible now even with the most powerful computers.

On the other hand, the system key is a session key because it is generated at each encryption and changes from one transaction to another. From this fact, we can say that this type of attack is not useful because it will involve a great effort and investment and will only be used in the current transaction.

6 Performance comparison

To study the performance of our encryption system, we will compare it with the most widely used symmetric encryption systems today, namely TripleDES and AES.

In what follows, we present two successive experiences allowing to compare the performance of TripleDES, AES, the old SEC and the new one that we presented in this work.

The first experiment (Figure 7) shows the encryption time for each system while the second (Figure 8) relates to the decryption time.

Regarding the encryption time, we note that the new version of SEC is faster than Triple DES and the old version of this same system, while it remains a little slower than the encryption of AES.

For the decryption process, our system is much faster than TripleDES and AES.

Also, if we sum the time spent to carry out all the operations of encryption and decryption of a plaintext, we see that our system was classified first as shown in graph of the Figure 9.

Figure 7 Encryption time comparison between the best ciphering systems (see online version for colours)

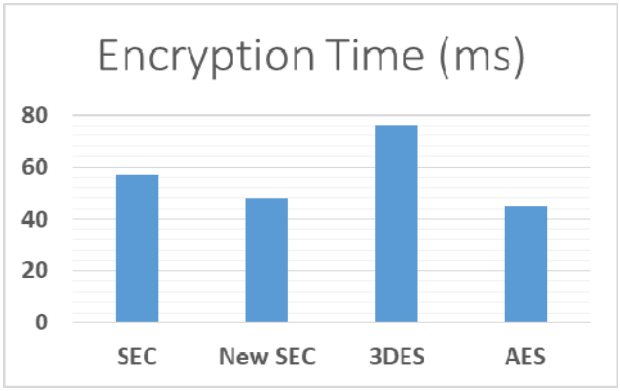


Figure 8 Decryption time comparison between the best ciphering systems (see online version for colours)

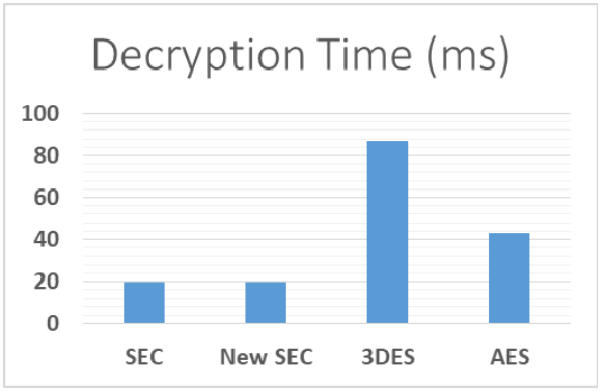
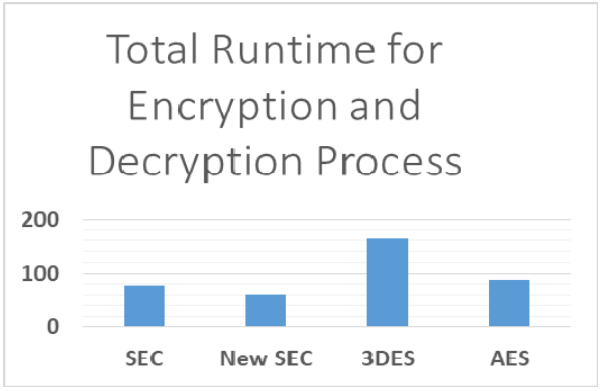


Figure 9 Comparison of total execution time between the best ciphering systems (see online version for colours)



7 Hamming distance and avalanche test

In this section, we focalisate on the avalanche effect to matematically prove the randomness of this ciphering and the independence of the output from the input (Marsaglia and Wai Wan, 2002). The avalanche effect is based on the calcul of the Hamming Distance between the output vectors generated by our ciphering function (C) for inputs generated by randomly changing one of its bits (Ziani and Fouzia, 2019; Even and Mansour, 1997; Steinberger, 2012). The result of this operation should be $n/2$.

Mathematicaly:

Let M_1 be the first input

And M_2 be the second input generated by changing 1 bit from M_1

Then:

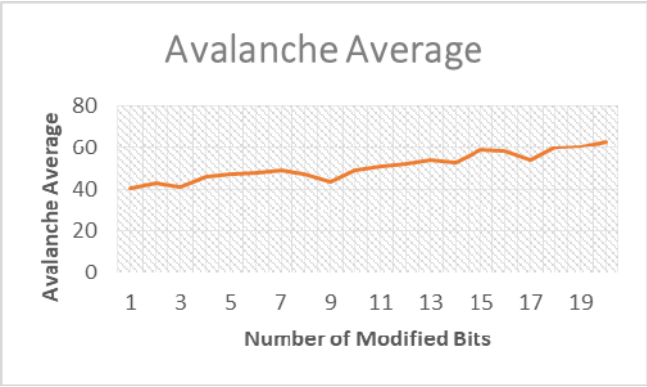
$$\forall\, M_1, M_2 \mid H(M_1, M_2) = 1,$$
$$\text{average}(H(C(M_1), C(M_2))) = n/2$$

n this experiment we suppose that we have 256 different character in the input ($n = 256$). Then we calculate the avarege of avalanche A for each modification in the input text as fellow :

$$A = (1/256) * H(C(M_1), C(M_2)) * 100$$

Figure 10 shows the obtained results in this experiment.

Figure 10 The avalanche effect on the different encrypted by changing each time a number of bits randomly (see online version for colours)



As we can notice in this experiment, the avalanche value is significantly good and take values close to 50%. also we note that this value increases and exceeds the porcentage 50% by increasing the number of bits modified in the input text.

8 Discussion and conclusion

After the conception and the realisation of the encryption system SEC based on the evolutionary algorithms, we are presently trying to see how can we increase the strength of this system and make it competitive against the different cryptographic systems that are currently used.

We then realised several works to achieve this goal:

Through this work, we have successfully designed a new evaluation function that cancels the probability that a character occupies the same setting in the original text. In fact, given that the SEC system tries to prevent cryptanalysis by appearance frequency and to find the most efficient distribution between the characters and their positions in the plaintext. Perhaps, sometimes we end up with an optimal solution having a maximum value of the evaluation function but which retains an infrequent character in its initial position since its value has no impact on the sum of this function.

This situation should not occur even if its probability is almost zero, why we thought to include this case in the evaluation function, and make a multi-criteria function.

To measure the efficiency of this function we conducted several experiments on texts that we could detect this problem.

The results were very significant. First, because we were able to end the possibility to have an optimal solution that preserves a non-prevalent character in its original place. Secondly, because this new function has allowed us to end up with a value of convergence equal to 0.

Furthermore, other interesting findings results in these experiments show the improvement of the system performance due to the application of this new fitness function. The cryptosystem SEC now is faster than before in terms of execution time and consumes less memory space because of decreasing in the number of iterations. Also, it can be competitive with the most powerful encryption systems as shown by comparing it with Triple DES and AES.

As part of our perspective, we think also to reduce the size of the key and convert the system to an asymmetric system based on the partition problem.

References

Bougrine, M., Omary, F., Trichni, S. and Boulahiat, B. (2012) 'New evolutionary tools for a new ciphering system SEC version', *2012 IEEE International Carnahan Conference on Security Technology (ICCST)*, Boston, MA, pp.140–146, doi: 10.1109/CCST.2012.6393549.

Castroa, J.C.H., Sierrab, J.M., Sezneca, A., Izquierdoa, A. and Ribagordaa, A. (2005) 'The strict avalanche criterion randomness test', *Mathematics and Computers in Simulation*, Vol. 68, pp.1–7.

Even, S. and Mansour, Y. (1997) 'A construction of a cipher from a single pseudorandom permutation', *J. Cryptology*, Vol. 10, pp.151–161, https://doi.org/10.1007/s001459900025

Florin, G. and Natkin, S. (2002) 'Techniques of cryptography', *Cnam 2002*.

Goldberg, D.E. (1989) *Genetic Algorithms In Search Optimisation & Machine Learning*, Addison-Wesley Publishing Company, Inc., USA.

Grefenstette, J.J. (1986) 'Optimization of control parameters for genetic algorithms', *IEEE Trans. on SMC*, Vol. 16, No. 1, January–February, pp.122–128.

Khan, P.C. (1988) *Heuristic and Evolutionary Algorithms*, Doctoral Thesis, University of Lille, October.

Marsaglia, G. and Wai Wan, T. (2002) 'Some difficult-to-pass tests of randomness', *J. Sta. Software*, Vol. 7, No. 3.

Omary, F. (2006) Application of Evolutionary Algorithms to Cryptography (Applications Des Algorithmes Evolutionnistes À La Cryptographie), Doctoral Thesis, University Mohammed V Agdal, Faculty Of Science, Rabat, Marocco, July.

Omary, F., Tragha, A., Lbekkouri, A., Bellaachia, A. and Mouloudi, A. (2005) *An Evolutionist Algorithm To Cryptography*, Lecture Series And Computational Sciences Volume 4, Brill Academic Publishers, pp.1749–1752.

Omary, F., Tragha, A., Lbekkouri, A., Bellaachia, A. and Mouloudi, A. (2006) *A New Ciphering Method Associated With Evolutionary Algorithm – Lecture Notes In Computer Science*, Computer Science-Volume 3984/2006, Springer Berlin/Heidelberg, ISSN: 0302-9743.

Steinberger, J. (2012) *Improved Security Bounds for Key-Alternating Ciphers via Hellinger Distance*, https://eprint.iacr.org/2012/481.pdf

Trichni, S., Omary, F., Boulahiat, B. and Bougrine, M. (2013) 'A new approach of mutation operator applied to the ciphering system SEC', *Iccit 2011*, Vol. 63, No. 9, September.

Ziani, F.E. and Fouzia, O. (2019) 'Partition ciphering system: a difficult problem based encryption scheme', *International Journal of Advanced Computer Science and Applications*, p.10, DOI: 10.14569/IJACSA.2019.0101139.

Bibliography

Back, T. (1996) *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, Oxford.

Caux, C., Pierreval, H. and Portmann, M-C. (1995) *Genetic Algorithms and Their Application to Scheduling Problems (Les Algorithmes Genetiques Et Leur Application Aux Problemes D'ordonnancement)*, Apii Vol. 29, Nos. 4–5, pp.409–443.

Deb, K. (2000) 'Introduction to selection', in Bäck, T., Fogel, D.B. and Michalewicz, Z. (Eds.): *Evolutionary Computation 1: Advanced Algorithms and Operators*, Institute of Physics Publishing, Bristol and Philadelphia, 331 Pages.

Echandouri, B., Omary, F., Ziani, Fatima, E. and Sadak, A. (2018) 'SEC-CMAC: a new message authentication code based on the symmetrical evolutionist ciphering algorithm', *International Journal of Information Security and Privacy*, Vol. 12, pp.16–26, 10.4018/IJISP.2018070102.

Bounsaythip, C.K. (1998) *Algorithmes heuristiques et évolutionnistes*, Thèse de doctorat université de Lille, le 9 octobre 1998.

Menezes, A.J., Van Oorschot, P.C. and Vanstone, S.A. (1997) *Handbook of Applied Cryptography*, CRC Press, Canada.

Mühlenbein, H. and Schlierkamp-Voosen, D. (1993) 'Predictive models for the breeder genetic algorithm-I, continuous parameter optimization', *Evolutionary Computation*, Vol. 1, No. 1, pp.25–49.

Shaul, D. (2014) 'Evolutionary algorithms', *Encyclopedia of Computational Neuroscience*, pp.1–7.