A New Cryptosystem using Vigenere and Metaheuristics for RGB Pixel Shuffling

Zakaria KADDOURI[1]
Laboratory of Computer Science Research, Department of Computers Science,
Mohammed V University Agdal – AbuDhabi, AbuDhabi, United Arab Emirates

Mohamed Amine Hyaya[2]
Physics Department
Mohammed V University – Faculty of Sciences Rabat,
Rabat, Morocco

Mohamed KADDOURI[3]
LMPHE Laboratory
Mohammed V University – Faculty of Sciences Rabat,
Rabat, Morocco

***Abstract*—In this article we present a new approach using Vigenere and metaheuristics to resolve a problem of pixel shuffling to cipher an image. First the image is adapted to match the resolution system by transforming it to a list of intensities and coordinates. The idea is to use Vigenere encryption to maximize the confusion by widening the domain of intensities. Then, metaheuristics play the major role of encryption, generating an appropriate Meta key in order to shuffle the lists. Thus, both Vigenere key and Meta-key are used for encryption and later in decryption by the recipient. Finally, a comparison of different metaheuristics is proposed to find the most suitable one for this cryptosystemt.**

***Keywords*—Cryptography; cryptosystem; Vigenere; metaheuristics; image; pixel shuffling**

I. Introduction

In theory, a combinatorial optimization problem can be defined by all its instances. In practice, the problem is reduced to mathematically solving one of these instances, by the algorithmic method [4]. Metaheuristics are a family of optimization algorithms that aim to solve general classes of mathematical problems by combining search procedures to quickly find the best solution.

In 2005, a new encryption system [16], called SEC (Symmetrical Evolutionist-based Ciphering) was introduced, which is strongly linked to evolutionary algorithms and represents the first adaptation of a metaheuristic to the domain of cryptography. Its principle consists in constructing lists containing the different positions of the characters of a plaintext, and it connects the evolutionary processes (evaluation, selection, crossing and mutation operator) applied on the order of these lists to obtain a maximum disorder without modifying their contents. At the end of the algorithm, a key known as “gene key” is generated and used for both encryption and decryption operations [7]-[9], [13]-[15].

In our approach, we used multiple metaheuristics to find a strong key for our encryption. Metaheuristics can be divided into two main groups: 1) Single Solution Algorithms; and 2) Population-based Algorithms.

A. Single Solution Algorithms

Single Search Algorithms, i.e. local and global searches, start with a random solution then tries to optimize it, following a given criteria. Various Algorithms are actually used and improved, such as Hill Climbing (**HC),** which is classified among Local Searches. It optimizes the solution following the highest lean in its neighborhood [10], [18], [19], Simulated Annealing (**SA)**, is a global search based on Monte Carlo methods [5], [20]. This algorithm avoids local optimums by choosing a less optimal solution if the aspiration criteria is met [3], and finally the Taboo search (**TS)**, is also a global search, that escapes the local optimums by memorizing a list of previous solutions and selecting only unexplored solutions [4], [5], [9].

B. Population-based Methods

Contrarily to the previous methods population based algorithms optimize multiple solutions simultaneously. Among many, Genetic Algorithm (**GA)** uses natural selection. It combines individuals from the initial population to give birth to the next generation of solutions then, only the fittest ones are chosen to reproduce and create the next one and so on [2], [5], [7], [10], while particle swarm optimization (**PSO)** is developed on swarm behavior of birds. The initial population is created then a goal is set, unlike **GA**, **PSO** is not eliminating individuals, but each individual evolves differently so the whole group would reach the goal in an optimal way [3]. Back to **GA** one can say that even a normal individual may have more room for improvement than the fittest, Memetic algorithm (**MA)** solves this problem since it uses a local search to optimize every solution (one individual) before choosing the fittest. **MA** is a hybrid algorithm, a population based method using a local search to optimize intermediate solutions [5], [8], [10], [17].

Section II describes the proposed approach, including the methods used to optimize the cyphering or adapt different components of the cryptosystem.

Section III, shows both qualitative and quantitative analysis conducted on our cryptosystem.

Finally Section IV, discusses briefly the proposed algorithm and potential optimizations.

II. Our Cryptosystem

A. Description

The cryptosystem generates a symmetrical encryption Key [12]. The main idea is to shuffle the colors of the image "M" with dimensions $(W \times H)$, using the generated key. First, we apply a preliminary encryption using Vigenere algorithm [18], directly on the RGB image [6], i.e. a random key is generated, as in vigenere encryption the key is repeated until it reaches the length of element to be encrypted. In our case, the ASCII number of each character of the key will be added to the RGB values of a pixel. Then the RGB components of the image are placed vertically, getting a (Wx3H) grayscale image "M'". The initial solution is created given as a Table "X" of 256 intensities, to each value we assign a list "L" of coordinates of that value in the image V. These lists will help reconstruct the image. The shuffling starts by permuting intensities, i.e.

$$X_0 = \{0,1,2,3,\dots,253,254,255\} \rightarrow$$
$$X_f = \{251,149,50,61,\dots,13,22,200\} \qquad (1)$$

The best solution is chosen using metaheuristic algorithms and evaluated by the evaluation function "f":

$$f(i) = \sum_{j=0}^{255} |X_i[j] - X_0[j]| \qquad (2)$$

Finally, the encrypted image is reconstructed using the new list of intensities and assigning them to the coordinates stored at beginning, i.e. Let Li from (1) be the best solution, then all coordinates initially black (intensity=0) will be assigned the value 251, and all coordinates with intensity 1 will be assigned, 149, as for the remaining 254 intensities.

B. Skeleton of our Cryptosystem

Let M be the RGB matrices of the image to be encrypted, with $M(x,y,z)$ a pixel of the image M $such\ as$ $x \in \{0,1,2,\dots,W\}$, $y \in \{0,1,2,\dots,H\}$ and $z \in \{0,1,2\}$.

We create a list of random values to be our Vigenere Key

$$0 \leq\ V(w) \leq 255\ with\ w \in \{0,1,\dots,V_l\ \} and\ 30 \leq V_l \leq 50$$

M' is the encrypted image using Vigenere Key V as follows:

$$\forall x, \forall y, M'(x,y,0) = [M(x,y,0) + V((y + x.W) mod\ V_l)] mod\ 256$$
$$\forall x, \forall y, M'(x,y,1) = [M(x,y,1) + V((y + x.W) mod\ V_l)] mod\ 256$$
$$\forall x, \forall y, M'(x,y,2) = [M(x,y,2) + V((y + x.W) mod\ V_l)] mod\ 256$$

I is a grayscale image made of vertical concatenation of RGB matrices:

$$\begin{gathered} I(x,y) = M'(x,y,0) \\ I(x+H,y) = M(x,y,1) \\ I(x+2H,y) = M'(x,y,2) \end{gathered} \qquad (3)$$

I is then represented using lists of different Intensities, each list contains the (x,y) coordinates of a given intensity, element of the set $\{0,1,2,\dots,255\}$. We denote by L_i $(0 \leq i \leq 255)$ a list of the different positions of the Intensity and X_{iter}: A list of all intensities in a given iteration.

The goal is to create a maximum disorder between intensities in a manner that the difference transcends a given threshold. Metaheuristics are used to generate a random key while maximizing to a certain degree the disorder in X. We denote X_{final}.

To cipher the image, we reconstruct it using the order of intensities in X_{final} for example (Table 1):

TABLE I. Example of List Permutations

X_0	Intensities	0	1	2	…	253	254	255
	Coordinates	L_0	L_1	L_2	…	L_{253}	L_{254}	L_{255}
X_{final}	Intensities	251	168	59	…	2	112	15
	Coordinates	L_0	L_1	L_2	…	L_{253}	L_{254}	L_{255}

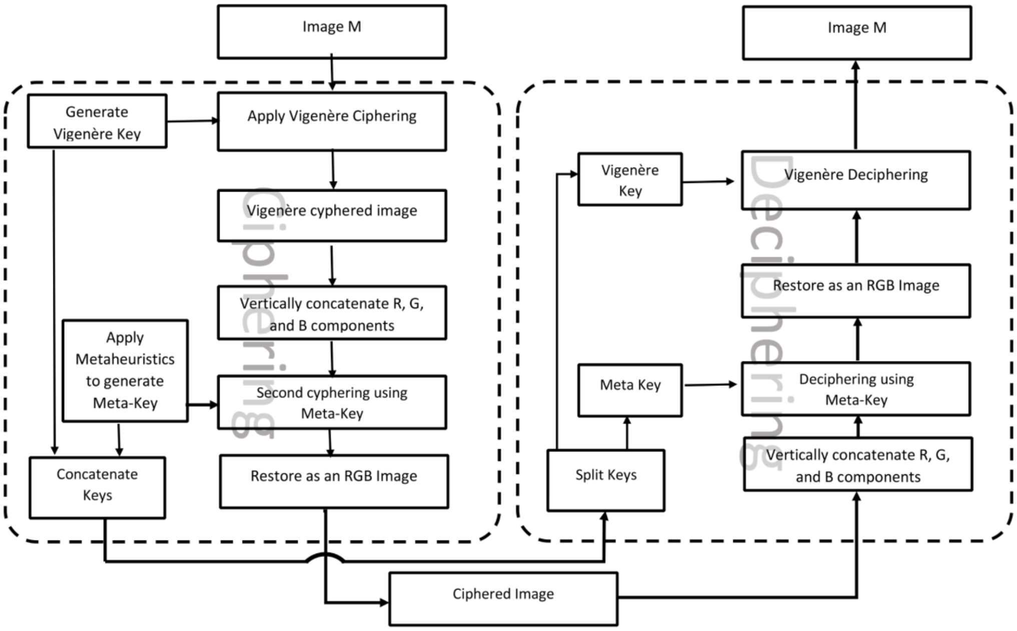
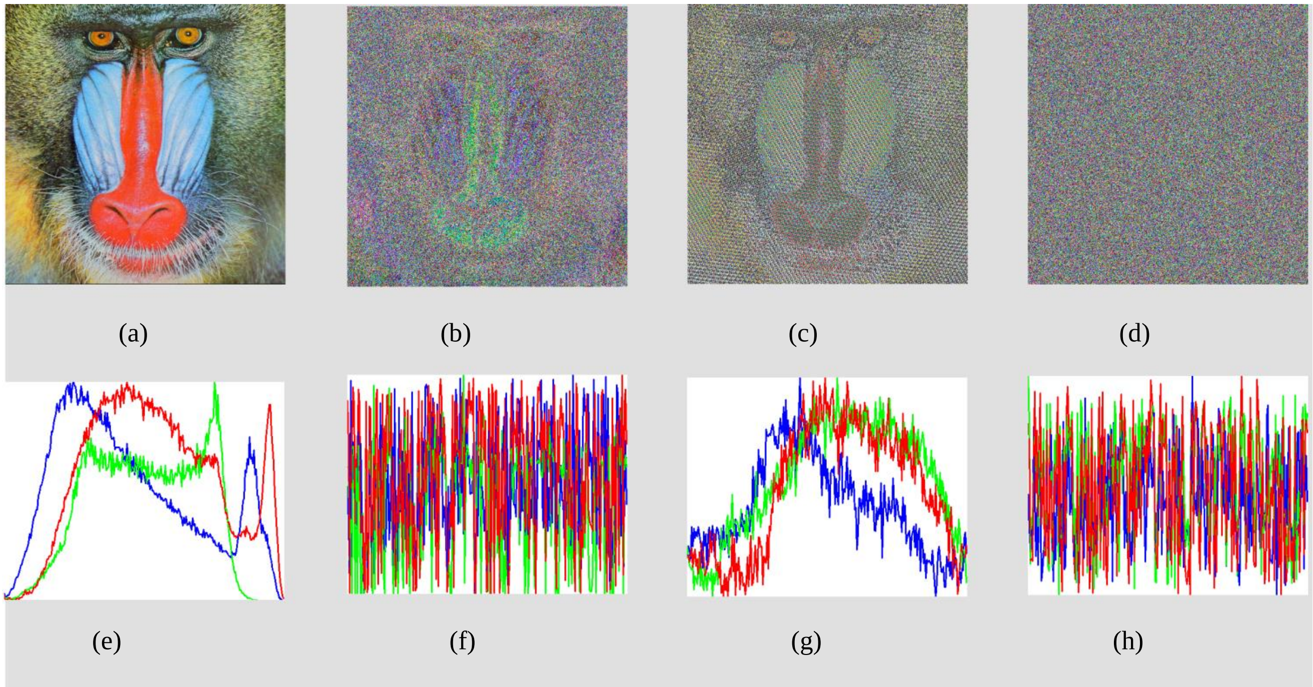


Fig. 1. Diagram of our cryptosystem.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 2. Baboon Image
(a) original , (b) pixel shuffling only, (c) Vigenere only, (d) proposed cryptosystem,
(e) Original histogram, (f) pixel shuffling histogram, (g) Vigenere histogram, (h) cryptosystem histogram.

X0 is the table containing the initial order of intensities and coordinates. The permutation only affects the intensities. As a final result (see X_{final}l) all the pixels that initially were pitch black "0" will be assigned the intensity 251 and pixels containing 1 will receive a value of 168 and so on. Fig. 1 summarizes our cryptosystem. During the encryption, the plain image is ciphered, using a randomly generated vigenere key, to enlarge the domain of colors to be shuffled. Then, Red, Green and Blue channels of the resulting image are separated and concatenated vertically, forming a grayscale like image. At this stage, a list of intensities is derived from the grayscale image. A single solution metaheuristic takes the list to be the initial solution, while population based metaheuristics, derive the initial population using random permutations on that list. At the end of the optimization, the solution returned, is called Meta-key, it allows the permutation of intensities as described previously. Finally, the RGB image is restored by rebuilding a three channels image by dividing the cyphered grayscale image. Decryption, is following the same methods except that both meta-key and vigenere key are shared. Grayscale image is constructed from the cyphered image. Then intensities get permuted using meta-key. Next, RGB image is restored and finally we use vigenere key to get the Plain Image.

III. Experimental Results

In this section, we use a benchmark image to study the efficiency of our cryptosystem, where we compare multiple metaheuristics including both local searches and population-based Algorithms.

A. Visual Tests

In Fig. 2 and 3, we propose an explanation for combining both Vigenere and metaheuristic keys, as one can observe the images ciphered by Vigenere and meta-key separately still recognizable by humans. First Vigenere encryption concatenates the key line by line, and changes the colors using the same pattern. If it encounters a big spot containing the same color, the patterns can be easily found (Fig. 2(c) and 3(c)). The same issue occurs for metaheuristic encryption, since the algorithm only permutes the colors. In consequence, we observe an image with similar forms but with different colors as seen in Fig. 2(b) and 3(b). Besides, the image encrypted by the combination of both algorithms is totally unrecognizable (Fig. 2(d) and 3(d)). In fact, Vigenere widens the domain of colors, breaks the contours of the image and adds a strong noise to the spots of similar colors, allowing the metaheuristics to permute intensities and ensure a maximum disorder in the final image. This can be observed by comparing histograms in Fig. 2 and 3 (e-f).

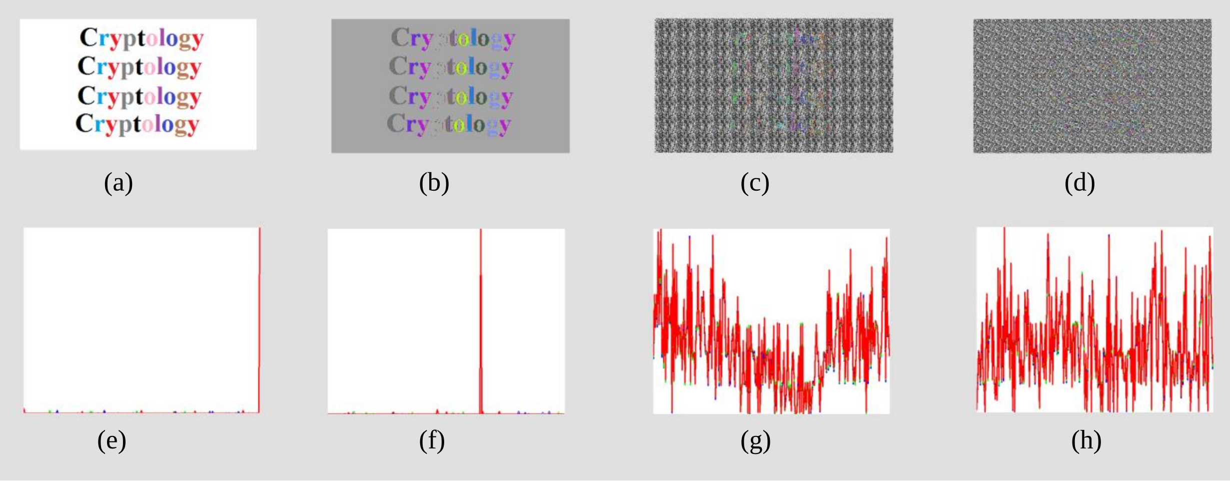


Fig. 3. Cryptology Image
(a) original , (b) pixel shuffling only, (c) Vigenere only, (d) proposed cryptosystem,
(e) Original histogram, (f) pixel shuffling histogram, (g) Vigenere histogram, (h) cryptosystem histogram.

B. Quality Tests

1) NPCR

$$NPCR = \frac{1}{m\,n}\sum_{i=1}^{m}\sum_{j=1}^{n}\delta_{I_0(i,j)}^{I_C(i,j)}$$
$$;with\ \delta: Kronecker\ delta$$

The number of pixel change rate (NPCR) is usually used to evaluate the absolute number of pixels change rate [21]. The more pixels change, the closer to 1 we get. In our case, as we can see, Fig. 4 presents the NPCR values between the original and ciphered image, for 10 different runs, the proposed algorithms gave nearly optimal values of NPCR.

However, this maximal value would also involve a binary image and its negative, the last, can be easily recognized. Thus, NPCR proves only that pixels of the original image changed, but it may still be recognizable. This is why we must perform PSNR to evaluate the noise ratio in the ciphered image and SSIM for similarity between the ciphered and the original image.

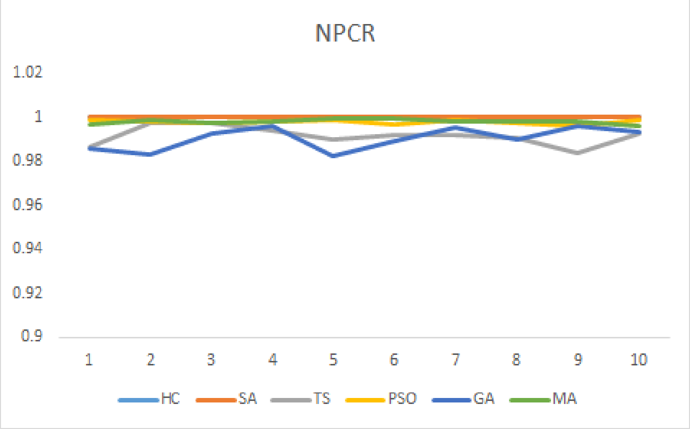


Fig. 4. NPCR values for different metaheuristics.

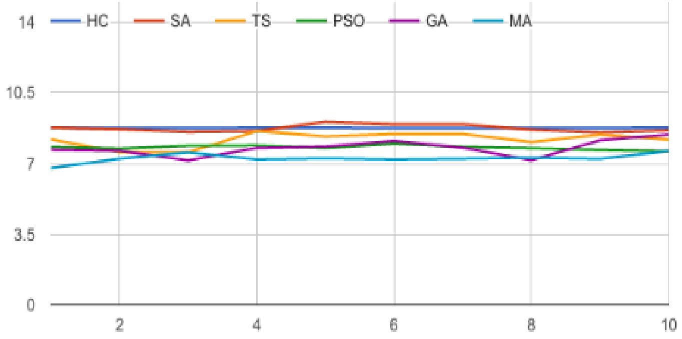


Fig. 5. PSNR values for different metaheuristics.

2) PSNR

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right)$$

$MSE = \frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}(I_0(i,j) - I_c(i,j))^2$ Peak signal to noise ratio (PSNR) is calculated to measure distortion in a digital image by calculating the amount of noise in the image [1]. The smaller the value of PSNR, the less signal is conserved. Let us consider two identical images and add 1 to one component of one pixel to the second image. The PSNR of those two images is going to be the highest possible after infinity, PSNR of two identical images. If we apply the previous condition to the size of our benchmark image:

$$PSNR_{max} = 102.3162028\ dB$$

While $PSNR_{min} = 0\ dB,$ considering the log scale, all ciphered images offering a $PSNR < 10\ dB$ can be considered a good encrypted image. We summarize the values of PSNR given by the experiment, previously described in Fig. 5.

3) SSIM

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2cov_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The Structural Similarity index map (SSIM) computes a similarity map between two digital images “x” and “y” as confirmed by [11] it allows simulating human perception in comparing two images. The map value $-1 \leq \mathrm{SSIM(i,j)} \leq 1$ where one means images are similar around that region. Thus for two identical images, all map values equal one. Meanwhile, negative values attest inverted regions. Finally, zero states totally different regions. Fig. 6 is computed as follows:

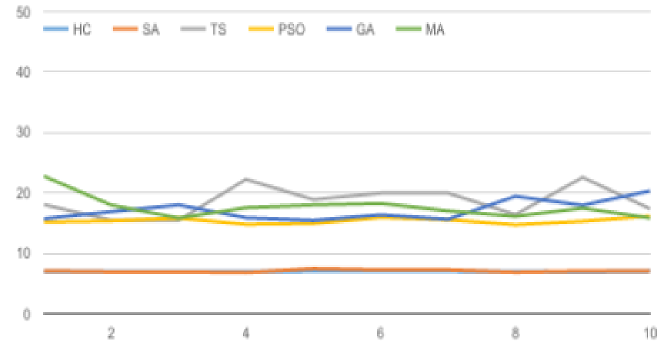


Fig. 6. SSIM values for different metaheuristics.

$$\overline{SSIM} = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}|SSIM(i,j)| \times 100\%$$

This means that the computed value (in percent) for two completely different Images is 0% while, 100% implies either identical images or an image and its negative.

All values obtained in this experiment are below 25%.

4) Encryption time

This time is actually is for the whole cryptosystem including key generation (Vigenere & Meta key) and pixel shuffling. We observe that the encryption is very fast since the size of the image used is 300KB. For example, in the case of HC metaheuristic, the encryption rate is (17 554 285 bytes/sec). We can notice in Fig. 7 that the execution time for GA and MA is too high compared to the other metaheuristics, but this is due to multipoint crossover that needs to eliminate duplicates every time it generates a child.

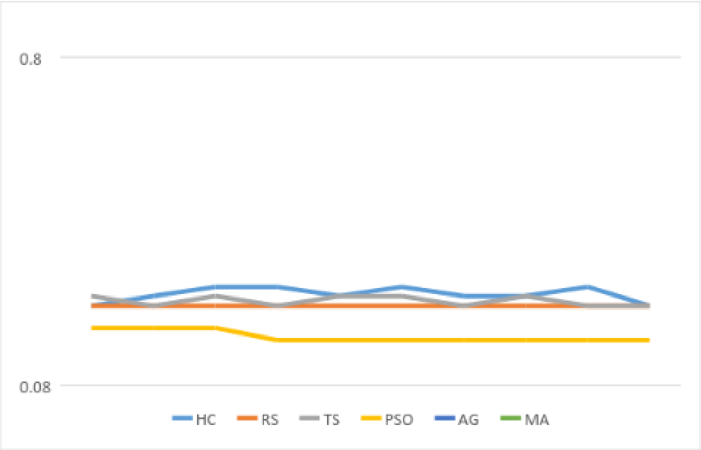


Fig. 7. Encryption time for different metaheuristics.

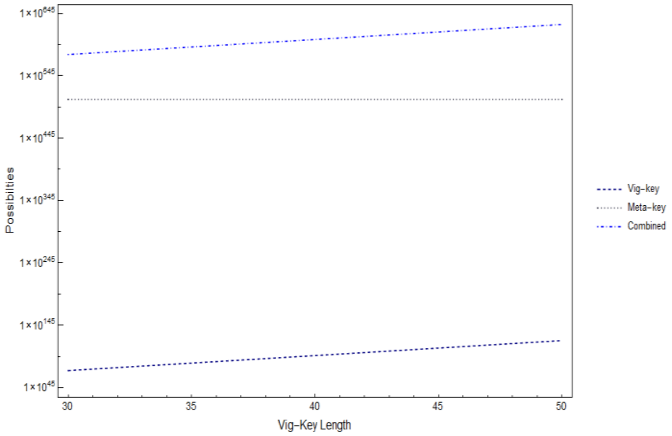


Fig. 8. Number of possible Keys.

5) Key strength

This cryptosystem proposes two complementary and symmetric encryptions. The final key to be shared is a simple concatenation of both keys. The challenge for breaking the key is that the generated key is totally independent from the image, and the length of the key is not fixed since Vigenere key length is between 30 and 50 Bytes. In addition, the Meta-key is 256 Bytes Long, which give us:

$$286 \leq key\ length \leq 306\ Bytes$$

The total number of combination is given by:

$$N = 256^{k} \times P_{256}^{256}; kVigenere\ key\ length$$
$$since: 30 \leq k \leq 50 \Rightarrow 8.92 \times 10^{575} \leq N \leq 4.13 \times 10^{622}$$

Fig. 8 is a semilog graph showing the number of possible keys for Vigenere key, Meta key separately and the combination of both versus Vigenere key length.

Table 2 gives the means and standard deviation obtained after 10 runs on each metaheuristics. We observe that the numbers are very stable and consistent since the runs were not selected. The experiment is totally independent: no seeds were planted for the pseudorandom generator. The values side by side are quite similar except for SSIM, the values vary from 7 to 15%.

TABLE II. QUALITY TESTS SUMMARY

	NPCR			**PSNR**			**SSIM**		
	μ	±	σ	μ	±	σ	μ	±	σ
HC	0.999909	±	0.00005	8.773	±	0.007	7.007	±	0.035
SA	0.999997	±	0.00000	8.752	±	0.171	7.078	±	0.200
TS	0.991533	±	0.00404	8.197	±	0.350	18.627	±	2.561
PSO	0.997794	±	0.00092	7.804	±	0.108	15.354	±	0.487
GA	0.990335	±	0.00495	7.780	±	0.388	17.146	±	1.715
MA	0.997813	±	0.00104	7.264	±	0.213	17.669	±	2.007

IV. Discussion and Conclusions

The proposed algorithm reveals very satisfying results. Overall, it is compatible with all the tested metaheuristics. However the parameters have to be set for every metaheuristic to obtain good results, but once set, the results are stable and render the encrypted image unrecognizable. On the other hand, if we compare the proposed metaheuristics to choose the best one(s), NPCR rates Simulated Annealing and Tabu search as the best ones, While PSNR values give a slight preference for population based algorithms, MA, GA and PSO, besides HC and SA, outclass the other metaheuristics according to SSIM. As for the encryption time all the metaheuristics except MA and GA, are very fast. Moreover, the key generated offers a high security level compared to the existing symmetrical cyphers. Despite being very satisfying, the algorithm is very flexible and allows many ameliorations. For instance, improving Vigenere encryption part or choosing different evaluation function.

References

[1] Ahmad, J., & Ahmed, F. (2010). Efficiency analysis and security evaluation of image encryption schemes. computing, 23, 25.

[2] Davis, L. (1991). Handbook of genetic algorithms.

[3] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.

[4] Glover, F. (1989). Tabu search—part I. ORSA Journal on computing, 1(3), 190-206.

[5] Glover, F. W., & Kochenberger, G. A. (Eds.). (2006). Handbook of metaheuristics (Vol. 57). Springer Science & Business Media.

[6] Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc..

[7] KADDOURI, Z., OMARY, F., & ABOUCHOUAR, A. (2013). BINARY FUSION PROCESS TO THE CIPHERING SYSTEM" SEC EXTENSION TO BINARY BLOCKS". Journal of Theoretical & Applied Information Technology, 48(1).

[8] Kaddouri, Z., & Omary, F. (2014). New Symmetrical Ciphering Approach Based on Memetic Algorithm. algorithms, 1, 5.

[9] Kaddouri, Z. (2014). Mise en oeuvre de nouvelles techniques pour la sécurité informatique basées sur les algorithmes évolutionnistes et les fonctions de Hachage.

[10] Kumar, R., Tyagi, S., & Sharma, M. (2013). Memetic Algorithm: Hybridization of Hill Climbing with Selection Operator. International journal of Soft Computing and Engineering, 3(2), 140-145.

[11] Li, C., & Bovik, A. C. (2009, January). Three-component weighted structural similarity index. In IS&T/SPIE Electronic Imaging (pp. 72420Q-72420Q). International Society for Optics and Photonics.

[12] Mewada, S., Sharma, P., & Gautam, S. S. (2016). Classification of Efficient Symmetric Key Cryptography Algorithms. International Journal of Computer Science and Information Security, 14(2), 105.

[13] Mouloudi, A., Omary, F., Tragha, A., & Bellaachia, A. (2006, November). An Extension of Evolutionary Ciphering System. In Hybrid Information Technology, 2006. ICHIT'06. International Conference on (Vol. 1, pp. 314-321). IEEE.

[14] Omary, F. (2006). Application des algorithmes évolutionnistes à la cryptographie.

[15] Omary, F., Tragha, A., Bellaachia, A., & Mouloudi, A. (2007). Design and evaluation of two symmetrical evolutionist-based ciphering algorithms. IJCSNS International Journal of Computer Science and Network Security, 7(2), 181-190.

[16] Omary, F., Tragha, A., Lbekkouri, A., Bellaachia, A., & Mouloudi, A. (2005). An Evolutionist Algorithm to Cryptography. Lecture Series and Computational Sciences, 4, 1749-1752.

[17] Radcliffe, N. J., & Surry, P. D. (1994, April). Formal memetic algorithms. In AISB Workshop on Evolutionary Computing (pp. 1-16). Springer Berlin Heidelberg.

[18] Stinson, D. R. (2005). Cryptography: theory and practice. CRC press.

[19] Tomita, M. (1982). Dynamic construction of finite-state automata from examples using hill-climbing. In Proceedings of the fourth annual cognitive science conference (pp. 105-108).

[20] Van Laarhoven, P. J., & Aarts, E. H. (1987). Simulated annealing. In Simulated Annealing: Theory and Applications (pp. 7-15). Springer Netherlands.

[21] Wu, Y., Noonan, J. P., & Agaian, S. (2011). NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT), 31-38.