This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Encryption of text file using a user controlled automatically-generated key

PLEASE CITE THE PUBLISHED VERSION

<https://doi.org/10.1109/ICGS3.2019.8688258>

PUBLISHER

IEEE

VERSION

AM (Accepted Manuscript)

PUBLISHER STATEMENT

Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Shnishah, Halima, and David Mulvaney. 2019. "Encryption of Text File Using a User Controlled Automatically-generated Key". figshare. <https://hdl.handle.net/2134/11202047.v1>.

Encryption of Images using a User-controlled Automatically-generated Key

Halima Abdul Halim Shnishah, David Mulvaney

Abstract—Traditional symmetrical cryptographic algorithms generally provide an adequate degree of immunity to attacks aimed at revealing secret keys. A number of approaches exist for the automated generation of secret keys, but, for high security applications, some end users remain wary of approaches that are controlled by third parties. Consequently, there remains interest in certain high-security applications in being able to retain control over the method used for the generation of keys. In this paper, keys for both image encryption and decryption are obtained using the evolutionary computing tool Eureqa, in its modelling of pseudo-random input data. The secret keys generated by this approach and when applied to the encryption and decryption of gray-scale images are validated in a range of statistical tests, namely histogram, chi-square, correlation of adjacent pixel pairs, correlation between original and encrypted images, entropy and key sensitivity. Experimental results obtained from methods show that the proposed image encryption and decryption algorithms are secure and reliable, with the potential to be adapted to high-security image communication applications.

Keywords—encryption quality measurements, image encryption, Eureqa, automated key generation.

I. INTRODUCTION

Every day, large volumes of digital data are exchanged through public channels. Much of the data are private or confidential, but the use of public channels makes it vulnerable to seizure by an adversary or an eavesdropper. A range of security techniques is available, many of which require the use of a key known only to the sender and the intended recipient and so requires protection from intruder attacks.

The contention of the current research work is that Kirchhoff's principle [1], which states that a 'cryptosystem should be secure even if everything about the system, except the key, is public knowledge', does not necessarily follow if the key is generated and distributed by a third party, often a commercial entity, government institution or military establishment. This is because there is no guarantee that a key is not in some way retained by that third party or communicated to an 'authority'. Only if the end user retains control over key generation can there be peace of mind that an intruder has not compromised the process [2].

In this work, a method has been established that permits the

automatic generation of keys by a process controlled by the cryptography users themselves. The evolutionary computing tool Eureqa [2] has been used to generate each key from a unique mathematical equation, which in turn describes a purposely-generated random number sequence. A method for using Eureqa in the field of cryptography was originally proposed by Blackledge *et al.* [2], in which the authors attempted to establish that the method is able to produce keys that provide security of strength similar to that of existing popular methods.

The paper is organized as follows. Related work that has investigated the generation of bespoke keys using Eureqa is discussed in Section II and Section III introduces the proposed technique. Section IV describes the statistical and security analysis and Section V presents the conclusions.

II. RELATED WORK

Evolutionary computing in cryptography is generally carried out by generating an initial seed to initiate the calculation of a mathematical equation, which can then be used as a secret key in traditional encryption and decryption processes. The work presented by Blackledge *et al.* [2] models 250 samples of a pseudo-random number sequence using Eureqa, and was reported to take around 23 hours to generate a single mathematical key for encryption and decryption purposes, but no assessment of the security of the approach was made. This encryption work was adopted by Dlamini [3] in the implementation of a risk-based multi-factor authentication system to monitor user behavior in order to protect access to cloud-based services. Eureqa was used to generate an encryption function, but its calculation took over 205 hours on a standard desktop system.

In [4], Blackledge trained a neural network to approximate input noise with the aim of generating a nonlinear function. Atmospheric noise generated from radio emissions affected by lightening, electronic noise and radioactive decay was provided to Eureqa, which then generated an encryption function to model the data. The output of Eureqa was approximated with the input noise to generate a non-linear function which is the encryption algorithm. The security of such an approach was assessed by a range of metrics, including the Lyapunov exponent, cycle length, key diffusion and entropy. Algorithmic processing time was not recorded in the paper.

The extended processing time required for the generation of keys in [2] and [3] is not acceptable in practical environments. For real-time streaming applications, a practical approach is needed to ensure that keys are not only generated and applied in a reasonable time, but also that the security of the approach must be demonstrated using appropriate statistical tests.

Halima Shnishah is a PhD Student in the Wolfson School of Mechanical, Electrical and Manufacturing Engineering at Loughborough University, LE11 3TU, UK (email:H.shnishah@lboro.ac.uk).

David Mulvaney is a Senior Lecturer in the Wolfson School of Mechanical, Electrical and Manufacturing Engineering at Loughborough University, LE11 3TU, UK (email:d.j.mulvaney@lboro.ac.uk).

In this paper, a solution is proposed that provides automated keys generated using Eureka with a much reduced execution time. In addition, a range of statistical tests is applied to demonstrate the security of the resulting encryption system.

III. PROPOSED APPROACH

Eureka was first released in 2009 by Hod Lipson of the Computational Synthesis Laboratory at Cornell University [5]. Eureka is able to perform ‘symbolic regression’, namely the automatic execution of iterative search operations to determine an underlying equation that models the input data. The main benefit of the tool is its ability to reduce the time and effort needed to generate such equations compared with a trial and error approach [6].

In this work, Eureka is used to describe a pseudo-random number (PRN) using a mathematical equation that is then used to generate a secret key that is needed in the encryption process. Perhaps the two most important aspects of every encryption system are the time taken to perform encryption and the security of the encryption key, but there is often a practical trade-off that needs to be established between the two. In the current approach, the time taken to determine a suitable equation increases if the PRN sequence it represents is made longer, but the longer the key the better becomes the security of the encryption provided. In this work, Eureka is used to generate an equation from a short PRN sequence, reducing calculation time compared to previous work, yet the

equation obtained is used to generate a key of arbitrary length. It is the demonstration that the security provided by this modified approach remains acceptable that is the main contribution of this paper.

A linear congruential generator (LCG) has been used to generate a random number sequence $x_i \in \{0, 1, 2, \dots, m-1\}$, $i = 0, 1, 2, \dots, n$. A seed x_0 is required to initiate the generation of the first value in the sequence and subsequent values are generated iteratively. A PRN sequence of length n can be obtained using

$$x_{i+1} = (ax_i + c) \bmod m \quad (1)$$

where a and c are integers.

Three main steps are required in the implementation of a cryptographic system, namely key generation, encryption and decryption. The degree of security provided by a cryptosystem principally depends on the key generation mechanism. Here, the Dual Modified Extended approach (DMEX) is introduced to obtain three separate mathematical equations E_1 , E_2 and E_3 that are then used to generate keys that are applied successively. Although each equation is generated by Eureka from only five PRNs, reducing calculation time to a matter of seconds rather than hours, each is used to generate a much longer key in matrix format (so it is suitable for operation on images), resulting in the three keys K_1 , K_2 and K_3 . The manner in which these keys are applied to the images to be encrypted is shown in Figure 1 and is described below.

Fig. 1 Block diagram of the DMEX encryption approach

In DMEX, for each mathematical equation five PRNs are used as input to Eureka and an example of the form of equation produced is shown in equation (2).

$$y = 530.3 + 500.8 * \cos(4.33 - 0.047 * x - 0.386 * \cos(-21 * x)) - 0.071 * x - 2.466 * \cos(-21 * x) * \cos(4.33 - 0.0465 * x - 0.386 * \cos(-21 * x)) - 22.62 \quad (2)$$

Such equations can then be used to generate arbitrary lengths of sequences of random numbers to form keys. For an image of dimensions $M \times N$, the following $M \times N$ key matrices K_1 , K_2 and K_3 are obtained for forming the encrypted image as described below.

1. K_1 contains MN elements (as an $M \times N$, matrix), generated from equation E_1 . Each element of K_1 is exclusively-ORed in a pairwise manner with the corresponding element of the input image I_1 to generate a partially encrypted image I_2 .
2. K_2 contains MN values, has M identical rows where each row contains the first N values generated from equation E_2 .

Each element of K_2 is exclusively-ORed in a pairwise manner with each corresponding element of I_2 to generate a second partially encrypted image I_3 .

3. K_3 contains MN values, has N identical columns where each column contains the first M values generated from equation E_3 . Each element of K_3 is exclusively-ORed in a pairwise manner with each corresponding element of I_3 to generate the final encrypted image.

To implement decryption, the application of the steps simply takes place in reverse order.

IV. STATISTICAL AND SECURITY ANALYSIS

A quad-core Intel i5 CPU with 64-bit architecture running under CentOS is used to generate the mathematical equations (secret keys) and to perform encryption and decryption. In the experiments, gray-scale images of resolution 256×256 and of 8-bit depth were used as test data and statistical properties of

relevance to encryption quality were then obtained, namely histogram, chi-square and entropy analyses. Key security is also investigated.

In 1949, Shannon stated that, ‘It is possible to solve many kinds of ciphers by statistical analysis and a good cipher should be robust against any statistical attack’ [7]. To demonstrate the robustness of the proposed approaches, results from a range of statistical tests are reported below.

Histogram analysis

The performance of an image encryption algorithm can be assessed by how closely the histogram of the encrypted image conforms to a uniform distribution [8]-[10]. Examples of histograms obtained for an unencrypted original image and a DMEX encrypted image are shown in Figure 2.

Fig 2. Histograms of original and DMEX encrypted images

Visually, the distribution in (d) for the DMEX encrypted images appears close to uniform, a distribution which is generally known to make statistical attacks more difficult in practice.

Chi-squared test

The uniformity between the outputs obtained from an encryption algorithm can be assessed by the chi-squared test given by

$$\chi^2 = \sum_{i=1}^N \frac{(v_i - N)^2}{N} \quad , \quad (3)$$

where, for the current application, v_i represents the frequency of occurrence of each of the N gray-scale values. In the literature, it is reported that the lower the chi-square value obtained, then the better is the uniformity of the image, indicating a higher degree of encryption [9]. The chi-square values obtained in Figure 2 were 41145 for the original Lena image and 258 for the encrypted image. The chi-square values for a range of different images both in their original and encrypted forms are shown in Table 1. The table also compares the performance of DMEX with two state-of-the-art

key generation methods, the first using an evolutionary approach [11] and the second using a technique based on chaos [12]. It can be seen that the DMEX approach exhibits a chi-square test value that is comparable with those found in the other recent approaches.

TABLE I
COMPARISON OF CHI-SQUARED TEST VALUES

Image name	Original image	Chi-squared test values		
		DMEX	Ref [11]	Ref [12]
Lena	41145	258	182	263
Cameraman	113650	291	234	257
Baboon	107480	289	241	266
Peppers	36778	282	218	274
House	299790	285	255	260
Splash	86024	294	211	271
All zero	16711680	306	220	-
Mean	2485221	285	223	265

Correlation between original and encrypted images

This statistical test is used to assess in a quantitative manner the difference between two images and can also be used to indicate the quality of the encryption method. The correlation between an original image A and an encrypted image B can be determined from equation (4) [11].

$$\rho = \frac{\sum_{i=1}^M \sum_{j=1}^N (A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\left(\sum_{i=1}^M \sum_{j=1}^N (A_{ij} - \bar{A})^2\right) \left(\sum_{i=1}^M \sum_{j=1}^N (B_{ij} - \bar{B})^2\right)}} \quad (4)$$

$$\bar{A} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N A_{ij} \quad \text{and} \quad \bar{B} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N B_{ij}.$$

The results obtained are shown in Table 2. Values of ρ close to unity are obtained for image pairs that are similar and values near zero indicate little similarity between images. As shown in Table 2, the correlation values ρ for DMEX and those published for the evolutionary approach in [11] are both close to zero. Again, the encryption performance of DMEX can be seen to be similar to that of another modern approach.

TABLE 2
COMPARISON OF ENCRYPTION QUALITY USING CORRELATION

Image name	Image correlation ρ	
	DMEX	Ref [11]
Lena	0.0032	0.0034
Cameraman	0.0063	0.0015
Baboon	0.00045	-0.00070
Peppers	0.0046	0.0024
House	-0.0021	-0.0028
Splash	0.00069	-0.0033
All zero	7.99	7.99
Mean	7.99	7.99

Entropy measurement

Entropy can be used to assess the random nature of the output values produced by an encryption algorithm and can be calculated using

$$H(s) = \sum_{i=0}^{2^M-1} P(s_i) \log_2 \frac{1}{P(s_i)} \quad , \quad (5)$$

where $P(s_i)$ represents the probability of being in state s_i and 2^M is the total number of states of the message source. A comparison of the entropy of the DMEX approach with other state-of-the-art algorithms is presented in Table 3.

TABLE 3
COMPARISON OF ENCRYPTION ENTROPY

Image name	Original image	Encryption entropy		
		DMEX	Ref [11]	Ref [12]
Lena	7.44	7.9992	7.9980	7.9977
Cameraman	6.97	7.9970	7.9974	7.9969
Baboon	7.99	7.9974	7.9973	7.9970
Peppers	7.53	7.9971	7.9976	7.9973
House	6.49	7.9969	7.9972	-
Splash	7.29	7.9969	7.9977	-
All zero	0	7.9973	7.9976	-
Mean	6.24	7.9974	7.9975	7.9972

The high value of entropy for the DMEX approach (close to

the maximum value of eight), indicates that the approach is robust against entropy attacks. The combination of high entropy and high chi-squared test value obtained for DMEX indicates that the approach provides similar security to other modern encryption algorithms.

Key sensitivity

During both the encryption and decryption processes, an encryption algorithm should provide high sensitivity with respect to changes in the value of the secret key [8]. For the encryption process, high sensitivity means that a substantially different encrypted image is obtained when there is only a minor change made to the secret key. It should not be possible to fully or partially retrieve the original encrypted image should a small change be made to the encryption key [13]. The images obtained during key sensitivity tests for DMEX are shown in Figure 3.

Fig. 3 Key sensitivity analysis results

The encrypted version of the original image is shown in Figure 3(b) and the decryption of the original image following changes to the first, second and third seeds are shown in (c), (d) and (e) respectively. Similarly, performing decryption following minor changes to the first, second and third encryption equations are shown in (f), (g) and (h) respectively. With respect to (f), (g) and (h), it is clear that in all cases the secret keys used in the DMEX approach exhibit high sensitivity.

V. CONCLUSION

In this paper, a new DMEX approach has been introduced and its application to image encryption demonstrated. To assess the robustness of DMEX in the face of attacks, a range of statistical tests has been performed using histogram

analysis, chi-squared test, correlation coefficient, information entropy and key sensitivity. It was found that DMEX exhibited encryption security performance similar to that found in the literature for other modern algorithms. The histogram results showed an almost uniform distribution, indicating low vulnerability to statistical attacks, and the values obtained for entropy indicated a uniform distribution of energy between pixels as expected of high-performance encryption algorithms. DMEX exhibits low correlation between adjacent pixels of the encrypted image, indicating strong resistance to statistical attacks. The proposed approach has also been demonstrated to be highly sensitive to minor changes to the secret key values. Overall, DMEX has been demonstrated to be a viable alternative method of generating keys that can be guaranteed to be known only to the

encryption user. DMEX also allows the calculation of keys in a time much reduced compared to its predecessors, while at the same time providing security at a level similar to other modern approaches.

REFERENCES

- [1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, 27, 3, pp.379-423, 1948.
- [2] J. Blackledge, S. Bezobrazov, P. Toibin, and F. Zamora, "Cryptography Using Evolutionary Computing," *ISSC13 Irish Signal Syst. Conf.*, pp. 1-9, 2013.
- [3] M. T. Dlamini, "Securing Cloud Computing's Blind-spots using Strong and Risk-based MFA," *CONF-IRM 22*, 2016.
- [4] J. Blackledge, S. Bezobrazov and P. Tobin, "Cryptography using Artificial Intelligence," *IJCNN*, pp.1-6, 2015.
- [5] L. Edwards, "Eureqa, the robot scientist," [Online]. Available: <http://phys.org/news/2009-12-eureqa-robot-scientist-video.html>. [Accessed: 9-Aug-2017].
- [6] D. Salisbury, "Robot biologist solves complex problem from scratch," 2011. [Online]. Available: <http://news.vanderbilt.edu/2011/10/robot-biologist/> [Accessed: 27-Aug-2017].
- [7] C. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, 28, 4, pp.656-715, 1949.
- [8] X. Wang, J. Zhao, and H. Liu, "A new image encryption algorithm based on chaos," *Opt. Commun.*, 285, 5, pp.562-566, 2012.
- [9] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals*, 21, 3, pp.749-761, 2004.
- [10] N. K. Pareek, V. Patidar, and K. K. Sud, "Diffusion-substitution based gray image encryption scheme," *Digit. Signal Process. A Rev. J.*, 23, 3, pp.894-901, 2013.
- [11] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, and M. R. Mosavi, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimed. Tools Appl.*, 71, 3, pp.1469-1497, 2014.
- [12] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Opt. Commun.*, 285, 1, pp. 29-37, 2012.
- [13] X. Chai, Z. Gan, K. Yang, Y. Chen, and X. Liu, "An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations," *Signal Process. Image Commun.*, 52, pp. 6-19, 2017.