Volume 57, Number 4, 2016 ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

EFFICIENT CHAOS-BASED CRYPTOSYSTEM FOR REAL TIME APPLICATIONS

Jean De Dieu NKAPKOP[1)], Joseph Yves EFFA[1)], Monica BORDA[2)], Anton CIPRIAN[2)], Laurent BITJOKA[3)] Alidou MOHAMADOU[4)]

[1)]The University of Ngaoundéré, Faculty of Science, P.O. Box 454 Ngaoundéré, Cameroon
[2)]Technical University of Cluj-Napoca, 26-28 Baritiu Street, 400027, Cluj-Napoca, Romania
[3)]National School of Agro-Industrial Sciences, University of Ngaoundere, 454, Cameroon
[4)]Department of Physics, Faculty of Science, University of Maroua, P.O.Box 46, Maroua, Cameroon
Email:jdd.nkapkop@gmail.com

Abstract: Nowadays depending on the application domain, the need to protect images against unauthorized users has become a challenge. We previously developed and implemented using Matlab software, a new image encryption algorithm that uses the permutation-diffusion structure. The permutation step was used to shuffle the positions of the pixels within the image, while the diffusion process was employed to improve the security by masking all the pixel values of the image. In this paper we present an implementation of this chaos-based algorithm under Qt environment using the C/C++ programming language to show the practical aspects of its implementation. This implementation also provides the opportunity of selecting three encryption stages to encrypt and decrypt any type of input images. Thus, this allows us to see that the permutation-diffusion structure is required for a secure encryption. In order to prove the efficiency of this algorithm a comparison of the encryption time has been made with the other similar algorithms implemented on Matlab or C/C++. The results show that the proposed approach performs faster with higher security level for the encryption task.

Keywords: *Chaos-based algorithm, image encryption; Graphical User Interface; C/C++ programming language.*

I. INTRODUCTION

In all public telecommunication networks, such as TV cable, mobile phones, satellite, the Internet, etc., digital image encryption became a field that is more increasingly important in many applications as diverse as military imaging systems, online personal photographic album, medical imaging systems, confidential video conferencing, etc. Thus, it is necessary to encrypt these types of data before transmission over the public network to preserve its security and prevent unauthorized access [1]. Image encryption is the term used for conversion of an image from its original form to ciphered form, while retrieving the original image from its cipher form is called Image decryption. Cryptanalysis is used to break the code and deduce a specific plain text or the key being used. All future and past images encrypted with that key are compromised [2].

To encrypt an image, an encryption algorithm is directly used, which has led to the development of a various number of theories based on encryption techniques [3,4]. However, for many image applications, real time, fast, secure and reliable monitoring is an essential requirement. For this purpose, many traditional encryption algorithms can be used [5], but some of these algorithms are hard to understand, slow for encryption and not suitable for real time applications. Also, their implementation, when they are realized by software is even more complicated because of the high correlation between the image pixels. Therefore, there was a lot of work needed to be done for the development of non-traditional encryption methods. As a result, a new concept of a chaotic system arose for highly secure, fast and easily implemented encryption systems for secure transmission networks [6-8].

Chaos-based algorithms have shown exceptionally superior properties in aspects such as security and complexity [9] due to some interesting intrinsic features of chaotic maps such as good pseudo-randomness, ergodicity, non-periodicity, non-convergence, unpredictability and extreme sensitivity to initial conditions and system parameters. They are considered the favorable trade-off of the digital image encryption between the security and the speed. We can find a brief description of the relationship between chaos and cryptography in [10]. Fridrich in [11], proposes an image encryption scheme based on three two-dimensional discrete chaotic maps, namely the Baker map, Standard map and the Cat map. To fulfill the request of Shannon's theory, the architecture of the scheme is composed of the confusion and diffusion mechanism, which is seen as the basic structure for chaos-based image encryption algorithms. Afterwards, various chaotic algorithms based on that architecture have been proposed for securing image applications [6-9, 12-20]. However, as chaotic systems are assumed to work in the real number domain, some of them lead to a longer processing time due to the repetition of encryption rounds to achieve a satisfactory security level. On the other hand, its randomness behavior will be deteriorated when a finite precision with fixed-point arithmetic is used.

To avoid the cyclic digitization of chaotic numbers in the generation of permutation and diffusion keys and thus

Manuscript received October 31, 2016; revised November 21, 2016

achieve high speed as well as high security level performances, we have carried out the development of a one round chaos-based image encryption scheme based on the fast generation of large permutation and diffusion keys with a good level of randomness and very high sensitivity on the keys [21]. By using Matlab software, simulation results have indicated that the developed algorithm could have satisfactory speed and high level of security if implemented in other platforms. Therefore, this paper presents the design of chaos-based image encryption software for the practical evaluation of the computation time under the Qt development environment using C/C++ as a programming language. In addition, it provides the opportunity of selecting three encryption stages to encrypt and decrypt any type of images input.

The rest of this paper contains in Section 2, a brief recall of the proposed cryptosystem. In Section 3, an implementation of this cryptosystem under Qt environment is proposed. In Section 4, the experimental results of encryption speed and the security are discussed. Finally, the conclusions are made in the last section.

II. PROPOSED CRYPTOSYSTEMS

We briefly recall the previously developed encryption algorithm [21]. The synoptic of the developed chaos-based image encryption scheme is shown in Figure 1.

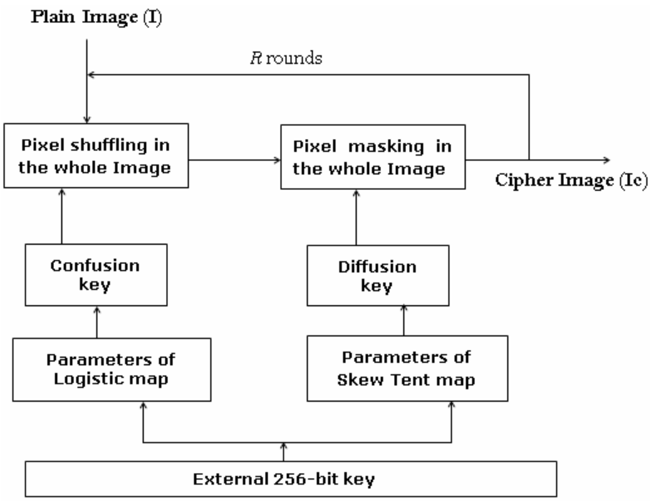

Figure 1. Synoptic of the developed chaos-based image encryption scheme.

As shown in the synoptic, the developed technique is a symmetric one-round chaos-based image encryption which relies on permutation-diffusion scheme. Therefore, the overall architecture of our cryptosystem was divided into two stages: permutation of pixels in the image, and the masking of the pixel values of the image as a whole.

Due to bulky data capacity and high correlation among pixels in image files, the permutation process is needed in order to decorrelate the relationship between the adjacent pixels. For this purpose, in this technique, we have proposed to use the integer sequences obtained by the sorting of chaotic Logistic Map (LM) by ascending order as the permutation key to shuffle the whole image. This technique has allowed us to avoid the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial conditions or control parameters is increased, as the true accuracy of the computer has been exploited. However, this stage as we will see, does not guarantee a good level of security. The diffusion stage is thus used to modify the permuted image pixels' values in order to confuse the relationship between the plain image and the encrypted image as we will see it in the results section. We have employed for the diffusion process, the combination of the iteration of chaotic Skew Tent Map (STM) and the XOR scheme, to mask the pixels. To enhance the security of the cryptosystem, the keystream used has been updated for each pixel and the computed encrypted pixel values depended on both to previously encrypted pixels and random keystream.

The mathematical modeling of the developed encryption algorithm is summarized below:

P is an image of length $M\times N$. Eq. (1) and Eq. (2) are respectively used to shuffle the positions of the image pixels and to change the pixel values (P).

$$P' = P(K) \tag{1}$$

Where P' is the permuted image and K the permutation key obtained by sorting chaotic sequences of the LM.

$$C_i = r \oplus \bmod\left(P_i' + C_{i-1} + a, 2^8\right) \tag{2}$$

Where C_i and C_{i-1} are the value of the currently and previously masking pixel respectively; C_0 can be set as a constant; P_i' is the permuted pixels; $\oplus$ is bitwise XOR operation; a is a positive integer and r is a diffusion key given according to the following formula:

$$r = \bmod\left(floor\left(y \times 2^{15}\right), 256\right) \tag{3}$$

Where y is the iteration of the STM, obtained with the parameters below for i=1:

$$\begin{cases} y_0 = \left(C_{i-1} + a\right) / \left(255 + a + b\right) \\ \alpha = \left(P_i' + a\right) / \left(M \times N + a + b\right) \end{cases} \tag{4}$$

The key r, is dynamically update for each pixel by choosing the positive parameters a and b to get α and y_0, the parameters of the STM in the interval (0,1).

The reconstruction of P cannot be made unless the distributions of r and K are determined. The inverse transform for deciphering is given by:

$$P_i' = \bmod\left(r \oplus C_i - C_{i-1} - a, 2^8\right) \tag{5}$$

$$P'(K) = P \tag{6}$$

For more understanding, the flowchart of encryption described in the Figure 2 is given.

The algorithm recalled and described above can thus be easily used in another platform as presented in the next section.

III. IMPLEMENTATION OF THE CRYPTOSYSTEM UNDER QT ENVIRONMENT USING THE C/C++

A graphical user interface (GUI), is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as display panels, action buttons, etc., as opposed to text-based interfaces, typed command labels or text navigation.

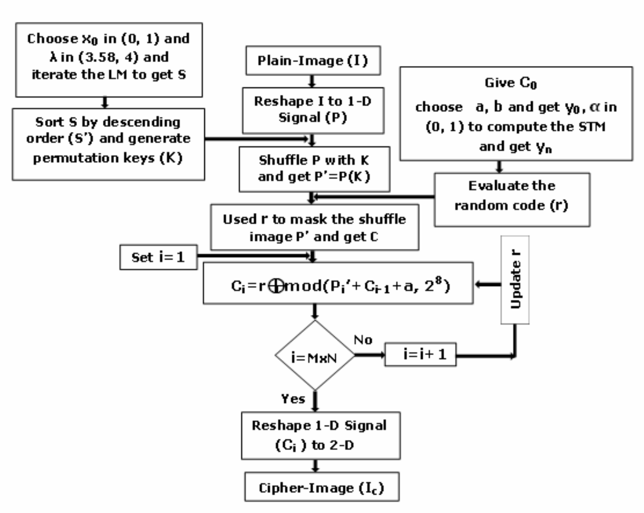

Figure 2. Flowchart of the encryption algorithm.

The C/C++ language does not have a native library for creating GUI. Therefore, one can find tools and libraries for creating GUI using the C/C++ programming language in [22]. For this research, a complete integrated development environment (IDE) Qt Creator was used with the Qt application framework to develop a GUI application. One of the major advantages of this framework is that it allows a team of developers to share a project across different development platforms with a common tool for development and debugging. It includes an integrated GUI layout, a code editor, a visual debugger and forms designer for designing and building GUI from widgets applications. This is useful and allows to transport the application across different computers in order to easily compare the execution time of the developed algorithm with others algorithms.

The pseudo-codes of the proposed image encryption algorithm are shown below:

```
PermutationEncryption(QImage *srcImg, QImage *dstImg, permutation_key_t *key)
 {
   if(key == NULL)
     return false;
   if(srcImg->width() != dstImg->width() ||
        srcImg->height() != dstImg->height())
     *dstImg = *srcImg;
   uint32_t  *outputBits = (uint32_t*)dstImg->bits();
   uint32_t  *sourceBits = (uint32_t*)srcImg->bits();
   int  pixelsNo = srcImg->width() * srcImg->height();  /* Number of pixels */
   /* Key must have the same length as number of pixels in the image */
   if(pixelsNo != key->len)
   {
     qDebug() << "Key is not good for this image";
     return false;
   }
   /* Permutation */
   for(int i = 0 ; i < key->len; i++)
     outputBits[i] = sourceBits[key->indexes[i]];
   return true;
 }
 DiffusionEncryption(QImage *dstImg, double a, double b)
 {
   /* Encrypt the first permuted pixel */
   for(int k = 1; k < T; k++)
   {
     if (x[k - 1] >= 0 && x[k - 1] <= q)
       x[k] = x[k - 1] / q;
     else if(x[k - 1] > q && x[k - 1] < 1)
       x[k] = (1 - x[k - 1]) / ( 1 - q );
   }
   chaoticState = x[T - 1];
   /* diffusion key */
   r = fmod(floor(chaoticState * pow(10.0, 20.0)), 256.0);
   r = 0;
   c[0] = r ^ ((p[0].blue + C0 + (int)a) % 256);
   /* Encrypt the other M*N-1 pixels */
   for(int i = 1; i < pixelsNo; i++)
   {
     x[0] = (c[i-1] + a) / (255.0 + a + b);
     q = (p[i].blue + a) / (pixelsNo + a + b);
     for(int k = 1; k < T; k++)
     {
       if (x[k - 1] >= 0 && x[k - 1] <= q)
         x[k] = x[k - 1] / q;
       else if(x[k - 1] > q && x[k - 1] < 1)
         x[k] = (1 - x[k - 1]) / ( 1 - q );
     }
     chaoticState = x[T - 1];
     /* diffusion key */
     r=fmod(floor(chaoticState* pow(10.0, 20.0)), 256.0);
     r = 0;
     c[i] = r ^ ((p[i].blue + c[i-1] + (int)a) % 256 );
   }
   for(int i= 0 ; i < pixelsNo; i++)
     p[i].blue = p[i].green = p[i].red = c[i];
   delete []x;
   delete []c;
 }
```

These pseudo codes include the permutation and diffusion strategies. The decryption procedure is simply the reverse of the encryption.

IV. NUMERICAL RESULTS AND DISCUSSION

We perform all the tests by loading Lena image of size 512×512 in the software. However, any types of images having any size can also be used. Encryption and decryption has been made using the following default parameters: $x_0 = 0.75$, $\lambda = 3.93695629843$, for the permutation and $a = 7$, $b = 4$, for the diffusion, with a desktop having the same properties than the computer used in [21]: 2.0 Ghz Pentium Dual-Core with 3GB RAM Running Windows XP.

Fig. 3 shows software after installation. As shown in this figure, the proposed image encryption software is very friendly because it provides to the user the possibility of choosing the type of architecture of his choice (permutation, diffusion, and permutation-diffusion) to encrypt and decrypt any type and any size of images. By choosing the permutation only, it can be observed in Table 1 and from

Figure 4 that, the developed permutation algorithm is reversible and just needs 1 *ms* for encryption and decryption. The correlation coefficients are also low and the pixels of the image are completely disturbed. This stage of our algorithm is then very fast compared to other permutation techniques [6,23].

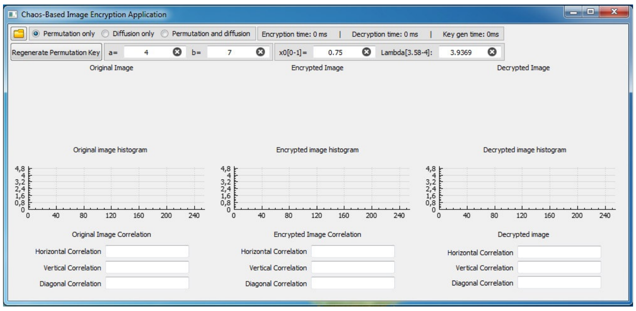

Figure 3. The software welcome interface.

Table I. The correlation coefficients and execution time at the permutation stage

Permutation only	Original image	Encrypted image	Execution time
Horizontal correlation	0.9718	0.0027	-
Vertical correlation	0.9849	0.0060	-
Diagonal correlation	0.9592	-0.0022	-
Encryption time	-	-	1 ms
Decryption time	-	-	1 ms

Unfortunately, as the permutation process scrambles the positions of the image pixels without disturbing their values, histogram of the permuted image thus remains the same as that of the original image as shown in Figure 4.

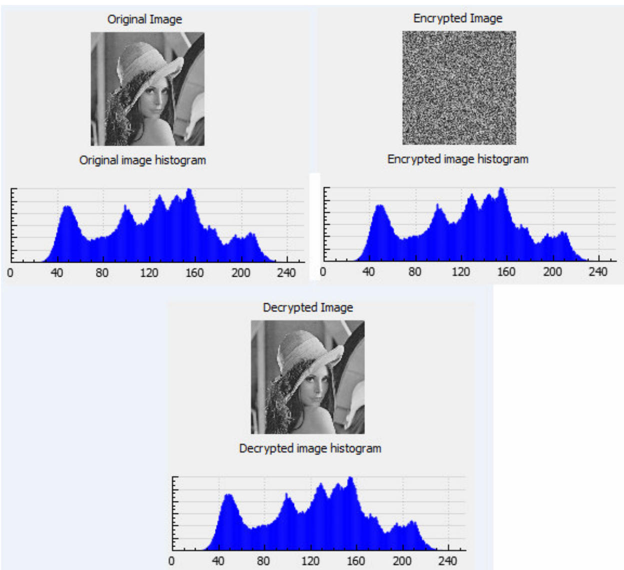

Figure 4. Chaos-based image permutation algorithm results.

By choosing the diffusion only, it can be observed in Table 2 that the developed diffusion algorithm needs 23 *ms* for encryption and decryption. The time of the diffusion process has now increased because at this stage the quantization of real chaotic values is now required to change the pixel values in order to confuse the relationship between the encrypted image and the original image. That is why the diffusion process still consumes the most time in a cryptosystem. The correlation coefficients for this stage are not good compared to those of the permuted image. However, the proposed diffusion technique is very fast compared to the ones published in [6,23].

Table 2. The correlation coefficients and execution time at the diffusion stage

Diffusion only	Encrypted image	Permuted image	Execution time
Horizontal correlation	-0.2904	0.0027	-
Vertical correlation	-0.0049	0.0060	-
Diagonal correlation	0.0103	-0.0022	-
Encryption time	-	1 ms	23 ms
Decryption time	-	1 ms	23 ms

The diffusion step is also highly secure as compared to the same references. As we can see in Figure 5, the histogram of the diffused image is fairly uniform. Therefore, the diffused image can resist against statistical attack.

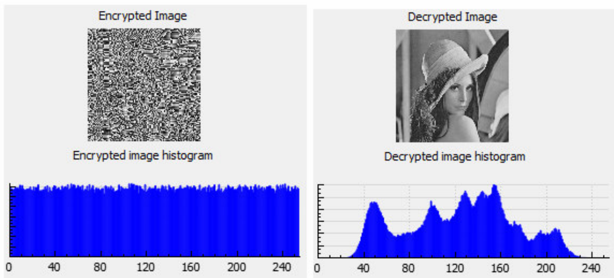

Figure 5. Chaos-based image diffusion algorithm results.

By choosing the permutation-diffusion structure, one can see in Table 3 and Table 4 that combining both schemes lead to good correlation coefficients and an acceptable encryption time. Indeed, it respects the Shannon's principle [24] which says that for more security, the permutation-diffusion architecture is required. Table 3 also shows that the developed encryption scheme requires only one overall round. While for conventional scheme DES and bit-level permutation scheme [25], at least two overall rounds are needed. One overall round is also achieved by the algorithm proposed in [6]. However, to get the correlation among the adjacent pixels completely unrecognizable in [6], the

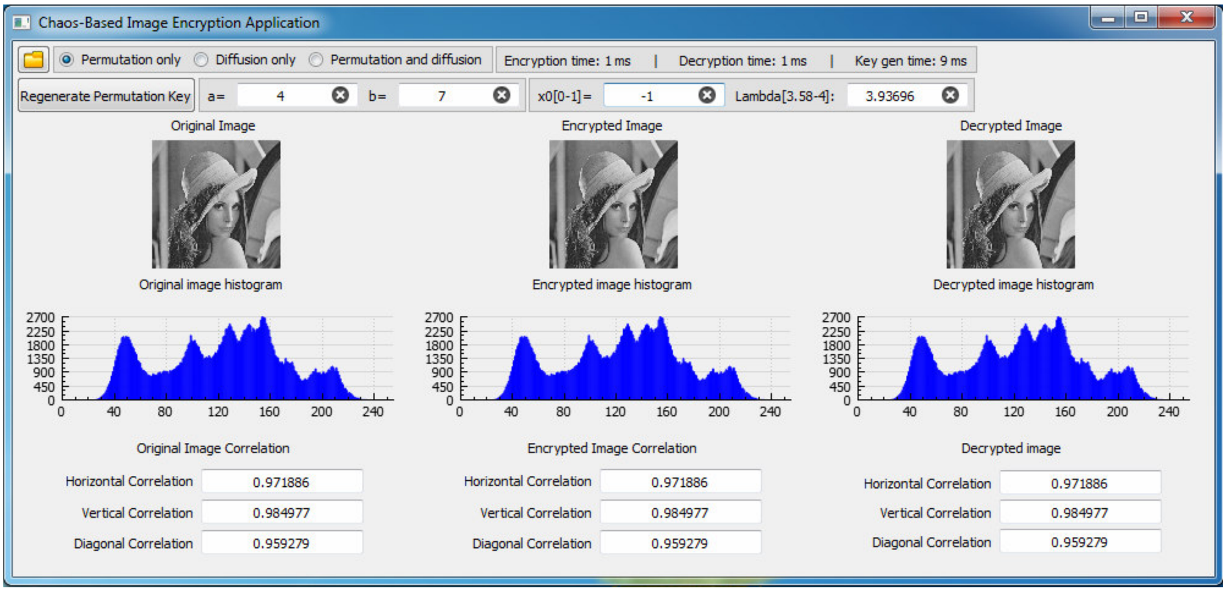

Figure 6. Chaos-based image permutation algorithm in the case of periodic logistic map.

permutation step needs five rounds of iterations while in our scheme, we only need one round. The running time proposed in our previous paper [21] was shorter than 100 *ms*. This is because the algorithm was implemented in Matlab environment. Matlab being an interpreted language is usually slower than compiled languages. With this proposed encryption software, implemented in C/C++ code, the average running speed is now shorter than 31 *ms* compared to other encryption schemes. It is now clear that we have developed a fast image encryption algorithm using chaos.

Table 3. Comparison of encryption time of the proposed with other encryption schemes

Algorithms	**Encryption rounds**	**Encryption times using**	
		C/C++ code (ms)	**Matlab (ms)**
Proposed	1	31	-
[6]	1	78	-
[21]	1	-	90
[25]	2	74	-
DES algorithm	16	170	-
Convention	3	141	-

Table 3. Comparison of the correlation coefficients and entropies of different ciphers

Permutation-diffusion	**Proposed**	**[6]**	**[21]**	**[25]**
Horizontal correlation	-0.0029	0.0088	-0.0002	0.0368
Vertical correlation	0.0020	-0.0087	0.0030	-0.0392
Diagonal correlation	-0.0032	-0.0060	0.0008	0.0068
Information entropy	7.9993	7.9902	7.9996	7.9880

It appeared from Table 4 that the proposed cryptosystem, even implemented using C/C++ code, has good statistical parameters which are close to those obtained when implemented using Matlab [21]. This table also shows its superiority compared to other cryptosystems.

For all above results, the success of the encrypted image is carried out through the chaotic maps which are mainly dependent on the initial conditions and parameters. Thus by filling the required parameters of chaotic systems in the software, the randomness property of the keystream is more suitable for image encryption. As shown in all presented figures, the required parameters are within the interval (0,1) for x_0, (3.58,4) for λ and a, b>0. Figure 6 shows the readable encrypted image when choosing x_0 = -1 for the permutation step for example. Thus, the proposed image encryption software is highly key sensitive.

IV. CONCLUSION

Our paper proposes an efficient chaos-based cryptosystem for real time applications. The cipher was implemented using C/C++ and a friendly graphical user interface was proposed. The paper has shown how the developed software was implemented under Qt environment and used for encrypting and decrypting an image using either the permutation algorithm, the diffusion algorithm, or the algorithm based on the permutation-diffusion scheme. The software application has been developed to work on any type of operating systems. It can be inferred from the result presented that the developed encryption software provides the user the opportunity of selecting an encryption stage of his choice to encrypt and decrypt the image. By analyzing each step, it appeared that the statistical parameters (correlation coefficients, histogram distribution) of the encrypted image obtained when using the permutation-diffusion scheme are better than those of the permuted and masked images. Therefore, using this software we have showed that a higher cryptographic security is achieved when permutation-diffusion structure is used, as Claude Shannon stated in [24]. Furthermore, the encryption time obtained in Matlab software is 35% improved. The computation time and the common analytical criterions of five existing encryption schemes was also determined and compared with the current approach and the results show that the proposed encryption algorithm is faster, highly secured and can be implemented using low end microprocessors. For future work we plan an implementation of this image encryption algorithm on the DSP boards.

ACKNOWLEDGEMENTS

This work has been funded with support from the AUF and from the European Commission through the Erasmus Mundus Action 2 project. J.D.D Nkapkop would like to thank the Vrije University of Brussels and the AUF for these financial supports.

REFERENCES

[1] D. Schonberg, S. C. Draper, C. Yeo, and K. Ramchandran, “Toward compression of encrypted images and video sequences”, in *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 749-762, December 2008, DOI:10.1109/TIFS.2008.2007244

[2] F-G. Jeng, W-L. Huang and T-H. Chen, “Cryptanalysis and improvement of two hyper-chaos-based image encryption schemes”, Image Communication, vol. 34, pp.45-51, May 2015. DOI:10.1016/J.IMAGE.2015.03.003

[3] L. Shubo, S. Jing and Z. Xu, “An Improved Image Encryption Algorithm based on Chaotic System”, Journal of Computers, vol. 4, no. 11, pp.1091-1011, November 2009. DOI:10.4304/jcp.4.11.1091-1100

[4] X. Liu, J. Zhang, Z. Jinhai and H.Xiqin, “Image Scrambling Algorithm Based on Chaos Theory and Sorting Transformation”, International Journal of Computer Science and Network Security, vol.8, no. 1, pp.64-68, January 2008.

[5] T. Morkel and J. H. P. Elof, “Encryption techniques: a timeline approach”, Information and Computer Security Architecture (ICSA) Research Group, University of Pretoria, 0002, Pretoria, South Africa, 2004.

[6] C. Fu, J-J. Chen and H. Zou, “A chaos-based digital image encryption scheme with an improved diffusion strategy”, Optics Express, vol. 20, no. 3, pp.2363-2378, January 2012. DOI:10.1364/OE.20.002363

[7] K. Gupta and S. Silakari, “New approach for fast color image encryption using chaotic map”, Journal of Information Security, vol. 2, no. 4, pp.139-150, October 2011. doi:10.4236/jis.2011.24014

[8] C. Li, S. Li and K-T. Lo, “Breaking a modified substitution-diffusion image cipher based on chaotic standard and logistic maps”, Communications in Nonlinear Science and Numerical Simulation, vol. 16, no. 2, pp.837-843, Feb. 2011. DOI:10.1016/J.CNSNS.2010.05.008

[9] G. Chen, Y. Mao and C. K. Chui, “A symmetric image encryption based on 3D chaotic cat map”, Chaos, Solitons and Fractals, vol. 21, no. 3, pp.749-76, July 2004. DOI:10.1016/J.CHAOS.2003.12.022

[10] L. Kocarev, G. Jakimoski, T. Stojanovski and U. Parlitz, “From chaotic maps to encryption schemes”, Circuits and Systems (ISCAS), 1998 IEEE Symposium on Circuits and systems, vol. 4, pp.514-517.

[11] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps”, Int. J. of Bifurcation and Chaos, vol. 8, no. 6, pp.12591284, December 1998.

[12] Y. Wang, X. Zhang, Z. Zheng and W. Qiu, “A colour image encryption algorithm using 4-pixel Feistel structure and multiple chaotic systems”, Nonlinear Dynamics, vol. 81, no. 1, pp.151-168, July 2015. Doi:10.1007/s11071-015-1979-3

[13] N. Bigdeli, Y. Farid and K. Afshar, “A novel image encryption/decryption scheme based on chaotic neural networks”, Engineering Applications of Artificial Intelligence, vol. 25, no. 4, pp. 753-765, June 2012. DOI:10.1016/J.ENGAPPAI.2012.01.007

[14] G. Ye and K-W. Wong, “An efficient chaotic image encryption algorithm based on a generalized Arnold map”, Nonlinear Dynamics, Springer, Vol. 69, no. 4, pp.2079-2087, September 2012. DOI:10.1007/s11071-012-0409-z

[15] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai, “On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption”, Communications in Nonlinear Science and Numerical Simulation, Vol. 17, no. 8, pp. 3303-3327, August 2012. DOI:10.1016/J.CNSNS.2011.12.015

[16] J-Y. Wang, G. Chen, “Design of a chaos-based digital image encryption algorithm in time domain”, in: *IEEE International Conference on Computational Intelligence and Communication Technology (CICT)*, pp.26-28, April 2015. DOI: 10.1109/CICT.2015.23

[17] A. A. Abd El-Latif, L. Li, T. Zhang and N. Wang, “Digital image encryption scheme based on multiple chaotic systems, Sensing and Imaging”, An international journal on continuing subsurface sensing technologies and applications, Springer, Vol. 56, no. 2, pp.67-88, June 2012. DOI:10.1007/s11220-012-0071-z

[18] S. Sam, P. Devaraj and R. S. Bhuvaneswaran, “A novel image cipher based on a mixed transformed logistic maps, Multimedia tools and applications”, An international Journal, Springer, Vol. 56, no. 2, pp. 315-330, January 2012. DOI:10.1007/s11042-010-0652-6

[19] M. Francois, T. Grosges, D. Barchiesi and R. Erra, “A new image encryption scheme based on a chaotic function”, Signal Processing: Image Communications, Vol. 27, No. 3, pp.249-259, March 2012.

[20] A. C. Dăscălescu, R. E. Boriga and A-V. Diaconu, “Study of a New Chaotic Dynamical System and Its Usage in a Novel Pseudorandom Bit Generator”, Mathematical Problems in Engineering, Vol. 2013, 2013.

[21] J. D. D. Nkapkop, J. Y. Effa, M. Borda and R. Terebes, “A Novel Fast and Secure Chaos-Based Algorithm for Image Encryption”, In I. Bica et al. (Eds.), *Innovative Security Solutions for Information Technology and Communications*, Lecture Note in Computer Science, Springer Switzerland, vol. 9522, pp.87-101, January 2016. DOI:10.1007/978-3-319-27179-8_7

[22] B. A. Myers, “User Interface Software Tools”, ACM Transactions on Computer-Human Interaction, vol. 2, pp.64-10, March 1995.

[23] P. Jagadeesh, P. Nagabhushan, Pradeep and R. Kumar, “A New Image Scrambling Scheme through Chaotic Permutation and Geometric Grid based Noise Induction”, International Journal of Computer Applications, vol. 78, no. 4, pp.38-45, September 2013. DOI:10.5120/13481-1181

[24] C. Shannon, “Communication theory of secrecy systems”, Bell System Technical Journal, vol. 28, pp.656-715, July 1949. DOI:10.1002/j.1538-7305.1949.tb00928.x

[25] C. Fu, B.-B. Lin, Y.-S. Miao, X. Liu and J.-J. Chen, “A novel chaos-based bit-level permutation scheme for digital image encryption”, Optics Communication, vol. 284, pp.5415-5423, November 2011, DOI:10.1016/J.OPTCOM.2011.08.013