Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption DRAFT VERSION 11-22-2004

Daniel Socek
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, Florida 33431–0991
Email: dsocek@brain.math.fau.edu

Spyros S. Magliveras
Department of Mathematical Sciences
Florida Atlantic University
Boca Raton, Florida 33431–0991
Email: spyros@fau.edu

Borko Furht
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, Florida 33431–0991
Email: borko@cse.fau.edu

***Abstract*— In the past few years, a number of image encryption algorithms based on chaotic maps have been proposed. A recently proposed Chaotic-Key Based Algorithm (CKBA) is based on a one-dimensional Logistic-map. However, it has been shown that the current CKBA model is unavoidably susceptible to chosen/known-plaintext attacks, and that the high security claims against ciphertext-only attack were overestimated by the authors. Some improvements of CKBA were suggested, but its susceptibility to chosen/known-plaintext attack remain. In addition, the chaotic Logistic-map yields unbalanced output. In this paper we enhance the CKBA algorithm three-fold: (1)– we change the 1-D chaotic Logistic-map to the Iterative Chaotic Map with Infinite Collapses (ICMIC) to improve the balance property, (2)– we increase the key space to 128 bits, and (3)– we randomly permute the bits of each encrypted pixel value so that chosen/known-plaintext attacks are no longer possible. The new cipher has much stronger security and its performance characteristics remain very good. A security analysis for the proposed system, is performed and presented.**

I. Introduction

The security of digital images has become increasingly more important in today's highly computerized and interconnected world. The media content must be protected in applications such as pay-per-view TV, confidential video conferencing, medical imaging, and in industrial or military imaging systems. With the rise of wireless portable devices, many seek to protect the private multimedia messages that are exchanged over the wireless or wired networks. Unfortunately, in many applications, conventional encryption algorithms (such as AES) are not suitable for image and video encryption [1–3]. In order to overcome this problem, many fast encryption algorithms specifically designed for digital images have been proposed [4–9]. Some video encryption algorithms are applicable to fixed images as well as videos [4], [5]. However, a number of these algorithms have been shown to be insecure [10–12].

The recently proposed image encryption methods based on chaotic maps generated considerable attention due to their fast performance and suitability for digital multimedia [2]. In [7], Yen and Guo proposed a chaotic key-based algorithm (CKBA) for image encryption. Subsequently, Li and Zheng [12] showed that the security claims for CKBA have been vastly overestimated. Not only is the complexity of a ciphertext-only attack against CKBA far lower than originally claimed, but chosen/known-plaintext attacks as described in [12] can be effectively applied. In [12], Li and Zheng suggest an improvement to CKBA by increasing the key length. As noted in [12] larger key size only improves the resistance of CKBA to a ciphertext-only attack, and not the algorithm's susceptibility to a chosen/known-plaintext attack. In this paper, we essentially propose a new cryptosystem that is based on the original CKBA.

Following the suggestions in [12] the new algorithm operates on an increased key size of 128-bits. In addition, a pseudo-random permutation generator (PRPG) based on the same chaotic map is introduced as an additional component in the encryption and decryption process to add much needed diffusion to the system. Finally, we replace the 1-D Logistic-map from the original CKBA with a 1-D Iterative Chaotic Map with Infinite Collapses (ICMIC) from [8], [13] in order to improve the chaotic properties. The new cryptosystem is significantly more secure than the original CKBA against both ciphertext-only attack and chosen/known-plaintext attack, with a very limited loss in speed.

The paper is organized as follows. Section II briefly introduces the original CKBA scheme from [7] and its security analysis from [12]. The framework of our proposed algorithm is described in Section III, with its security analysis and experimental results given in Sections IV and V respectively. The last section summarizes our conclusions.

II. On the Security of the Original CKBA

In essence, CKBA is a value transformation algorithm. The encryption of an $M \times N$ image I by CKBA is realized as follows. Without loss of generality, assume $8|MN$. Select two secret 8-bit keys k_1 and k_2, and a secret 16-bit initial condition $x(0)$ of a one-dimensional chaotic system. Iteratively run the chaotic system $MN/8 - 1$ times to produce a sequence of 16-bit numbers $\{x(i)\}_{i=0}^{MN/8-1}$, with $\{b(i)\}_{i=0}^{2MN-1}$ being its binary representation. If $I(x, y)$ is an 8-bit pixel value in the plaintext image I, with $0 \leq x < M$ and $0 \leq y < N$, the corresponding ciphertext pixel $I'(x, y)$ is defined by the following rule:

$$I'(x,y) = \begin{cases} I(x,y) \oplus k_1, & \text{if } b'(x,y) = 3; \\ I(x,y) \oplus \overline{k_1}, & \text{if } b'(x,y) = 2; \\ I(x,y) \oplus k_2, & \text{if } b'(x,y) = 1; \\ I(x,y) \oplus \overline{k_2}, & \text{if } b'(x,y) = 0, \end{cases} \quad (1)$$

where $b'(x,y) = 2b(l) + b(l+1)$ and $l = 2(xN + y)$.

As a security requirement, although the keys k_1 and k_2 are chosen at random, it is required that the Hamming distance between them be 4. That is, k_1 and k_2 should differ at exactly $|k_i|/2$ positions, $i \in \{1, 2\}$.

Finally, a quick observation shows that the decryption process is the identical mapping since XOR is an involution.

As Li and Zheng showed [12], the security of the aforementioned algorithm was highly overestimated in [7], where the authors claimed that the key search space for the ciphertext-only attack is 2^{2MN}. The chosen/known-plaintext attacks were not even considered in [7]. As Li and Zheng showed [12], the actual key search space for the ciphertext-only attack is

$$2^{|x(0)|+|k_i|} \times \binom{|k_i|}{\frac{|k_i|}{2}},$$

where $i \in \{1, 2\}$, which for the original set of parameters enumerates to only $2^{24} \times 70$.

Furthermore, the original scheme is subject to well-defined chosen/known-plaintext attacks [12]. That is, CKBA can be completely broken if only one plaintext image and its corresponding ciphertext image are known. Suppose we have the images I and its CKBA encryption I' obtained by using secret key $(k_1, k_2, x(0))$. By virtue of the algorithm's definition, I' can be obtained from I by XOR-ing it with a particular image mask I_m. Consequently, the image mask I_m can be obtained simply by XOR-ing images I and I'. This mask can then be used to completely decrypt all other images of same or smaller size for which the same keys k_1, k_2, and $x(0)$ were used. Fig. 1 demonstrates chosen/known-plaintext attack on CKBA where the same key is used to encrypt both 128×128 "Lena" and 128×128 "Barb". The attacker can easily recover "Barb" from unknown ciphertext in Fig. 1(d).

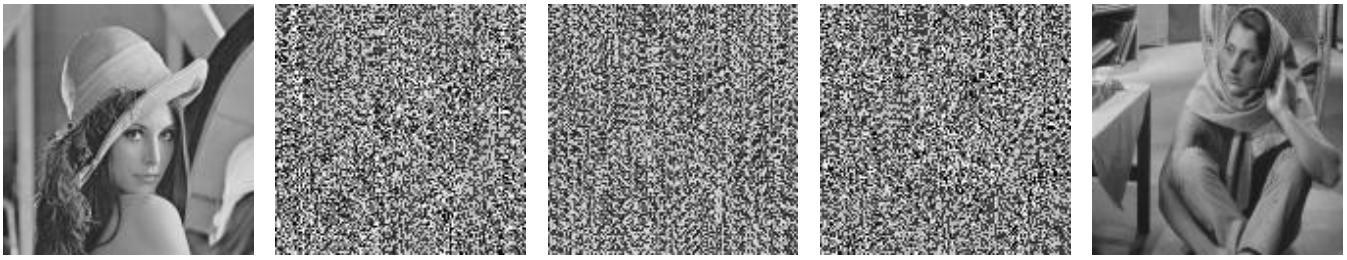
In addition, Li and Zheng constructed a $O(MN)$ brute force algorithm for the CKBA scheme that could be used to completely recover the keys k_1, k_2, and $x(0)$, provided that I_m is known, which makes it possible to recover all images encrypted with the same key, regardless of the image size [12].

A cryptosystem that is susceptible to chosen/known-ciphertext attacks is not recommended in general. Having to change the key from image to image is a big drawback for many applications. Additionally, such cipher cannot maintain security when applied to videos (sequences of images). Therefore a cipher that can resist these kinds of attacks is much more preferable.

Li and Zheng [12] proposed an improvement to CKBA based on increasing the key sizes, but as they noted, this only improves the resistance to a ciphertext-only attack, and does nothing to prevent the chosen/known-ciphertext attacks. Once a mask image is obtained, everybody can decrypt all images of same or smaller size that were encrypted with that same key by a simple XOR operation. Images of larger sizes could be decrypted partially, or fully when applying the brute force key recovery method with $O(MN)$ complexity as described in [12]. The main drawback of value substitution approaches such as CKBA is their susceptibility to chosen/known-ciphertext attacks via the substitution mask. Therefore, performing a substitution only, i.e, using only an S-box alone, is not recommended from a cryptanalytic point of view. However, if we change this simple substitution by a substitution followed by a variable pseudo-random permutation of the bits within each pixel value, we would have created an SP-network which is much harder to cryptanalyze [14]. Note that performing only a permutation transformation to pixel values is not sufficient, since the pixel values whose binary representation consists of all zeros or all ones will not be changed at all. For example, the encryption of an X-ray medical image would reveal too much visual information since such an image contains large blocks of black and white pixels whose binary representation consists of all zeros and all ones, respectively. Fig. 2 clearly shows this undesired effect.

III. The Enhanced CKBA (ECKBA) for Image Encryption

Let I_{MNb} be an $M \times N$ image with b-byte(s) pixel values, where a pixel value is denoted by $I_{MNb}(i)$, $0 \leq i < M \times N \times$



(a) "Lena" (b) Encrypted "Lena" (c) XOR mask (d) Unknown ciphertext (e) Cracked ciphertext "Barb"

Fig. 1. Chosen/known-ciphertext attack on CKBA: the attacker calculates (c) by XOR-ing (a) and (b), and then (e) by XOR-ing (c) and (d).

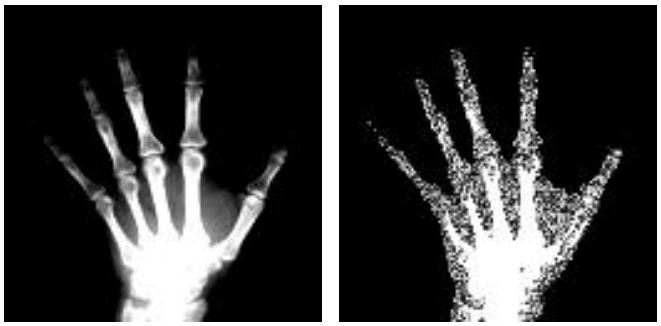


Fig. 2. Weak X-ray image encryption as a result of pixel value permutation without the substitution step (S-box).

b, scanned left-to-right and top-to-bottom. Let $\mathcal{C}_\mu$ be a one-dimensional chaotic map with a real coefficient μ obtained by normalizing a 32-bit integer μ_{Int32} to a chaotic interval. Let $x(0)$ be the initial condition for $\mathcal{C}_\mu$ obtained by normalizing a 32-bit integer $x(0)_{Int32}$ to a point range defined for $\mathcal{C}_\mu$. For a given n-bit segment x, let $l(x)$ denote its low significant half and $h(x)$ its high significant half. Finally, let π_i, $0 \leq i < 8!$ be a permutation of degree 8 whose index in the full symmetric group S_8 sorted in lexicographical cartesian order is i. Without loss of generality assume $4|MNb$. The proposed encryption scheme is realized by the following algorithm:

Data: Image I_{MNb} and 128-bit key k.
Result: Image I'_{MNb}.

1 **begin**
2 $x(0)_{Int32} \longleftarrow l(l(k))$; $y(0)_{Int32} \longleftarrow h(l(k))$
3 $\alpha_{Int32} \longleftarrow l(h(k))$; $\beta_{Int32} \longleftarrow h(h(k))$
4 **for** $1 \leq i < \frac{MNb}{4}$ **do**
5 $x(i) \longleftarrow \mathcal{C}_\alpha(x(i-1))$
6 **end**
7 **for** $1 \leq i < \frac{MNb}{2}$ **do**
8 $y(i) \longleftarrow \mathcal{C}_\beta(y(i-1))$
9 **end**
10 **for** $0 \leq i < \frac{MNb}{4}$ **do**
11 $I'_{MNb}(4i) \longleftarrow I_{MNb}(4i) \oplus l(l(x(i)_{Int32}))$
12 $I'_{MNb}(4i+1) \longleftarrow I_{MNb}(4i+1) \oplus h(l(x(i)_{Int32}))$
13 $I'_{MNb}(4i+2) \longleftarrow I_{MNb}(4i+2) \oplus l(h(x(i)_{Int32}))$
14 $I'_{MNb}(4i+3) \longleftarrow I_{MNb}(4i+3) \oplus h(h(x(i)_{Int32}))$
15 $\pi_{l(y(2i)_{Int32}) \bmod 8!}(I'_{MNb}(4i))$
16 $\pi_{h(y(2i)_{Int32}) \bmod 8!}(I'_{MNb}(4i+1))$
17 $\pi_{l(y(2i+1)_{Int32}) \bmod 8!}(I'_{MNb}(4i+2))$
18 $\pi_{h(y(2i+1)_{Int32}) \bmod 8!}(I'_{MNb}(4i+3))$
19 **end**
20 **end**

Algorithm 1: ECKBA Encryption

Algorithm 1 transforms an image I using an SP-network generated by a one-dimensional chaotic map and a 128-bit secret key. The lines 4-9 are used to generate two pseudo-random sequences, $\{x(i)\}_{i=0}^{MNb/4}$ and $\{y(j)\}_{j=0}^{MNb/2}$, both with 32-bit entries. The sequence $\{x(i)\}_{i=0}^{MNb/4}$ is used as a value substitution mask in lines 11-14, while the sequence $\{y(j)\}_{i=0}^{MNb/2}$ is used as a PRPG in lines 15-18.

The decryption algorithm is practically identical to *Algorithm 1*, except for the following.

- Replace the permutation π_k by its inverse permutation π_k^{-1} in lines 15-18.
- Replace I'_{MNb} by I_{MNb} in lines 15-18.
- Interchange the block of lines 11-14 and the block of lines 15-18.

In *Algorithm 1*, we need to obtain a permutation for a given index in the lexicographically sorted permutation group S_8. The fastest way to achieve this is by using a table-lookup approach. This approach is fast, but the memory requirements are high. In applications where this is not acceptable, such as small wireless devices with low memory capacity, a computational approach is needed. In the Appendix section we present an efficient algorithm for computing the permutation of a given index.

Both CKBA and ECKBA use a one-dimensional chaotic map $\mathcal{C}$ with a specified initial condition x_0. The original CKBA uses the following map, known as the Logistic-map:

$$x_n = \mathcal{C}_\mu(x_{n-1}) = \mu x_{n-1}(1 - x_{n-1}),$$

where $\mu \in (0, 4]$ and $x_i \in (0, 1)$. The Logistic-map has been well-studied in the past, and it had been shown that the positive constant μ should be greater than the accumulation point 3.569945672 in order to maintain the highly chaotic state. This is a desirable property in cryptographic applications, and the implementations of ECKBA and CKBA should limit μ to the real interval $(3.569945672, 4.0]$. Regardless of that, due to the poor balance property of a Logistic-map (see Section IV), we recommend ECKBA (and CKBA) implementations to use a better balanced ICMIC map from [13]:

$$x_n = \mathcal{C}_\mu(x_{n-1}) = \sin(\frac{\mu}{x_{n-1}}),$$

where the positive real constant μ should be in a chaotic interval and $x_i \in [-1, 0) \cup (0, 1]$. He et al. [13] showed that there are infinitely many chaotic intervals for μ in ICMIC. For example, if μ is in the real interval $(1.8627, 2.5429)$, ICMIC behaves chaotically [13].

All of the computer calculations involving enumerations of a chaotic map $\mathcal{C}_\mu$ are done in some fixed precision. Using the finite precision to create a recursive sequence defined in a real domain may cause losing some of the important pseudo-random properties, such as the high periodicity. Results from [15] indicate that when a chaotic system is created using smaller finite computing precision, the cycle length of the chaotic orbits, or the periodicity, becomes smaller. The original CKBA uses a 16-bit precision. For ECKBA, the floating-point output is approximated and mapped into a 32-bit words. The periodicity of the ECKBA is further investigated in Section IV.

IV. Security Analysis

In this section, we analyze ECKBA from a security point of view. Our ECKBA scheme is conceptually based on the CKBA scheme from [7], however, we claim that the security of our scheme is much higher than that of the original CKBA.

A. The Key Space

In ECKBA, the key space is vastly increased. Namely, the *Algorithm 1* works with a 128-bit secret key, as opposed to the original CKBA which works with a limited 32-bit secret key. By today's standards, a key of at least 64-bits, and preferably of 128-bits or 256-bits is required for symmetric-key cryptosystems [14]. The white-box analysis of CKBA from [12] reveals that its actual key-space is $log_2(2^{24} \times 70)$-bits, which enumerates to about 30-bits. Since the ECKBA scheme does not have any limitations on the secret key, the key space is 128-bits. Therefore, a ciphertext-only attack based on exhaustive key search (brute-force attack) is not feasible.

B. Logistic-Map vs. ICMIC

Since the pseudo-random output of a one-dimensional chaotic map is used for both confusion and diffusion, we need a map with better chaotic properties. In the ECKBA framework, it is particularly desired that the chosen chaotic map satisfy the balance property (or uniformity). That is, the number of zeros and ones in both of the output sequences $\{x\}$ and $\{y\}$ must be roughly equal for large sample sizes. In addition, the map must also have sufficiently large periodicity.

One standard measure of the chaotic behavior of a given map is the *Lyapunov exponent* λ. When λ is 0 or negative, the system is stable or asymptotically stable, respectively. A positive value of λ indicates that the system is unstable and chaotic. The higher Lyapunov exponent is, the more unstable and chaotic the system is. From [13], we see that the average value of Lyapunov exponents for the Logistic-map is 0.6941, while the average value of of Lyapunov exponents for the ICMIC with $\mu = 2$ is 1.6695, thus indicating a much better chaotic behavior for ICMIC.

In addition, our experiments confirm that ICMIC have much better balance property than the Logistic-map. We generated a set of 100 sequences for each chaotic map with 16-bit and 32-bit precision, denoted by S_{16}^L, S_{32}^I, S_{16}^L, and S_{32}^I, where the superscript represents the chaotic map (I stands for ICMIC and L for Logistic-map) while the subscript n indicates an n-bit precision. Each sequence in S_n^m contained 100000 n-bit words, totaling in $100000 \times n$ bits, and was computed using randomly selected parameters μ and $x(0)$. The constant μ was normalized to the chaotic range $[3.6, 4.0]$ for the Logistic-map and $[1.9, 2.5]$ for ICMIC. By the balance property of random sequences, it is expected that the number of ones and zeros for such large sequences be roughly the same. As Fig. 3 and Fig. 4 show, the sequences based on the Logistic-map had a visibly larger percentage of ones. On the other hand, ICMIC was well-balanced since the number of ones and zeros were about the same (Fig. 5 and Fig. 6). Table I shows the average percentage of zeros and ones for each set.

	Avg. Percentage of Zeros	Avg. Percentage of Ones
S_{16}^L	0.47618128	0.52381873
S_{32}^L	0.48781678	0.51218322
S_{16}^I	0.49987147	0.50012857
S_{32}^I	0.49999758	0.50000248

TABLE I

AVERAGE PERCENTAGES OF ZEROS AND ONES FOR EACH SET.

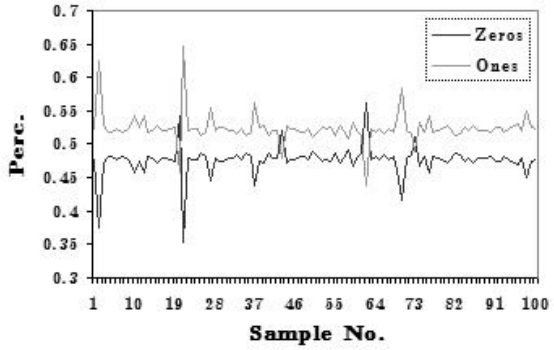


Fig. 3. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{16}^L.

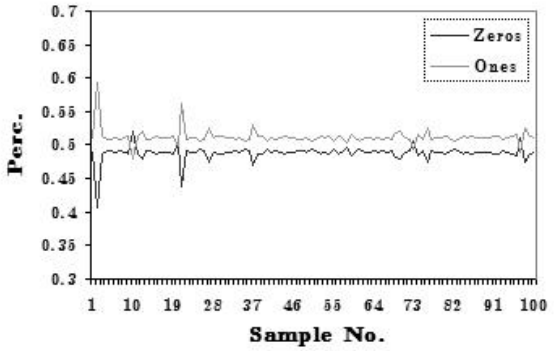


Fig. 4. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{32}^L.

C. SP-Network of ECKBA

In Section II we saw that CKBA is extremely vulnerable to the chosen/known-ciphertext attack. By introducing the variable permutation component in the encryption process, we eliminate the types of attacks discussed in [12].

Suppose we have three $M \times N$ images I, I', and J'. Furthermore, suppose we know that I' is an encryption of I using key k, and that a ciphertext image J' was encrypted using the same algorithm and the same key k. For ECKBA, since each pixel is uniquely permuted after the XOR operation, the mask image I_m obtained by XOR-ing the plaintext image I with its corresponding ciphertext image I' cannot be directly applied to recover the unknown ciphertext image J' encrypted with the same key. Fig. 7 demonstrates an unsuccessful chosen/known-plaintext attack on ECKBA, which is the analog to the attack in Fig. 1. In the case of CKBA the attack was possible because of the following property.

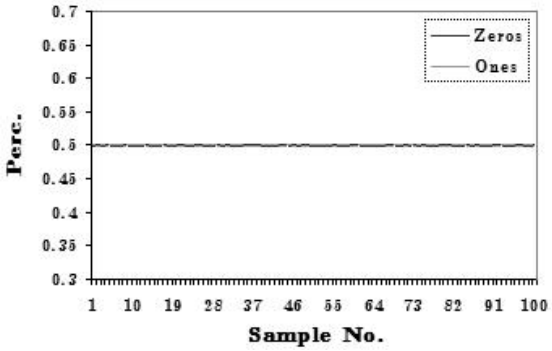


Fig. 5. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{16}^I.

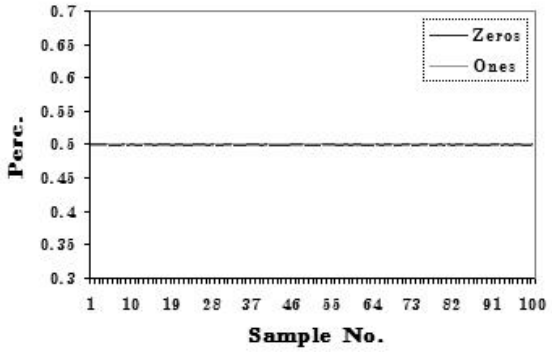


Fig. 6. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{32}^I.

Let $p = I(i,j)$ be a pixel value of I at coordinates (i,j). Then $p' = I'(i,j)$ can be expressed as $p' = p \oplus x$, for some 8-bit binary string x. By the framework of CKBA, $q' = J'(i,j) = q \oplus x$, where $q = J(i,j)$. If q is unknown and p, p' and q' are known, then q is easily calculated by

$$p \oplus p' \oplus q' = p \oplus (p \oplus x) \oplus (q \oplus x) =$$

$$= p \oplus p \oplus x \oplus x \oplus q = q.$$

In the case of ECKBA, the equation is not solvable. Namely, $p' = I'(i,j)$ can be expressed as $p' = \pi(p \oplus x)$, for some 8-bit binary string x and a permutation $\pi \in S_8$. Similarly, we know that $q' = J'(i,j) = \pi(q \oplus x)$. If π^{-1} is known, we can easily obtain the solvable CKBA equation from above, but if the inverse of π is not known, we are stuck. It is not even possible exhaustively search for π^{-1} among the 8! permutations, since we do not know x, and consequently, the value of $p \oplus x$.

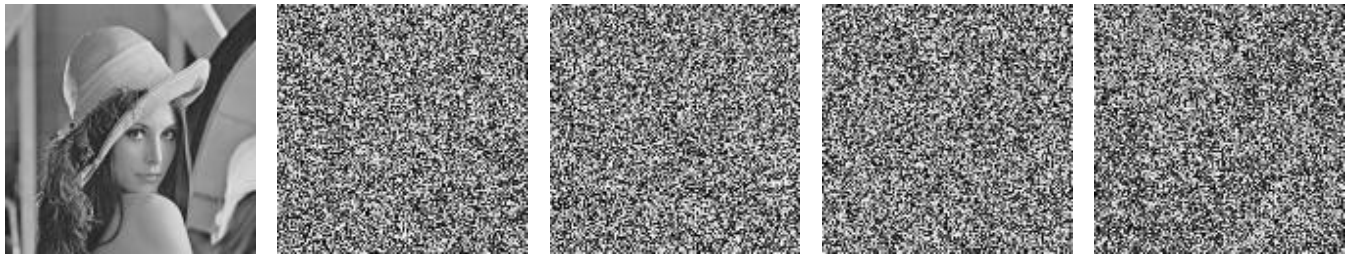
An important security consideration for both CKBA and ECKBA is that if the images I and J are very similar, which is the case for consecutive video frames, then I' and J' will be just as similar since $I(i,j) = J(i,j)$ implies that $I'(i,j) = J'(i,j)$ for both schemes if the same key was used to encrypt I and J. For such applications, the encryption models of CKBA and ECKBA should be implemented in the chaining (CBC) mode. Then, $I(i,j) = J(i,j)$ does not imply that $I'(i,j) = J'(i,j)$ as long as there exist some $0 \le x < i$ and $0 \le y < j$ for which $I(x,y) \neq J(x,y)$.

D. On the Periodicity in ECKBA

The results from [15] show that finite precision roundoff errors affect the periodicity of a chaotic map. Assuming that some finite precision is used, say a 32-bit precision, let the periodicity of a chaotic map $\mathcal{C}_\mu$ with the initial condition x be P_μ^x. The ECKBA scheme uses two distinct chaotic maps, $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$, with corresponding periodicities $P_\alpha^{x(0)}$ and $P_\beta^{y(0)}$. By the definition of the encryption transformation, for the key to start repeating both periodicities must be synchronized. However, this means that the effective periodicity of ECKBA is $\mathrm{lcm}(P_\alpha^{x(0)}, P_\beta^{y(0)})$, which normally is of much larger magnitude than either of the two periodicities.

V. Experiments

In Section IV we showed that ECKBA is much more secure than the original CKBA from [7]. In this section, we run experiments to evaluate the performance of ECKBA in comparison to CKBA. Since ECKBA introduces an additional step, and it uses higher precision and more complex map than the CKBA, its is expected that the running time of encryption/decryption algorithm goes up. We have implemented both ECKBA and CKBA methods. ECKBA was implemented both using table-lookup approach and the computational approach



(a) "Lena" (b) Encrypted "Lena" (c) XOR mask (d) Unknown ciphertext (the original image was "Barb") (e) Failed attempt to crack the ciphertext of "Barb"

Fig. 7. Unsuccessful chosen/known-ciphertext attack on ECKBA: the attacker calculates (c) by XOR-ing (a) and (b), and then (e) by XOR-ing (c) and (d).

Image	Size [$M \times N \times b$ bytes]	CKBA Encryption Time
camera	$256 \times 256 \times 1$	0.002 sec
barb	$512 \times 512 \times 1$	0.01 sec
lena	$1024 \times 1024 \times 1$	0.03 sec
tulips	$768 \times 512 \times 3$	0.04 sec
frymire	$1118 \times 1104 \times 3$	0.11 sec

TABLE II
PERFORMANCE OF CKBA ENCRYPTION

Image	Size [$M \times N \times b$ bytes]	ECKBA Encryption Time
camera	$256 \times 256 \times 1$	0.01 sec
barb	$512 \times 512 \times 1$	0.03 sec
lena	$1024 \times 1024 \times 1$	0.14 sec
tulips	$768 \times 512 \times 3$	0.16 sec
frymire	$1118 \times 1104 \times 3$	0.50 sec

TABLE III
PERFORMANCE OF ECKBA ENCRYPTION USING TABLE-LOOKUP APPROACH

Image	Size [$M \times N \times b$ bytes]	ECKBA Encryption Time
camera	$256 \times 256 \times 1$	0.07 sec
barb	$512 \times 512 \times 1$	0.27 sec
lena	$1024 \times 1024 \times 1$	1.08 sec
tulips	$768 \times 512 \times 3$	1.22 sec
frymire	$1118 \times 1104 \times 3$	3.82 sec

TABLE IV
PERFORMANCE OF ECKBA ENCRYPTION USING COMPUTATIONAL APPROACH

using *Algorithm 2* from the Appendix section. Both modes of ECKBA used a 32-bit precision and an ICMIC chaotic map. CKBA was implemented by using Logistic-map, and the precision was kept to be 16 bits in order to keep the original framework described in [7]. We measured the performances for images with different sizes, since the content of the image hardly affects the execution time of both CKBA and ECKBA. The performance experiments were run on a 1.3GHz Intel(R) Pentium(R) M processor, and the results are summarized in Tables II, III and IV. The experimental results suggest that the running time of ECKBA implemented using a table-lookup approach is comparable to that of CKBA. The computational overhead is minimal, but the security level is highly improved, as shown in Section IV. For larger images, the ECKBA implemented using computational approach performs slow. However, such approach is likely to be used with small devices of limited memory capacity that usually deal with smaller images, due to the obvious restrictions such as the small storage size, the small display size, etc. For smaller images, ECKBA implemented using computational approach is of an acceptable performance.

VI. CONCLUSION

We proposed a novel image encryption algorithm, called ECKBA, that is based on the previously proposed method by Yen and Guo [7]. Our approach is similar to the approach by Yen and Guo, but with much higher security level. The enhanced security comes from the following changes to the original algorithm: higher key space, more chaotic one-dimensional map, and the use of an SP-network. As the experiments suggest, the performance is only marginally affected. There are other possible improvements that could be considered. Namely, encryption process could also include a transformation on the block of pixels, or preferably on the entire image. CKBA and ECKBA only transform the pixel values. Improved confusion and diffusion is expected when applying transformations on the positions of pixels as well. It is preferred that the permutations that reorder the pixel positions be of high degrees. Currently, we are investigating pseudo-random permutation generators based on chaotic maps and group bases of permutation groups, that are capable of efficiently producing permutations of high degrees.

APPENDIX

In the Appendix section we present an algorithm for computing a permutation from the full symmetric group S_n that corresponds to a given index in its lexicographically sorted cartesian form. A permutation is usually given in its *standard form*. Suppose $\pi \in S_8$, and in its standard form π is defined as follows:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 3 & 8 & 5 & 7 & 1 & 4 & 2 \end{pmatrix}.$$

We can also represent π in its *cartesian form*, also called the *brackets form*. When defined in the cartesian form, π looks as follows:

$$\pi = [6\ 3\ 8\ 5\ 7\ 1\ 4\ 2].$$

Without the square brackets, this represents a more natural way of storing permutations in a computer. The full symmetric group S_8 can be represented as an ordered set $\{\pi_i\}_{i=0}^{8!-1}$ of permutations defined in their cartesian form and sorted in a lexicographic order. In this set, each permutation have a unique index x between 0 and $8!-1$, and π_x denotes such permutation.

If π is a permutation of degree n and $0 \leq i < n$, let $\pi[i]$ denote a value at $(i+1)^{th}$ position of π. For a given index x the following algorithm could be used to computationally retrieve a permutation π_x in the ordered set $\{\pi_i\}_{i=0}^{n!-1}$.

Algorithm 2 can be utilized for ECKBA encryption and decryption in applications where implementing a table-lookup approach is too memory expensive. Utilizing such computational approach decreases the amount of needed memory, however, it affects the performance (see Section V for experimental results).

ACKNOWLEDGMENT

Appropriate acknowledgements will go here prior to submission.

Data: Index x satisfying $0 \leq x < n!$, and the permutation degree n.
Result: Permutation π_x.

```
begin
    m ⟵ n − 1
    τ ⟵ the identity of S_n
    for 0 ≤ i < n do
        π_x[i] ⟵ τ[⌊x/m!⌋]
        for ⌊x/m!⌋ ≤ j ≤ m do
            τ[j] ⟵ τ[j + 1]
        end
        x ⟵ x mod m!
        m ⟵ m − 1
    end
end
```

Algorithm 2: Computing the permutation for given index.

REFERENCES

[1] B. Furht and D. Socek. Multimedia security: Encryption techniques. In *IEC Comprehensive Report on Information Security, International Engineering Consortium*, Chicago, 2003.

[2] Guanrong Chen, Shujun Li and Xuan Zheng. *Multimedia Security Handbook* edited by B. Furht and D. Kirovski, volume 4 of *Internet and Communications Series*, chapter "Chaos-Based Encryption for Digital Images and Videos". CRC Press, December 2004.

[3] Borko Furht, Daniel Socek, and Ahmet M. Eskicioglu. *Multimedia Security Handbook* edited by B. Furht and D. Kirovski, volume 4 of *Internet and Communications Series*, chapter "Fundamentals of Multimedia Encryption Techniques". CRC Press, December 2004.

[4] L. Tang. Methods for encrypting and decrypting MPEG video data efficiently. In *Proceedings of the 4th ACM International Multimedia Conference*, pages 219–230, 1996.

[5] C. Shi, S. Wang, and B. Bhargava. MPEG video encryption in real-time using secret key cryptography. In *Proceedings of the 4th ACM International Multimedia Conference*, pages 219–230, 1999.

[6] H. Cheng and X. Li. Partial encryption of compressed images and video. In *IEEE Transactions on Signal Processing*, volume 48, pages 2439–2451, 2000.

[7] Jui-Cheng Yen and Jiun-In Guo. A new chaotic key-based design for image encryption and decryption. In *Proceedings of 2000 IEEE International Conference on Circuits and Systems (ISACS 2000)*, volume 4, pages 49–52, 2000.

[8] F. Belkhouche and U. Qidwai. Binary image coding using 1d chaotic maps. In *IEEE Region 5 Technical Conference*, pages 39–42, New Orleans, April 2003.

[9] M. Van Droogenbroeck. Partial encryption of images for real-time applications. In *Fourth IEEE Signal Processing Symposium*, pages 11–15, Hilvarenbeek, The Netherlands, 2004.

[10] K. Nahrstedt, L. Qiao and I. Tam. Is MPEG encryption by using random list instead of zigzag order secure? In *IEEE International Symposium on Consumer Electronics*, Singapore, 1997.

[11] T. Seidel, D. Socek and M. Sramka. Cryptanalysis of video encryption algorithms. In *Proceedings of The 3rd Central European Conference on Cryptology (TATRACRYPT '04)*, Bratislava, Slovak Republic, 2003.

[12] Shujun Li and Xuan Zheng. Cryptanalysis of a chaotic image encryption method. In *Proceedings of 2002 IEEE International Symposium on Circuits and Systems (ISCAS 2002)*, volume 2, pages 708–711, 2002.

[13] L.-G. Jiang H.-W. Zhu D. He, C. He and G.-R. Hu. Chaotic characteristics of a one-dimensional iterative map with infinite collapses. In *IEEE Transactions on Circuits and SystemsI: Fundamental Theory and Applications*, volume 48, pages 900–906, 2001.

[14] Douglas R. Stinson. *Cryptography: Theory and Practice*. CRC Press, second edition, 2002.

[15] W. Li X. Mou S. Li, Q. Li and Y. Cai. Statistical properties of digital piecewise linear chaotic maps and their roles in cryptography and pseudo-random coding. In *Cryptography and Coding - 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 205–221. Springer-Verlag, Berlin, 2001.