Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption

Daniel Socek*, Shujun Li†, Spyros S. Magliveras‡ and Borko Furht§
*§Department of Computer Science and Engineering
‡Department of Mathematical Sciences
*‡Center for Cryptology and Information Security
Florida Atlantic University, Boca Raton, Florida 33431–0991
Email: *dsocek@brain.math.fau.edu, §borko@cse.fau.edu, ‡spyros@fau.edu
†Department of Electronic and Information Engineering
Hong Kong Polytechnic University, Hung Hom
Kowloon, Hong Kong SAR, China

***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. 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 a piecewise linear chaotic map (PWLCM) to improve the balance property, (2)– we increase the key size to 128 bits, and (3)– we add two more cryptographic primitives and extend the scheme to operate on multiple rounds so that the 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 users 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 [4–7]. Some video encryption algorithms are applicable to still images as well as videos [4], [7]. However, a number of these algorithms have been shown to be insecure [8–10].

The image encryption methods based on chaotic maps attract considerable attention recently due to their potential for digital multimedia encryption [2]. In [6], Yen and Guo proposed a chaotic key-based algorithm (CKBA) for image encryption. Subsequently, Li and Zheng [10] 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 described in [10] can be effectively applied. As discussed in [10], the security against ciphertext-only attacks can be improved by increasing the key length. However, such a simple countermeasure cannot improve the security against known/chosen-plaintext attacks. In this paper, we essentially propose a new cryptosystem that is based on the ideas from the original CKBA.

Following the suggestions in [10] the new algorithm operates on an increased key size of 128-bits. We also replace the 1-D Logistic map in the original CKBA with a 1-D piecewise linear chaotic map (PWLCM) used in [14], [15], in order to improve the statistical properties of the secret bits generated by the chaotic map. Next, a pseudo-random permutation generator (PRPG) based on the new chaotic map is introduced as an additional component in the encryption and decryption processes to create a permutation box (P-box), and thus add a much needed diffusion to the system. We also introduce the addition modulo the pixel value space to build a more complex substitution box (S-box). Finally, multiple rounds are employed in the encryption and decryption processes to build a stronger security wall. The new cryptosystem is significantly more secure than the original CKBA against both ciphertext-only attack and chosen/known-plaintext attacks, with an acceptable loss in speed.

The paper is organized as follows. Section II briefly introduces the original CKBA scheme and the cryptanalysis results reported in [10]. 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 concludes this paper.

II. On the Security of the Original CKBA

In essence, CKBA is a value transformation cipher. 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 $l = 2(x+yM)$.

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, where $i \in \{1,2\}$ and $|k_i|$ denotes the bitsize of k_i.

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

As Li and Zheng showed [10], the security of the aforementioned algorithm was highly overestimated in [6], 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 [6]. In fact, the actual key space for the ciphertext-only attack is

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

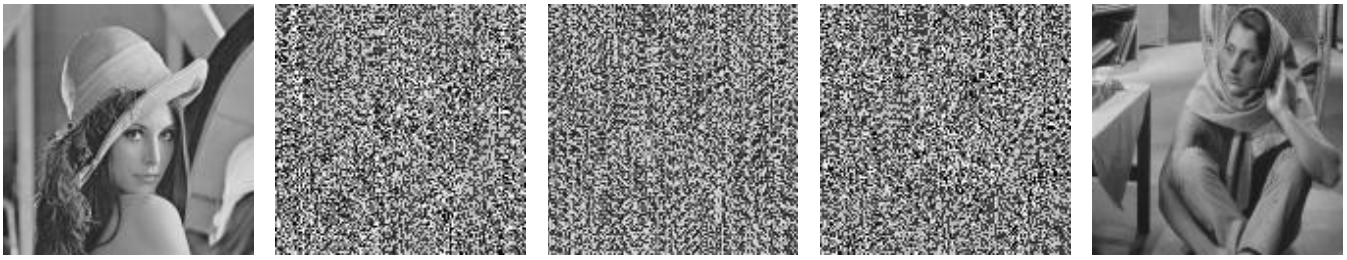
where $i = 1$ or 2, which for the original set of parameters enumerates to only $2^{24} \times 70$.

Furthermore, the original scheme is subject to well-defined chosen/known-plaintext attacks [10]. 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 [10].

In applied cryptography, 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 [10] 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 [10]. 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 [11]. Note that performing only a permutation transformation to pixel values is not sufficient either, 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



(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).

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.

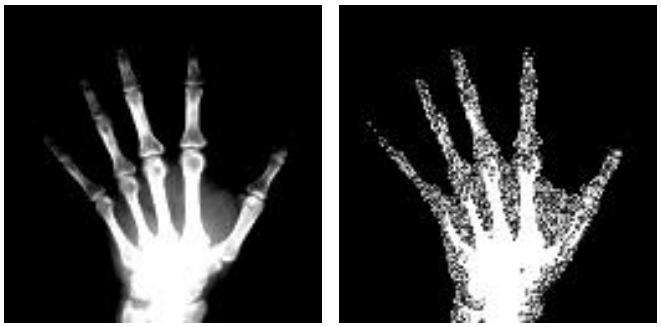


Fig. 2. Weak X-ray image encryption as a result of pixel value permutation without the substitution step (S-box).

Furthermore, an additional weakness exist in the systems where S-box consists of only one cryptographic primitive and where only one iteration of SP-network is performed during the encryption. Namely, such systems are subject to differential cryptanalysis [11]. To resist the differential chosen-plaintext attack, it is necessary to further enhance the SP-network and to introduce multiple-round iteration.

III. The Enhanced CKBA (ECKBA)

Let I be an $M \times N$ image with b-byte pixel values, where a pixel value is denoted by $I(i)$, $0 \leq i < M \times N \times b$, scanned in the raster order. Let $\mathcal{C}_\mu$ be a one-dimensional chaotic map with a real coefficient μ obtained by normalizing a 32-bit integer μ_{I32} to a chaotic interval. Let $x(0)$ be the initial condition for $\mathcal{C}_\mu$ obtained by normalizing a 32-bit integer $x(0)_{\mathrm{I32}}$ 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. In addition, we define an S-box transformation σ_r and its inverse σ_r^{-1} as follows:

$$\sigma_r(u,v) = \begin{cases} u \oplus v, & \text{if } r \text{ is even;} \\ u + v \bmod 256, & \text{if } r \text{ is odd,} \end{cases} \quad (2)$$

$$\sigma_r^{-1}(u,v) = \begin{cases} u \oplus v, & \text{if } r \text{ is even;} \\ u - v \bmod 256, & \text{if } r \text{ is odd,} \end{cases} \quad (3)$$

where u and v are two bytes.
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 that $4|r$ and $r|MNb$, where r specifies the number of rounds. The proposed encryption scheme is realized by *Algorithm 1*. In the algorithm we make use of the following notation: if x_{I32} denotes a 32-bit integer variable, then x automatically denotes its normalized floating-point representation that corresponds to the relevant real interval, and vice versa.

Algorithm 1 transforms an image I using an SP-network generated by a one-dimensional chaotic map and a 128-bit secret key. The algorithm performs r rounds of an SP-network

Data: An $M \times N \times b$ plain-image I, 128-bit key k and the number of rounds r.
Result: An $M \times N \times b$ cipher-image I'.

```
1  begin
2      x(r/4 − 1)_I32 ← l(l(k)); α_I32 ← h(l(k))
3      y(r/2 − 1)_I32 ← l(h(k)); β_I32 ← h(h(k))
4      I'(−1) ← 0
5      for i ← 0 to r/4 − 1 do
6          z(i) ← 0
7      end
8      for i ← 0 to MNb − 1 do
9          if i = 0 mod r then
10             if i > 0 then
11                 for j ← 0 to r/4 − 1 do
12                     t ← i − r + 4j
13                     z(j)_I32 ←
                       I'(t)||I'(t + 1)||I'(t + 2)||I'(t + 3)
14                 end
15             end
16             for j ← 0 to r/4 − 1 do
17                 x(j) ← C_α(x(j − 1 mod r/4))
18                 x(j) ← x(j) + z(j) mod 1
19                 c(4j) ← l(l(x(j)_I32))
20                 c(4j + 1) ← h(l(x(j)_I32))
21                 c(4j + 2) ← l(h(x(j)_I32))
22                 c(4j + 3) ← h(h(x(j)_I32))
23             end
24             for j ← 0 to r/2 − 1 do
25                 y(j) ← C_β(y(j − 1 mod r/2))
26                 x(j) ← x(j) + z(j mod r/4) mod 1
27                 d(2j) ← l(y(j)_I32) mod 8!
28                 d(2j + 1) ← h(y(j)_I32) mod 8!
29             end
30         end
31         I'(i) ← I(i) ⊕ I'(i − 1)
32         for j ← 0 to r − 1 do
33             I'(i) ← σ_j(I'(i), c(i + j mod r))
34             I'(i) ← π_{d(i+j mod 8!)}(I'(i))
35         end
36     end
37 end
```

Algorithm 1: ECKBA Encryption

on each pixel. Lines 10-30 are used to generate two pseudo random (chaotic) sequences $\{x\}$ and $\{y\}$ that are respectively used in the substitution step in line 33 and a permutation step in line 34. In lines 11-14 the next iteration of the chaotic map is controlled using the previous cipher-block, which improves the resistance against both linear and differential cryptanalysis. In addition to this, line 31 of the algorithm implements a cipher-block chaining (CBC) encryption mode.

To decrypt an encrypted image, one has to perform the sequence of inverse transformations. The decryption algorithm differs very little from the encryption algorithm (*Algorithm 1*), and the differences are summarized in the following list:

- Replace I' by I in lines 4 and 13.
- Replace line 31 by $I'(i) \leftarrow I(i)$.
- Reverse the iteration in line 32 (iterate i from $r-1$ to 0).
- Insert a new line $I'(i) \leftarrow I'(i) \oplus I(i-1)$ between lines 35 and 36.

Data: An $M \times N \times b$ cipher-image I, 128-bit key k and the number of rounds r.
Result: An $M \times N \times b$ plain-image I'.

1 **begin**
2 $x(r/4-1)_{\text{I32}} \leftarrow l(l(k))$; $\alpha_{\text{I32}} \leftarrow h(l(k))$
3 $y(r/2-1)_{\text{I32}} \leftarrow l(h(k))$; $\beta_{\text{I32}} \leftarrow h(h(k))$
4 $I(-1) \leftarrow 0$
5 **for** $i \leftarrow 0$ ***to*** $r/4-1$ **do**
6 $z(i) \leftarrow 0$
7 **end**
8 **for** $i \leftarrow 0$ ***to*** $MNb-1$ **do**
9 **if** $i = 0 \bmod r$ **then**
10 **if** $i > 0$ **then**
11 **for** $j \leftarrow 0$ ***to*** $r/4-1$ **do**
12 $t \leftarrow i - r + 4j$
13 $z(j)_{\text{I32}} \leftarrow I(t)||I(t+1)||I(t+2)||I(t+3)$
14 **end**
15 **end**
16 **for** $j \leftarrow 0$ ***to*** $r/4-1$ **do**
17 $x(j) \leftarrow \mathcal{C}_\alpha(x(j-1 \bmod r/4))$
18 $x(j) \leftarrow x(j) + z(j) \bmod 1$
19 $c(4j) \leftarrow l(l(x(j)_{\text{I32}}))$
20 $c(4j+1) \leftarrow h(l(x(j)_{\text{I32}}))$
21 $c(4j+2) \leftarrow l(h(x(j)_{\text{I32}}))$
22 $c(4j+3) \leftarrow h(h(x(j)_{\text{I32}}))$
23 **end**
24 **for** $j \leftarrow 0$ ***to*** $r/2-1$ **do**
25 $y(j) \leftarrow \mathcal{C}_\beta(y(j-1 \bmod r/2))$
26 $x(j) \leftarrow x(j) + z(j \bmod r/4) \bmod 1$
27 $d(2j) \leftarrow l(y(j)_{\text{I32}}) \bmod 8!$
28 $d(2j+1) \leftarrow h(y(j)_{\text{I32}}) \bmod 8!$
29 **end**
30 **end**
31 $I'(i) \leftarrow I(i)$
32 **for** $j \leftarrow r-1$ ***to*** 0 **do**
33 $I'(i) \leftarrow \pi^{-1}_{d(i+j \bmod 8!)}(I'(i))$
34 $I'(i) \leftarrow \sigma_j^{-1}(I'(i), c(i+j \bmod r))$
35 **end**
36 $I'(i) \leftarrow I'(i) \oplus I(i-1)$
37 **end**
38 **end**

Algorithm 2: ECKBA Decryption

- Replace σ by σ^{-1} and π by π^{-1} in lines 33-34.
- Interchange lines 33 and 34.

For clarity and completeness, we present the ECKBA decryption algorithm here as *Algorithm 2*.

In *Algorithm 1* and *Algorithm 2*, 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 considerably 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 (*Algorithm 3*).

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:

$$\begin{aligned} x(n) &= \mathcal{C}_\mu(x(n-1)) \\ &= \mu x(n-1)(1-x(n-1)), \end{aligned}$$

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 the following Zhou's map with better balance property [13]:

$$\begin{aligned} x(n) &= \mathcal{C}_\mu(x(n-1)) \\ &= \begin{cases} x(n-1) \cdot \frac{1}{\mu}, & \text{if } x(n-1) \in [0,\mu); \\ (x(n-1)-\mu)\frac{1}{0.5-\mu}, & \text{if } x(n-1) \in [\mu, 0.5]; \\ \mathcal{C}_\mu(1-x(n-1)), & \text{if } x(n-1) \in [0.5, 1); \end{cases} \end{aligned}$$

where the positive real constant $\mu \in (0, 0.5)$ and $x(i) \in (0,1)$. This map was a typical PWLCM with four linear segments and firstly used by Zhou in [13]. Its properties as a chaotic digital pseudo random sequence generator were further studied in in [14], [15]. In Section IV we show how the PWLCM map exhibits much better chaotic properties than the Logistic map.

All of the computer calculations involving enumerations of a chaotic map $\mathcal{C}_\mu$ are done in some finite 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 long periodicity. As reported previously (see Sec. 2 of [15] for a survey), when a chaotic system is created using smaller finite computing precision, the cycle length of the chaotic orbits, or the periodicity, becomes much smaller than the number of all finite states in the precision. The original CKBA uses a 16-bit precision, which is too small. For ECKBA, the double-precision (i.e., 64-bit) floating-point output is approximated and mapped into a 32-bit integer. 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 [6], 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 [11]. The white-box analysis of CKBA from [10] reveals that its actual key-size is $log_2(2^{24} \times 70)$ bits, which enumerates to about 30 bits. Since the ECKBA scheme does

	Avg. Percentage of Zeros	Avg. Percentage of Ones
S_{16}^{L}	0.47618128	0.52381873
S_{32}^{L}	0.48781678	0.51218322
S_{16}^{P}	0.49995843	0.50004157
S_{32}^{P}	0.49998275	0.50001726

TABLE I

AVERAGE PERCENTAGES OF ZEROS AND ONES FOR EACH SET.

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. PWLCM

Since the pseudo-random output of a one-dimensional chaotic map is used for both confusion and diffusion, we need a map with better statistical 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.

In chaos literature, it is well known that the Logistic map has a non-uniform invariant density function [12], that is, it has a poor balance property. On the other hand, the PWLCM map has a uniform invariant density function and resembles a much better uniformity [15]. Our experiments further confirm that PWLCM have much better balance property in comparison to 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}^{L}, S_{16}^{P}, and S_{32}^{P}, where the superscript represents the chaotic map (the superscript P stands for PWLCM 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 to the chaotic range $(0, 0.5)$ for PWLCM. By the balance property of pseudo-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, PWLCM 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.

In addition, it is found that the Zhou's PWLCM map has the following cryptographically good properties [15]: it is highly chaotic with large positive Lyapunov exponent; it is exact, mixing and ergodic; and it has an exponentially decreasing auto-correlation.

C. SP-Network of ECKBA

In Section II we have discussed that CKBA is extremely vulnerable to the chosen/known-plaintext attack. By introducing

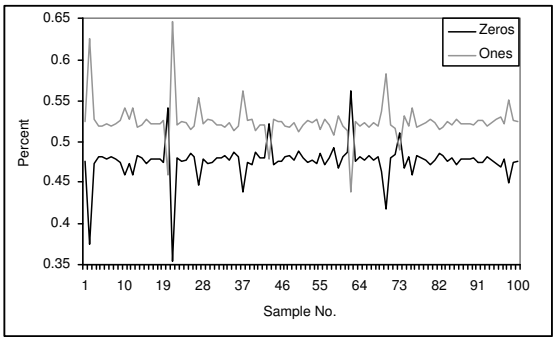


Fig. 3. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{16}^{L}.

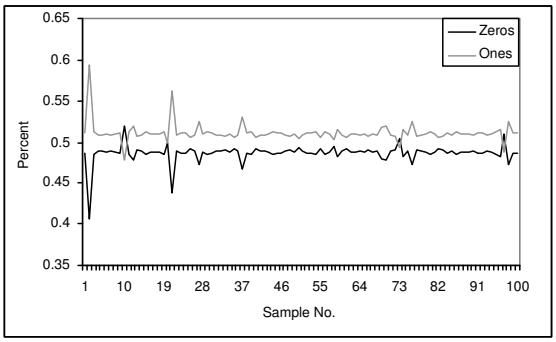


Fig. 4. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{32}^{L}.

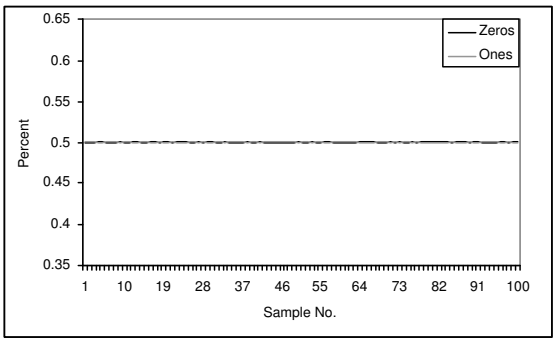


Fig. 5. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{16}^{P}.

the multi-round iteration, an S-box with two primitives, and the variable permutation component in the encryption process, we eliminate the types of attacks discussed in [10].

Suppose we have three $M \times N$ images I, I', and J'. Furthermore, suppose we know that I' is the encryption result of I using key k, and that a ciphertext image J' was encrypted using the same algorithm with the same key k. For ECKBA, since each pixel is uniquely substituted and permuted multiple times, the mask image I_m obtained by simply XOR-ing the plaintext image I with its corresponding ciphertext image I' cannot be directly applied to recover the unknown plain-image J encrypted with the same key. Figure 7 demonstrates

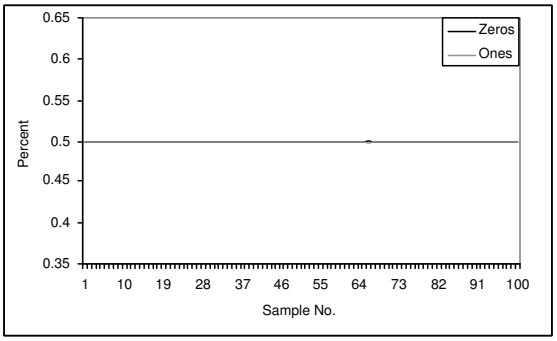


Fig. 6. The percentages of zeros and ones calculated over the sample of 100 sequences from a set S_{32}^{P}.

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

$$\begin{aligned} p \oplus p' \oplus q' &= p \oplus (p \oplus x) \oplus (q \oplus x) \\ &= p \oplus p \oplus x \oplus x \oplus q = q. \end{aligned}$$

In the case of ECKBA, the corresponding equation cannot be solved. Namely, if $p' = I'(i + jN)$ and $p^* = I'(i + jN - 1)$ then $p' = p^{(r)}$ where $p^{(0)} = p \oplus p^*$ and $p^{(i)} = \pi^{(i)}(\sigma_i(p^{(i-1)}, x^{(i)}))$, $1 \le i \le r$, for some 8-bit binary strings $x^{(1)}, \ldots, x^{(r)}$ and some permutations $\pi^{(1)}, \ldots, \pi^{(r)} \in S_8$. Similarly, if $q' = J'(i + jN)$ and $q^* = J'(i + jN - 1)$ then $q' = q^{(r)}$ where $q^{(0)} = q \oplus q^*$ and $q^{(i)} = \pi^{(i)}(\sigma_i(q^{(i-1)}, x^{(i)}))$. However, it is not clear how to combine $p^{(0)}$, p', q^* and q' to recover q without knowing the the permutations $\pi^{(1)}, \ldots, \pi^{(r)}$ and the values $x^{(1)}, \ldots, x^{(r)}$.

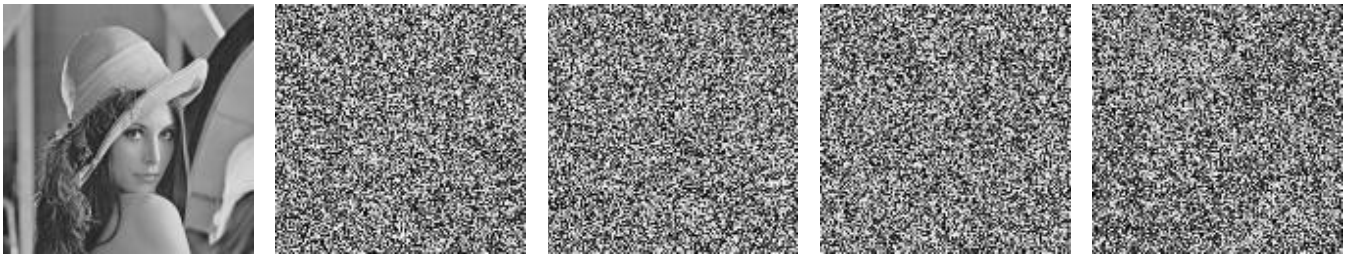
An important security weakness of CKBA is that if the images I and J are very similar, which is the case for consecutive video frames, then I' and J' will be just as similar since $I(i + jN) = J(i + jN)$ implies that $I'(i + jN) = J'(i + jN)$ if the same key was used to encrypt I and J. However, ECKBA is implemented in the cipher-block chaining (CBC) mode. In addition, in ECKBA the previous cipher block is used to control the next iteration of the chaotic map. As a result, the SP-network itself is dependant on the ciphertext (and also on the plaintext). Thus, in the case of ECKBA, $I(i + jN) = J(i + jN)$ does not imply that $I'(i + jN) = J'(i + jN)$ as long as there exist some $0 \le x < i$ and $0 \le y < j$ for which $I(x + yN) \ne J(x + yN)$.

D. On the Periodicity in ECKBA

The results from [14], [15] show that finite-precision quantization errors affect the periodicity of a chaotic map. Assuming that some finite precision is used, 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. In addition to this, the ECKBA encryption algorithm directly perturbs the chaotic orbit of the one-dimensional chaotic map. Perturbing a chaotic system usually prolongs the cycle length of a pseudo-orbit [15]. In Sec. 6.1.3. of [15], it was argued that the perturbation of the chaotic orbit has better performance than the perturbation of the control parameter.

V. Experiments

In Section IV we showed that ECKBA is much more secure than the original CKBA from [6]. In this section, we run experiments to evaluate the performance of ECKBA in comparison to CKBA. Since ECKBA introduces additional steps, and it uses higher precision and a more complex map than CKBA, its is expected that the running time of the encryption/decryption algorithm increases. We have implemented both ECKBA and CKBA methods. ECKBA was implemented both using table-lookup approach and the computational approach using *Algorithm 3* from the Appendix section. Both modes of ECKBA



(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).

used a PWLCM chaotic map iterated using a 64-bit double floating-point precision, and the resulting sequence was further clipped to a 32-bit precision. CKBA was implemented by using the Logistic map, and the precision was kept at 16 bits in order to keep the original framework described in [6]. We measured performances for selected standard images of 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 relatively small. The encryption speed is still high, but the security level is considerably improved, as shown in Section IV. For larger images, the ECKBA implementation using the computational approach for selecting random permutations performs slower. However, such an 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 the computational approach has a satisfactory performance.

Image	$M \times N \times b$	Encryption Time
boat	$128 \times 96 \times 1$	0.002 sec
mandril	$176 \times 144 \times 1$	0.004 sec
camera	$256 \times 256 \times 1$	0.008 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

TABLE II
PERFORMANCE OF CKBA ENCRYPTION

Image	$M \times N \times b$	Enc. time $r = 4$	Enc. time $r = 8$
boat	$128 \times 96 \times 1$	0.004 sec	0.008 sec
mandril	$176 \times 144 \times 1$	0.01 sec	0.02 sec
camera	$256 \times 256 \times 1$	0.04 sec	0.06 sec
barb	$512 \times 512 \times 1$	0.16 sec	0.27 sec
lena	$1024 \times 1024 \times 1$	0.65 sec	1.11 sec
tulips	$768 \times 512 \times 3$	0.73 sec	1.25 sec

TABLE III
PERFORMANCE OF ECKBA ENCRYPTION USING TABLE-LOOKUP APPROACH

Image	$M \times N \times b$	Enc. time $r = 4$	Enc. time $r = 8$
boat	$128 \times 96 \times 1$	0.01 sec	0.02 sec
mandril	$176 \times 144 \times 1$	0.03 sec	0.04 sec
camera	$256 \times 256 \times 1$	0.09 sec	0.12 sec
barb	$512 \times 512 \times 1$	0.39 sec	0.49 sec
lena	$1024 \times 1024 \times 1$	1.57 sec	1.99 sec
tulips	$768 \times 512 \times 3$	1.77 sec	2.24 sec

TABLE IV
PERFORMANCE OF ECKBA ENCRYPTION USING COMPUTATIONAL APPROACH

VI. CONCLUSIONS

We proposed a novel image encryption algorithm, called ECKBA, based on the previously proposed method by Yen and Guo [6]. Our approach resembles some similarity to the Yen-Guo approach, but attains a much higher security level. The enhanced security comes from the following changes to the original algorithm: a larger key space, a more chaotic one-dimensional map, and the use of a multi-round SP-network. As the experiments suggest, the speed of our algorithm is still very good. There are other possible improvements that could be considered. For example, the encryption process could also include a transformation of blocks of pixels, or preferably of 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 degree. Currently, we are investigating pseudo-random permutation generators based on chaotic maps and group bases (logarithmic signatures) of permutation groups [16], [17], that are capable of efficiently producing permutations of much higher degrees.

APPENDIX

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

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

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 a 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 3 can be utilized for ECKBA encryption and decryption in applications where a table-lookup is too memory intensive or demanding for the particular device in use. Utilizing such a computational approach decreases the amount of needed memory, but affects the performance (see Section V for experimental results).

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

begin
  $m \leftarrow n - 1$
  $\tau \leftarrow$ the identity of S_n
  for $0 \leq i < n$ **do**
    $\pi_x[i] \leftarrow \tau[\lfloor \frac{x}{m!} \rfloor]$
    for $\lfloor \frac{x}{m!} \rfloor \leq j < m$ **do**
      $\tau[j] \leftarrow \tau[j+1]$
    end
    $x \leftarrow x \bmod m!$
    $m \leftarrow m - 1$
  end
end

Algorithm 3: Computing the permutation for given index.

ACKNOWLEDGMENTS

The work of the authors D.S. and S.S.M. was partially supported by a Federal Earmark grant for *Research in Secure Telecommunication Networks* (2004-05).

REFERENCES

[1] B. Furht and D. Socek. Multimedia security: encryption techniques. In *IEC Comprehensive Report on Information Security*, International Engineering Consortium, Chicago, IL, pages 335–349, 2004.

[2] S. Li, G. Chen, and X. Zheng. Chaos-based encryption for digital images and videos, in B. Furht and D. Kirovski (Eds.), Multimedia Security Handbook, Vol. 4 of *Internet and Communications Series*, Ch. 3, CRC Press, December 2004.

[3] B. Furht, D. Socek, and A.M. Eskicioglu. Fundamentals of multimedia encryption techniques, in B. Furht and D. Kirovski (Eds.), Multimedia Security Handbook, Vol. 4 of *Internet and Communications Series*, Ch. 3, 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] H. Cheng and X. Li. Partial encryption of compressed images and videos. In *IEEE Transactions on Signal Processing*, volume 48, pages 2439–2451, 2000.

[6] J.-C. Yen and J.-I. 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.

[7] B. Bhargava, C. Shi, and S.-Y. Wang. MPEG video encryption algorithms. Multimedia Tools and Applications, Kluwer Academic Publishers, Vol. 24, No. 1, pages 57–79, 2004.

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

[9] T. Seidel, D. Socek and M. Sramka. Cryptanalysis of video encryption algorithms. In *Proceedings of The 3rd Central European Conference on Cryptology (TATRACRYPT '03)*, Bratislava, Slovak Republic, June 26-28 (2003), Tatra Mt. Mathematical Publications, Vol. 29, pages 1–9, 2004.

[10] S. Li and X. 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.

[11] D.R. Stinson. *Cryptography: theory and practice*. CRC Press, second edition, 2002.

[12] A. Lasota and M.C. Mackey. Chaos, fractals, and noise– stochastic aspects of dynamics. Springer-Verlag, New York, 2nd Ed., 1997.

[13] H. Zhou. A design methodology of chaotic stream ciphers and the realization problems in finite precision. Ph.D. thesis, Department of Electrical Engineering, Fudan University, Shanghai, China, 1996.

[14] S. Li, Q. Li, W. Li, X. Mou 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.

[15] S. Li, G. Chen and X. Mou. On the Dynamical Degradation of Digital Piecewise Linear Chaotic Maps. accepted by the Tutorial-Review section of International Journal of Bifurcation and Chaos in August 2004, tentatively scheduled for publication in vol. 15, no. 10, 2005.

[16] S.S. Magliveras. A cryptosystem from logarithmic signatures of finite groups. In *Proceedings of the 29th Midwest Symposium on Circuits and Systems*, pages 972–975. Elsevier, Ampsterdam, 1986.

[17] S.S. Magliveras and N.D. Memon. Random permutations from logarithmic signatures. In *Computing in the 90's - First Great Lakes Computer Science Conference*, volume 507 of *Lecture Notes in Computer Science*, pages 91–97. Springer-Verlag, Berlin, 1989.