Short Paper: Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption

Daniel Socek*, Shujun Li[†], Spyros S. Magliveras[‡] and Borko Furht[§]
*[‡]Center for Cryptology and Information Security and *[§]Department of Comp. Sci. and Engineering
Florida Atlantic University, 777 Glades Road, Boca Raton, Florida 33431–0991
[†]Department of Electronic and Information Engineering
Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, China
Emails: *dsocek@fau.edu, [†]www.hooklee.com, [‡]spyros@fau.edu, [§]borko@cse.fau.edu

Abstract

A recently proposed Chaotic-Key Based Algorithm (CKBA) has been shown to be unavoidably susceptible to chosen/known-plaintext attacks and ciphertext-only attacks. 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.

1 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. Unfortunately, in many applications, conventional encryption algorithms (such as AES) are not suitable for image and video encryption [1, 2]. In order to overcome this problem, many fast encryption algorithms specifically designed for digital images have been proposed [3, 4]. However, a number of these algorithms have been shown to be insecure [5, 6].

The image encryption methods based on chaotic maps attract considerable attention recently due to their potential for digital multimedia encryption [2]. In [3], Yen and Guo proposed a chaotic key-based algorithm (CKBA) for image encryption. Subsequently, Li and Zheng [6] showed that the security claims for CKBA have been vastly overestimated and that ECKBA is susceptible to several types of attacks. In this paper, we propose a more secure cryptosystem that is based on the ideas from the original CKBA.

Following the suggestions in [6] 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) from [7], 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, with an acceptable loss in speed.

2 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:

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

$$\sigma_r^{-1}(u,v) = \begin{cases} u \oplus v, & \text{if } r \text{ is even;} \\ u - v \bmod 256, & \text{if } r \text{ is odd,} \end{cases} \tag{2}$$

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

Algorithm 1: ECKBA Encryption

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 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 inverse transformations (*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 (*Algorithm 3*).

```
**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 $I'$.
begin
    $x(r/4-1)_{\mathrm{I32}} \leftarrow l(l(k)); \alpha_{\mathrm{I32}} \leftarrow h(l(k))$
    $y(r/2-1)_{\mathrm{I32}} \leftarrow l(h(k)); \beta_{\mathrm{I32}} \leftarrow h(h(k))$
    $I(-1) \leftarrow 0$
    for $i \leftarrow 0$ to $r/4-1$ do
        $z(i) \leftarrow 0$
    end
    for $i \leftarrow 0$ to $MNb-1$ do
        if $i = 0 \mod r$ then
            if $i > 0$ then
                for $j \leftarrow 0$ to $r/4-1$ do
                    $t \leftarrow i - r + 4j$
                    $z(j)_{\mathrm{I32}} \leftarrow I(t)||I(t+1)||I(t+2)||I(t+3)$
                end
            end
            for $j \leftarrow 0$ to $r/4-1$ do
                $x(j) \leftarrow \mathcal{C}_{\alpha}(x(j-1 \mod r/4))$
                $x(j) \leftarrow x(j) + z(j) \mod 1$
                $c(4j) \leftarrow l(l(x(j)_{\mathrm{I32}}))$
                $c(4j+1) \leftarrow h(l(x(j)_{\mathrm{I32}}))$
                $c(4j+2) \leftarrow l(h(x(j)_{\mathrm{I32}}))$
                $c(4j+3) \leftarrow h(h(x(j)_{\mathrm{I32}}))$
            end
            for $j \leftarrow 0$ to $r/2-1$ do
                $y(j) \leftarrow \mathcal{C}_{\beta}(y(j-1 \mod r/2))$
                $x(j) \leftarrow x(j) + z(j \mod r/4) \mod 1$
                $d(2j) \leftarrow l(y(j)_{\mathrm{I32}}) \mod 8!$
                $d(2j+1) \leftarrow h(y(j)_{\mathrm{I32}}) \mod 8!$
            end
        end
        $I'(i) \leftarrow I(i)$
        for $j \leftarrow r-1$ to $0$ do
            $I'(i) \leftarrow \pi^{-1}_{d(i+j \mod 8!)}(I'(i))$
            $I'(i) \leftarrow \sigma^{-1}_{j}(I'(i), c(i+j \mod r))$
        end
        $I'(i) \leftarrow I'(i) \oplus I(i-1)$
    end
end
```

Algorithm 2: ECKBA Decryption

```
**Data**: Index $x$ satisfying $0 \leq x < n!$, and the permutation degree $n$.
**Result**: Permutation $\pi_x$.
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
        $x \leftarrow x \mod m!$
        $m \leftarrow m-1$
    end
end
```

Algorithm 3: Computing the permutation for 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 Logistic map. However, due to the poor balance property of the Logistic map, we recommend ECKBA (and CKBA) implementations to use the following Zhou's map with better balance property:

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

where the positive real constant $\mu \in (0, 0.5)$ and $x(i) \in (0, 1)$. It was found that the Zhou's PWLCM map has the

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 1. 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 2. 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 3. Performance of ECKBA encryption using computational approach

following cryptographically good properties [7]: it is highly chaotic with large positive Lyapunov exponent; it is exact, mixing and ergodic; and it has an exponentially decreasing auto-correlation.

3 Experiments

Since ECKBA introduces additional steps and uses a more complex map than CKBA, its is expected that the running time of the encryption/decryption algorithm increases. We implemented both ECKBA and CKBA methods. ECKBA was implemented both using table-lookup approach and the computational approach using *Algorithm 3*. Both modes of ECKBA 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 [3]. The performance experiments were run on a 1.3GHz Intel(R) Pentium(R) M processor, and the results are summarized in Tables 1, 2 and 3. The experimental results suggest that the ECKBA algorithm implemented using a table-lookup approach is reasonably fast. 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.

4 Conclusions

We proposed an image encryption algorithm, called ECKBA, based on the previously proposed method by Yen and Guo [3]. Our approach resembles some similarity to the Yen-Guo approach, but attains a much higher security level. As the experiments suggest, the performance of our algorithm is still very good for most applications.

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.