Kaos Tabanlı Bir Şifreleme Yöntemi ve Analizi

Mir Mohammad Reza Alavı Milani, Hüseyin Pehlivan, Sahereh Hosein Pour
Karadeniz Üniversitesi, Bilgisayar Mühendisliği Bölümü, Trabzon
milani@ktu.edu.tr, pehlivan@ktu.edu.tr, hoseinpour@ktu.edu.tr

Özet: Günümüz bilgisayar destekli şifreleme teknikleri oldukça yüksek düzeyli bilgi gerektiren karmaşık güvenlik önlemleriyle yoğrulmuş teknikler içerir. Öncekilerden daha güvenli olduğu sanılan her bir yeni tekniğin zaman içerisinde başka güvenlik açıklarının bulunduğuna şahit olmaktayız. Dolayısıyla, temel ilke olarak herhangi bir şifreleme yönteminin kırılamaz olmadığını ve sonlu bir süre sonunda şifresinin çözülebileceğini söyleyebiliriz. Görüntü verilerine uygulanabilen şifreleme yöntemlerinin sayısı da günden güne artmaktadır. Bu çalışmada, Henon kaotik sistemleri ile lojistik haritanın rastgele özelliklerinden yararlanılarak görüntü şifrelemede kullanılabilecek hızlı bir algoritma geliştirilmiştir. Siyah-beyaz ve renkli resimler üzerindeki uygulamalardan elde edilen sonuçlar algoritma güvenliğinin yüksek olduğunu göstermektedir.

Anahtar Sözcükler: Kaos, Görüntü Şifreleme, Lojistik Harita, Rasgele.

A Chaotic Based Encryption Method and Its Analyse.

Abstract: Today's computer-aided encryption techniques requires knowledge of very complicated and complex security measures. While each claims to be more secure than the previous, with every coming days we are witnessing how the previous passwords are broken. Therefore, based on the basic principles learned in theory, it is possible to say that any encryption method cannot become "unbreakable" and any password can break in a limited period of time. The number of encryption methods on images is increasing gradually. In this study, by means of a logistic map, we propose a simple and fast encryption algorithm to encrypt the images, using the Henon chaotic systems, and logistic properties of random maps. This algorithm is applied to both the black-and-white and color images. The results indicate a greater security of the proposed algorithm.

Keywords: Chaos, image encryption, logistic map, random numbers

1. Giriş

Internet ve kablosuz ağlar üzerinde görüntü şifreleme ve güvenli görüntü iletim sistemlerinin önemi giderek artmakla birlikte, görüntü ve video boyutlarının büyük olmasından dolayı AES, DES, IDEA, RSA [1] gibi klasik algoritmaların kullanılması uygun görünmemektedir. Özellikle gerçek zamanlı sistemlerde, video konferans gibi uygulamalarda, bu tür algoritmaların hızları düşük olduğundan kullanılmamaktadır. Bu zorlukların üstesinden gelebilmek için, çok fazla sayıda mültimedya şifreleme algoritmaları önerilmiştir [2,3]. Bu algoritmaların bir başka sorunu anahtar boylarıdır ve eğer bir şifreli veri küçük boyutlu anahtar ile kullanılırsa, ataklar karşısında zayıf kalır. Genel bir tasarım ilkesi olarak, şifrelemede temel blokların düzeltilmesinde doğrusal olmayan fonksiyonlar kullanılmaktır [4]. Ayrık ve sürekli zaman kaotik sistemlerini birleştiren daha karmaşık bir sistem Guan ve arkadaşları tarafından tasarlanmıştır [5]. Ayrıca başka bir yöntem de hızı ve güvenliği artırmak için geliştirilmiştir [6]. Örneğin, kaotik sistem özelliklerini kullanan birkaç algoritma da önerilmiştir [7,8]. Kaotik algoritmalar değişik bir yol kullanır; bu algoritmalar çok basittir ve hesaplama maliye-

leri azdır, ama görüntü şifreleme için çok iyi olabilirler, bu durum kaotik sistemlerin başlangıç değeri, sistem parametreleri ve random özelliklerine dayanır. Kaotik sistemlere dayalı algoritmalar basit olduğundan dolayı, bu algorıtmalarla yapılan sistemlerin hızı daha yüksek olabilmektedir. Blok metodunda bu tarz algoritmalar kullanırsa, blok ve iterasyon sayısının kontrolüyle bu algoritmaların hesaplama hızı ve hassasiyeti uygun şekilde seçilebilmektedir. Son olarak, kaotik sistemleri kullanan metotlarda, sistemin anahtar değişikliğine çok hassas olduğunu göz önüne alarak, bu sistemlerin daha güvenli olduğunu söyleyebiliriz.

Bu çalışma aşağıdaki gibi yapılandırılmıştır. 2. Bölümde, lojistik haritanın özellikleri tartışılmıştır. 3. Bölümde, bir rastgele sayı jeneratörü ve 4. Bölümde, şifreleme yöntemi önerilmiştir. 5. Bölümde ise önerilen yöntem analiz edilmiştir.

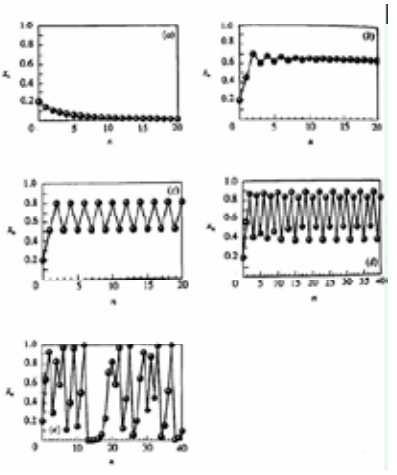
2. Lojistik Harita

Bilgi taşımak için kaotik sinyallerin kullanılması, ilk olarak Hayes ve arkadaşları tarafından 1993 yılında ortaya atılmıştır [9]. Kaos tabanlı şifreleme programları temelde kaotik denklemleri kullanarak sözde rastgele sayı üreteçleri gibi uzun bir rastgele sayı dizisi üretip bu dizi ile bir düz görüntüyü şifrelerler [10]. Basit ve en çok çalışılan doğrusal olmayan sistemlerden biri lojistik haritadır. Bu sistem aslında 1838 yılında Pierre Franois Verhulst tarafından demografik bir model olarak tanıtılmıştır. 1947 yılında, Ulam ve von Neumann [11] rastsal sayı üreteci olarak lojistik haritayı çalıştı. Görüntülerin şifrelenmesinde, lojistik haritaları, onların başlangıç koşullarına hassas bağımlılığı, rastgeleye benzer davranış göstermesi ve tekrarlı olmayan özellikleri içermesinden dolayı S-box kutularının yerine kullanılır [10]. Kaos tabanlı şifreleme programları temelde, kaotik haritaları kullanarak rastsal sayı üreteçleri olarak bir uzun rastgele sayı dizisi üreterek düz görüntüyü bu rastgele sayılarla şifrelerler [12,10].

Lojistik harita aşağıdaki gibi verilir:

$$X_{n+1} = \mu X_n(1 - X_n) \qquad (1)$$

Burada sırasıyla $X_n \in (0,1)$ ve μ sistem değişkeni ve parametresi, n ise yineleme sayısıdır. Böylece, bir başlangıç değeri x0 ve bir parametre k alarak, $\{X_n\}_{n=0}^{\infty}$ serisi hesaplanır. Bu çalışmada, X_0 ve μ değerleri lojistik haritanın başlangıç değerleri olarak adlandırılacaktır. Bu başlangıç değerlerinin, özellikle μ değerinin, lojistik haritada çok önemli bir işlevi vardır. Bu önemi göstermek için aşağıdaki durumu elde alalım: Şekil 1'de lojistik haritanın zaman içinde elde edilen miktarları $X_0 = 0.2$ ve farklı μ değerleri gösterilmektedir.

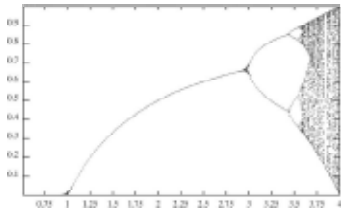


Şekil 1: (a)$\mu = 0.9$,(b)$\mu = 2.6$,(c) $\mu = 3.2$,(d)$\mu = 3.5$,(e) $\mu = 4$

Şekil 1'e göre lojistik harita $\mu = 0.9, 2.6, 3.2$ değerlerinde değil, $\mu = 3.5$,4 değerlerinde kaotik özellikler gösterir. Bir başka gösteri ile μ değerinin farklı miktarlardan lojistik haritanın ne kadar etkilendiği çatallanma[1] diyagramı ile Şekil 2'de gösterilmiştir. Bu bir μ fonksiyonu olarak, lojistik haritanın bir komplosudur. $0 \le \mu \le 1$ için elde edilen çözüm sadece bir sabit noktadır. $1 < \mu \le 3$ için, yine sabit bir nokta

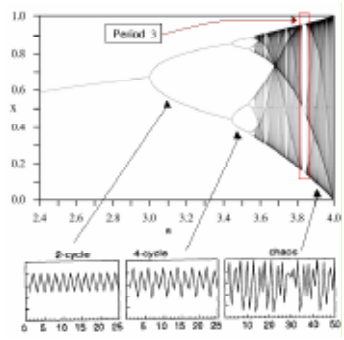
1 bifurcation

vardır. $3 < \mu \leq 3.57$ arasında, haritanın iki katına çıkarılması sergilenir. $3.57 < \mu < 4$ için, harita kaotik olur. Nihayet, $\mu = 4$ durumunda, kaos 0-1 arasında çeşitli değerlerin oluşturulduğunu görüyoruz.



Şekil 2: Çatallanma (bifurcation) diyagramı

Şekil 3'de, Şekil 2'de açıkça görünmeyen 2.4 ile 4.0 noktaları arasındaki harita özellikleri daha ayrıntılı olarak gösterilmiştir.



Şekil 3: Şekil 2 nin $2.4 \leq \mu \leq 4.0$ diyagramı

Bu çalışmada, rastgele sayıların oluşturulması için, lojistik haritayı aşağıdaki gibi kullanacağız:

$X_{n+1} = \mu X_n(1-X_n)$, for $X_n \in (0,1)$, and $\mu \in (3.9996,4]$

3. Önerilen Rastgele Sayı Jeneratörü

Kaotik özellikleri kullanan şifreleme yöntemleri, genellikle kaotik sistemlerden oluşturulan rastgele sayılar kullanırlar [10,12]. Bu çalışmada önerilen yöntemde lojistik harita aşağıdaki gibi kullanılarak birbirinden farklı 256 rastgele sayı üretilmiştir:

$X_{n+1} = \mu X_n(1-X_n)$, for $X_n \in (0,1)$, and $\mu \in (3.9996,4]$

Yöntemi görüntü şifreleme işlemlerinde ullanırken μ 'nın değeri 3.99999 olarak seçilmiştir. Elde edilen X_n'ler $[0,1]$ arasında olacağından, bu aralığı 256 parçaya bölmek için bir $\varepsilon = \frac{1}{256}$ parametresi tanımlanmıştır. Böylece [0,1] aralığında bulunan i. parça $((i-1)\varepsilon, i\varepsilon)$ arasında olacaktır. Lojistik haritanın kullanımında başlangıç değeri olarak X_0, algoritmanın anahtarından seçilir. Anahtar kelime, en fazla 80 bitten oluşan bir kelime veya herhangi bir veri olabilmektedir. Bu veriyi 10 ASCII karakteri olarak (her biri 8 bit) $K_0, K_1, K_2, ..., K_9$ biçiminde ifade edebiliriz ve buradaki her bir K_i'yı da 8 bit'ten meydana geldiğinden $K_{i1}, K_{i2}, K_{i3}, ..., K_{i8}$ gibi gösterebiliriz. Bir rastgele sayı listesi oluşturan algoritma aşağıda sunulmuştur:

$\varepsilon \leftarrow \frac{1}{256}$

$\mu \leftarrow 3.9999$

$X_0 \leftarrow [K_0 * 2^9 + K_0 * 2^8 + K_0 * 2^7 + ... + K_9 * 2^0 +]/2^8$

Yeni X_i ($X_{i+1} \leftarrow \mu X_i(1-X_i)$)

$R \leftarrow$ Yeni X_i in ait olduğu parça

Eğer R önceden iterasyon listesinde yoksa

R'yi listeye ekle

(d),(e) ve (f) adımlarını listede 256 sayı olana kadar tekrarla.

Bu algoritma ile 256 sayılı (0-255 arasında ve tekrarsız) bir iterasyon listesi oluşturulmuştur. Bir sonraki aşamada bu sayılar kullanılarak, bir görüntü verisi şifrelenmiştir. Aşağıdaki C kodu ile bir anahtar kelimeden (KeyStr) bir X_0 değeri hesaplanmaktadır.

```
double Createx0(String KeyStr){
```

```
  int n,k=8;
  double sum=0;
  n=KeyStr.Length();
  for(int i=1;i<=n;i++,k+=8)
       sum+=(double)KeyStr.operator
[](i)*pow(2,k);
  sum+=(double) KeyStr.operator []
(1)*pow(2,k);
  k+=8;
  return  sum/pow(2,k);
}
```

Bu X_0 değeri ile iterasyon listesi oluşturan C kodu ise aşağıda gösterilmiştir.

```
double İtr_Creator(String KeyStr,int
chk,int itr[]){
double xx;
x0 = createx0(KeyStr);
xx=x0;
R= 3.9999;
itration[0]= (int) (x0*256)+1;
for(int i=1;i<256;i++)
   while(1){
          x1=(double) r*x0*(1-x0);
          x0=x1;

      int xn=(int) (x1*256);
      int chkFound=0;
      for(int k=0;k<i;k++)
           if(itration [k]==xn)
chkFound=1;

      if(chkFound==0){
          itration [i]=xn;
          break;
     }
  }
 return xx;
}
```

4. Önerilen Şifreleme Yöntemi

Tüm görüntüler sınırlı sayıda pixel'lerden oluşur. Bu pixel'ler aslında 0 ile 255 arasında bir değere sahiptir ve pixel'in rengi bu değerlerle temsil edilir. Bu özellik hem siyah-beyaz hem de renkli görüntülerde geçerlidir. Ancak renkli görüntülerde her pixel için 3 farklı deger vardır ve bu değerler sıra ile Kırmızı , Yeşil ve Mavi renk bileşenlerini oluştururlar.

Önceki aşamada elde edilen rastgele sayıları iterasyon kümesi olarak adlandırarak, bir görüntünün pixel'lerini şifreleyebiliriz. Bunun için görüntü dosyasından tüm pixel'leri okuyarak aşağıdaki algoritma yardımıyla yeni değerler elde edilir.

Şifreleme algorıtması:

Görüntünün tüm değerlerini bir P listesine yerleştir. $P = \{p_1, p_2, p_3 ..... p_{m*n}\}$ (m,n: görüntünün boyutları).

Tüm $P_i = j$'ler için C_i'leri hesapla, $C_i = Iteration(pos)$, burada

Pos = (i+k) mod 256 ve Iteration(k) = j

Böylece bir düz görüntüden şifrelenmiş görüntüye dönüşüm yapılabilir. Algoritmaya göre önce bir pixelin değeri seçilir, sonra o değerin iterasyon kümesindeki karşılığı olan index ile bu pixelin bulunduğu index toplanarak elde edilen değerin 256 ile modu hesaplanır. Bu son değer iterasyon kümesinde tekrar index olarak kullanılarak diğer bir değere erişilir ve bu değer orijinal pixel değerinin şifrelenmiş karşılığı olarak alınır. Bu işlem tüm pixel'lere uygulandığında tüm görüntü şifrelenmiş olacaktır.

Bir şifrelenmiş görüntüyü deşifre etmek için, aynı rastgele sayılar yeniden üretilerek bir iterasyon kümesinde tutulur ve şifreleme algoritması tekrarlanarak düz görüntü elde edilebilir.

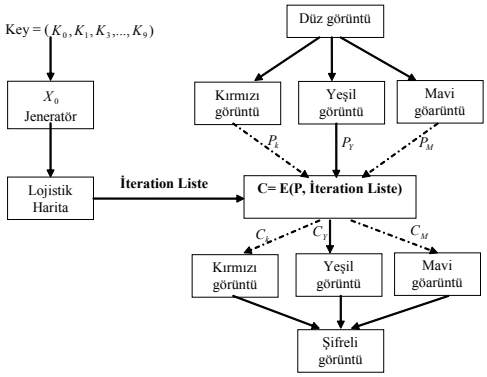
Bu algoritmanın C kodu aşağıda verilmiştir.

```
for(int n=0;n<Image0->Width;n++)
  for(int m=0;m<Image0->Height;m++){
     int i=(n*Image0->Width)+m;
     R = GetRValue(Image0->Canvas-
>Pixels[n][m]);
```

```
      for (int s=0;s<255;s++)
          if (iteration[s]==R)
{k=s;break;}
      int pos=(i+k) % 255;
      ppR= iteration [pos];

      Image1->Canvas->Pixels[n][m] =
RGB(ppR,ppR,ppR);
    }
```

Önerilen yöntemin akış şeması, Şekil 4'te gösterilmiştir.



Şekil 4: Önerilen Yöntemin Akış şeması

5. Önerilen Yöntemin Analizi

Yöntemin şifreleme işlemlerinde kullanılıp kullanılamayacağını göstermek için aşağıda bazı analizler yapılmış ve deney sonuçları irdelenmiştir.

5.1 Rastgele Sayı Jeneratörün Analizi

Kullandığımız yöntem birbirinden farklı 256 rastgele sayı dizisi gerektirdiği ve bu sayıların üretimi lojistik harita ile yapıldığından dolayı, belli bir anahtar ve farklı | değerleri ile bu dizi elemanlarının üretilebilmesi için çeşitli denemeler yapılmıştır. Bu denemelerin sonuçları Tablo 1'de verilmiştir.

Tablo 1'deki deneme sonuçlara göre, | değerinin algoritmanın doğru çalışmasını ve hızını etkilediğini söyleyebiliriz.

Sayı =256, anahtar = exam			
| değeri	Bulunan eleman sayısı	Kalan eleman sayı	Deneme sayısı
4.10	59	197	100000
4.05	111	145	100000
4.01	89	167	100000
4.00	256	0	2539
3.99999	256	0	1687
3.9999	256	0	1936
3.999	256	0	2053
3.99	254	2	100000
3.9	226	30	100000
3.8	198	58	100000
3.7	172	84	100000
3.6	102	154	100000
3.5	10	246	100000

Tablo 1: Çeşitli | değerleri ile deneme sonuçları

5.2 Güvenlik Analizi

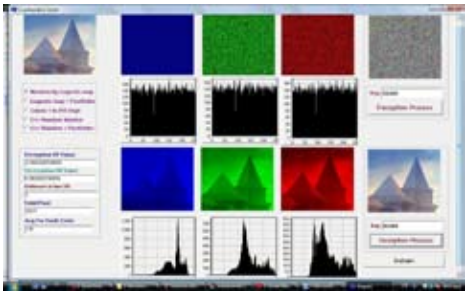
Yöntemin güvenilirliğini göstermek için burada birkaç analiz daha yapılmıştır. Bu analizler üç başlık altında toplanabilir:

- Histogram Analizi
- Korelasyon Katsayısı Analizleri[2]
- Bilgi Entropi[3]

5.2.1 Histogram Analizi

Düz görüntü ve şifrelenmiş görüntünün histogramı Şekil 5'te gösterilmektedir. Burada gösterilen düz görüntü renkli olduğundan dolayı, histogramlar hem düz hem de şifreli görüntüler üzerindeki kırmızı, yeşil ve mavi renk dağılımına göre yapılmıştır. Şekil 5'e göre düz görüntülerin histogramının istatistiksel analize ne kadar elverişli olduğu ve önerilen yaklaşımın istatistiksel analize karşı ne kadar sağlam durduğu açıkça görülmektedir.

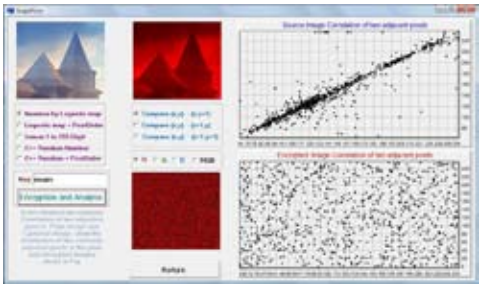
2 correlation coefficient analyses
3 Information Entropy



Şekil 5: Düz ve şifreli görüntüler histogramı

5.2.2 Korelasyon Katsayısı Analizleri

Basit korelasyon analizi, iki değisken arasındaki ilişkinin düzeyini (derecesini-siddetini-gücünü) ve yönünü belirlemek amacıyla yapılır. Her iki değişkenin de sürekli değişken olması ve değişkenlere ilişkin verilerin normal dağılım göstermesi durumunda değişkenler arasındaki ilişki Pearson korelasyon katsayısı ile belirlenir. Korelasyon katsayısı ile belirlenen ya da ölçülen, söz konusu değişkenler arasındaki doğrusal ilişkidir. Eğer değişkenler arasındaki ilişki doğrusal değil ise hesaplanan korelasyon katsayısı değişkenler arasındaki ilişkiyi ölçmek için uygun bir istatistik değildir. Burada görüntülerin komşu pixel'ler arasında doğrusal ilişkilerinin olup olmadığını belirlemek için Şekil 6'da gösterilen analiz yapılmıştır .



Şekil 6: Korelasyon Katsayısı Analizi

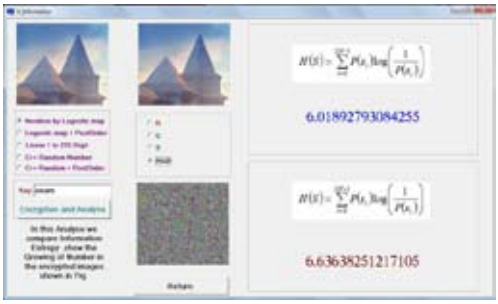
Şekil 6'ya göre düz görüntüde bu ilişki doğrusal ve şifreli görüntüde ise doğrusal değildir. Bundan dolayı, yapılan şifreleme işleminin istatistiksel analizlere kapalı olduğu sonucunu kolayca çıkarabiliriz.

5.2.3 Bilgi Entropi

Bilgi teorisinde, entropy rastgele sayılar arasında belirsiz bir ilişkiyi bulmak demektir. Bu terim aslında **Shannon entropy'**sine dayandırılmıştır ve kısaca aşağıdaki denklem ile ifade edilebilir :

$$H(S) = \sum_{i=0}^{2^N-1} P(s_i) \log\left(\frac{1}{P(s_i)}\right) \qquad (2)$$

Bu ifadenin küçük değerler üretmesi istatistik analizlerde kullanımının daha uygun olacağı anlamına gelir. Bir şifrelenmiş görüntünün başka yönlerden güvenli olup olmadığını araştırmak için, bilgi entropy'sinden yararlanabiliriz. Bunun için her görüntünün pixel değerleri P ile temsil edilip görüntü boyutu da N*N alınırsa, denklem (2) düz ve şifreli görüntülerin her biri için H(S) değerlerini hesaplayabilir. Şekil 7'de bu değerler düz ve şifrelenmiş görüntüler için gösterilmiştir.



Şekil 7 : Bilgi Entropy analisi

Şekil 7'deki veriler önerilen yöntemin başarılı olduğunu ortaya koymaktadır.

6. Sonuç

Bu çalışmada bir görüntü şifreleme yöntemi önerilmiş olup, yöntemin güvenliği düz görüntüler ile şifrelenmiş görüntüler arasında gerçekleştirilen dönüşümler göz önünde bulundurularak analiz edilmiştir. Önerilen yöntem 256 elemanlı bir rastgele sayılar listesine dayandırılmıştır ve bu listedeki elemanların rastgeleliğini artırmak için Lojistik harita kullanılmıştır.

Bu yol içerisinde üretilen rastgele sayılar ağırlıklı olarak başlangıç değerlerine bağlı olacaktır ve dolayısı ile kullanılan anahtar kelimenin daha hassas olduğu söylenebilir.

7. Kaynaklar

[1] Daemen J, Sand B, Rijmen V. The design of Rijndael: AES – the advanced encryption standard. Berlin: Springer-Verlag; 2002.

[2] Socek D, Magliveras S, C'ulibrk D, Marques O, Kalva H, Furht B. Digital video encryption algorithms based on correlation-preserving permutations. EURASIP J Inform Security 2007.

[3] Chang C, Hwang M, Chen T. A new encryption algorithm for img. cryptosystems. J Syst Software 2001;58:83–91.

[4] Preneel B. Design principles for dedicated hash functions. In: Fast software encryption, Cambridge security workshop, Lecture notes in computer science, vol. 809, Springer, Berlin; 1993. p. 71–82.

[5] Guan Z H, Huang F, Guan W. Chaos based image encryption algorithm. Phys Lett A 2005;346:153–7.

[6] Menezes AJ, van Oorschot PC, Vanstone SA. Handbook of applied cryptography. CRC Press; 1997.

[7] Pareek NK, Patidar V, Sud KK. Image encryption using chaotic logistic map. Image Vision Comput 2006;24:926–34.

[8] Chen G, Mao Y, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fract 2004;21:749–61.

[9] Hayes S, Grebogi C, Ott E. Communicating with chaos. Phys Rev Lett 1993;70(20):3031–4.

[10] Pisarchik AN, Flores-Carmona NJ, Carpio-Valadez M. Encryption and decryption of images with chaotic map lattices. Chaos: Interdiscipl J Nonlinear Sci2006;16(3):033118.

[11] Ulam SM, von Neumann J. On combination of stochastic and deterministic processes. Bull Am Math Soc 1947;53:1120.

[12] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos 1998;8:1259–84.