International Journal of Bifurcation and Chaos, Vol. 8, No. 6 (1998) 1259–1284

SYMMETRIC CIPHERS BASED ON TWO-DIMENSIONAL CHAOTIC MAPS

JIRI FRIDRICH
Center for Intelligent Systems,
Department of Systems Science and Industrial Engineering,
SUNY Binghamton, Binghamton, NY 13902-6000, USA

Received July 1, 1997; Revised December 8, 1997

In this paper, methods are shown how to adapt invertible two-dimensional chaotic maps on a torus or on a square to create new symmetric block encryption schemes. A chaotic map is first generalized by introducing parameters and then discretized to a finite square lattice of points which represent pixels or some other data items. Although the discretized map is a permutation and thus cannot be chaotic, it shares certain properties with its continuous counterpart as long as the number of iterations remains small. The discretized map is further extended to three dimensions and composed with a simple diffusion mechanism. As a result, a symmetric block product encryption scheme is obtained. To encrypt an $N \times N$ image, the ciphering map is iteratively applied to the image. The construction of the cipher and its security is explained with the two-dimensional Baker map. It is shown that the permutations induced by the Baker map behave as typical random permutations. Computer simulations indicate that the cipher has good diffusion properties with respect to the plain-text and the key. A nontraditional pseudo-random number generator based on the encryption scheme is described and studied. Examples of some other two-dimensional chaotic maps are given and their suitability for secure encryption is discussed. The paper closes with a brief discussion of a possible relationship between discretized chaos and cryptosystems.

1. Introduction

The idea of using chaos for data encryption is certainly not new and can be traced to the classical Shannon's paper [1952]. Even though he does not use the word chaos, he proposes mixing, measure preserving transformations which depend on their arguments in a "sensitive" way. He explicitly mentions the basic stretch-and-fold mechanism of chaos: *"... Good mixing transformations are often formed by repeated products of two simple noncommuting operations. Hopf has shown, for example, that pastry dough can be mixed by such a sequence of operations. The dough is first rolled out into a thin slab, then folded over, then rolled, and then folded again, etc..."* Sloane [1982] also points out the importance of chaos for generating random permutations in groups.

Chaotic maps have been utilized in several different ways in cryptography. Matthews [1989] derives a one-dimensional chaotic map which exhibits chaotic behavior for parameter values and initial values within a specified range. He suggests to use this map for generating a sequence of pseudo-random numbers, which can be used as a one time pad for encrypting messages. His work has been criticized by Wheeler [1989], who shows that when the chaotic map is implemented on digital computers, the discretized map produces cycles which are unpredictable and often short. Habutsu *et al.* [1991] suggest a cryptosystem in which the *inverse* of the one-dimensional tent map is applied N times to an initial condition representing the plain-text. In each iteration, one of the two possible preimages is chosen at random. The decryption is achieved by applying the tent map N times. For N inverse

iterations there are 2^N possible cipher-texts which encode the same plain-text. Biham [1991] pointed out that the cryptosystem could be easily broken using a chosen cipher-text type of attack, and the complexity of known plain-text type of attack is 2^{38}.

Bianco *et al.* [1991, 1994] use the logistic map to generate a sequence of floating point numbers, which is then converted to a binary sequence which is XOR-ed with the plain-text. The parameter of the logistic map together with the initial condition form part of the ciphering key. The conversion from floating point numbers to binary values is done by choosing two disjoint interval ranges (not necessarily covering the whole unit interval) representing 0 and 1. The authors claim that this irreversible process makes it impossible to recover the original values. However, it is a well-known fact from symbolic dynamics that when a chaotic orbit is converted to a sequence of symbols — sets from some partition — it may be possible to calculate the initial condition with a much better accuracy than the size of the partition sets [Fridrich, 1995a, 1995b, 1997a]. The fact that this method is based on floating point arithmetic constitutes a possible disadvantage because this makes it machine dependent, and care needs to be exercised when implementing the schemes in software. Also, while for most common chaotic maps there are numerous exact results guaranteeing aperiodic, chaotic sequences for parameters from a set of nonzero Lebesgue measure, we cannot directly transfer the results to computer approximations. It has been pointed out by Jackson [1991] and Wheeler [1989] that computer implementations of chaotic maps can exhibit surprisingly different behavior, e.g. very short cycles, depending on the particular numerical representation. While it is probably true that the typical behavior of finite approximations of chaotic systems should "converge" to that of their continuum counterparts, only very little is known at present time. The same comments apply to the work of Protopopescu [1995]. He uses m different chaotic maps, $m > 1$, which are initialized using a secret key. If the maps depend on parameters, these, too, are determined by the key. The maps are iterated using floating point arithmetic and m bytes are extracted from their floating point representations, one byte from each map. These m numbers are then combined using an XOR operation. The process is repeated to create a one-time pad which is finally XOR-ed with the plain-text.

Deffeyes [1991] describes a method in which a one time pad is generated in two-dimensional $N \times M$ blocks based on a key-dependent geometrical procedure which remotely resembles a generalized two-dimensional Baker map. The key is used to generate an initial block of bits, which is then repeatedly permuted and XOR-ed with itself. After $\log_2 (N \times M)$ iterations, a pseudo-random looking block is obtained. This block is then XOR-ed with the plain-text. The procedure is linear with respect to the plain-text and its cryptanalysis requires inverting a matrix of order $N \times M$. The author argues that since the complexity of matrix inversion scales with the third order of the matrix rank, $N \times M$, by making the block sufficiently large, the method becomes secure.

Caroll and Pecora [1990, 1991, 1992, 1993a, 1993b, 1995a, 1995b], Cuomo and Oppenheim [1994, 1995], Murali [1993], Kocarev [1992], Parlitz [1992], Papadimitriou [1992] describe encryption schemes based on synchronized chaotic circuits. These *analog* encryption schemes belong more to the field of steganography and stealth communication. Bernstein and Lieberman [1991] use chaotic circuits to build a pseudo-random number generator. Since the main focus of this paper is software encryption of digital data, such as digital imagery or electronic archives, hardware-based encryption techniques are not discussed in this paper.

Gutowicz [1993, 1994] describes an encryption scheme based on one-dimensional cellular automata. An interesting feature of his scheme is that one plain-text can correspond to many cipher-texts in a random manner. This feature makes the cipher-text slightly larger than the plain-text, however. There are no known cryptographic weaknesses of this scheme known to the author of this paper.

As discussed above, virtually all today's chaos based software encryption techniques use the one time pad. However, one time pad is not suitable for encryption of large amounts of data, such as digital imagery, electronic databases and archives. The scheme presented in this paper, is a symmetric block encryption technique based on two-dimensional chaotic maps. Possible advantages of the proposed scheme over other available encryption schemes are discussed below. A good introductory text on encryption is [Schneier, 1996].

Public key encryption schemes are not suitable for encrypting of large amounts of data and archival

because of their relatively slow performance. Also, the security of public key cryptographic schemes lies in the computational complexity of certain problems, such as factorization of large numbers or computing of the discrete logarithm problem. Advances in algorithmic techniques, number theory, and distributed computing are unpredictable and are likely to force us to reencrypt large databases and archives with a longer key to maintain a sufficient degree of security. In addition to that, the newly emerged field of quantum computing could, theoretically, make those methods totally unusable in the future [Brassard, 1988].

It is a better idea to encrypt large data files with private-key symmetric block encryption schemes. Although advances in crypt-analytic techniques and quantum computing threaten symmetric encryption schemes as well [Brassard, 1997], those schemes can provide a more stable framework with a higher degree of security and are certainly much faster than public-key schemes. Today's most common block encryption scheme, the Data Encryption Standard, DES, was designed for hardware implementation and software implementation is relatively slow. New, bulk encryption schemes, such as Blowfish or IDEA, perform much better in software and offer higher encryption rates. Depending on the number of rounds, Blowfish is 3–5 times faster than DES, and IDEA is twice as fast as DES (based on [Schneier, 1996] the throughput for DES is 35 kB/sec. on a 33 MHz 486).

This paper can be considered as an extension of the work of Pichler and Scharinger [1994, 1995] who first introduced the idea of using discretized two-dimensional chaotic maps for cryptography. In this paper, it is shown how to adapt invertible chaotic two-dimensional maps on a torus or on a square for the purpose of encryption. Although a detailed study is presented for a two-dimensional Baker map only, the techniques and ideas are applicable to other two-dimensional chaotic maps.

A chaotic map is first generalized by introducing parameters and then discretized to a finite square lattice of points which represent data items. In the rest of this paper, we refer to the square lattice as an "image" and the data item will be called a "pixel". The values of pixels will be called gray levels. It is clear that this terminology does not limit our method to digital imagery. Although the discretized map is a permutation and thus cannot be chaotic, it shares certain sensitivity and mixing properties with its continuous counterpart. It is shown that the average length of cycles and the number of different cycles correspond a typical random permutation. The discretized map is further extended to three dimensions and composed with a simple diffusion mechanism to obtain a block product encryption scheme. The performance of the cipher is studied using computer simulations. The cipher has good diffusion properties with respect to the plain-text and the key.

The main features of the encryption scheme studied in this paper are a variable key length, a relatively large block size (several kB or more), and a high encryption rate (1 Mb unoptimized C code on a 60 MHz Pentium). The cipher is based on two-dimensional chaotic maps, which are used for creating complex, key-dependent permutations. Unlike most of today's symmetric encryption schemes, which rely on complex substitution rules while somewhat neglecting the role of permutations, the new cipher is based on complex permutations composed with a relatively simple diffusion mechanism.

Section 2 describes a general five-step process of building a symmetric block cipher from a two-dimensional chaotic map. The process of building a cipher that utilizes the two-dimensional Baker map is explained in detail in Sec. 3. This cipher is further studied in the next four sections. In Sec. 4, the total number of keys for an $N \times N$ image is counted. Since the security of the cipher lies in its permutation step, which is key-dependent, it is important to study the permutations as functions of the key. For the Baker map, this is analyzed in Sec. 5 by defining a measure of similarity between permutations. It is shown that similar keys (parameters of the chaotic map) produce similar permutations and form clusters. The size of the largest cluster of similar keys is estimated. The length of the cycles forming the permutations is studied and a comparison is made with random permutations. It is shown that after several iterations, the map behaves as a typical random permutation. The similarity measures are used in Sec. 6, to study the cryptographic strength of the Baker-map-induced permutations for a direct search for the key under a known plain-text attack and a cipher-text only attack. In Sec. 7, a new type of a pseudo-random number generator is introduced and its suitability for encryption properties is evaluated. Some examples of other simple chaotic maps are given in Sec. 8

and their suitability for cryptographic purposes is discussed. Section 9 contains a brief comparison between chaotic systems and cryptosystems. The basic features of the chaos-based encryption scheme, its advantages and disadvantages and future research directions are summarized in the last Sec. 10.

2. A Method for Creating a Chaos-Based Cipher

The process of developing a chaos-based cipher can be summarized as follows. First, a chaotic map is generalized by introducing parameters into the map. Geometrical arguments are often used at this stage. Then, the map is modified so that its domain and range are both the same square lattices of points (pixels, or some other general data items). The map is extended to three dimensions so that the values of the pixels (the gray levels) can be changed. A diffusion step is introduced by composing the generalized discretized map with a simple diffusion mechanism. Let us consider square images consisting of $N \times N$ pixels with L levels of gray. The method for developing a cipher consists of the following five steps.

2.1. *Choosing the basic map*

In this step, the mathematical form of a chaotic two-dimensional map f which maps the unit square $I \times I$, where $I = [0, 1]$, onto itself in a one-to-one manner is chosen. There are a number of different chaotic maps which seem to be suitable for ciphering purposes. However, the only maps of interest are those which are simple so that the ciphering/deciphering phases can be performed quickly. The map should allow natural parametrization to create a short ciphering key with a large number of possible keys. Such maps are often described geometrically (e.g. the Baker map, the Cat map, the Standard map, etc.).

2.2. *Generalization*

In the second step, a set of parameters is introduced into the map to create a part of the ciphering key. If the basic map is described in geometric terms, the parametrization is usually straightforward. If it can be done in several different ways, the one which best suits the purpose of secure ciphering needs to be chosen. Two-dimensional chaotic maps will be characterized by a sequence of integers. Another parameter is the number of applications of the chaotic map. It is typically an integer less than 15.

2.3. *Discretization*

This step consists of modifying the generalized map to account for the fact that an image is a finite lattice of points. The domain and range of the map is changed from the unit square $I \times I$ to the lattice $\mathbf{N}_0^N \times \mathbf{N}_0^N$, where $\mathbf{N}_0^N = \{0, \ldots, N-1\}$ with N equal to the number of pixels in one row. The discretized map F takes each pixel and assigns it to some other pixel in a bijective manner (e.g. the discretized version is a permutation of pixels). The discretization must satisfy the following asymptotic property:

$$\lim_{N\to\infty} \max_{0\le i,j<N} |f(i/N,\, j/N) - F(i,\, j)| = 0\,, \quad (1)$$

where f is the continuous basic map and F is the discretized version. The formula requires the discretized map to become increasingly closer to the continuous map as the number of pixels tends to infinity.

2.4. *Extension to three dimensions*

At this point, the cipher is just a permutation cipher. By extending the map to three dimensions, the pixel values are also modified and a good substitution cipher is obtained. This can be easily achieved with a very little increase in cipher complexity. A general procedure which can be applied to any two-dimensional map is described in this paper.

2.5. *Composition with a diffusion mechanism*

Since the chaotic map extended to three dimensions is a complicated substitution cipher with no diffusion properties, it is necessary to compose the map with some simple diffusion mechanism. Linear feedback registers with carry over [Pichler & Scharinger, 1994, 1995] or other simple nonlinear mechanisms can be used to achieve this goal. The resulting cipher is a product cipher with good diffusion and confusion properties.

3. Cipher Construction for the Baker Map

3.1. *The two-dimensional Baker map*

The **Baker map**, B, is described with the following formulas

$$B(x, y) = (2x, y/2) \quad \text{when } 0 \le x < 1/2\,,$$

$$B(x, y) = (2x - 1, y/2 + 1/2) \quad \text{when } 1/2 \le x \le 1\,.$$

The map acts on the unit square as depicted in Fig. 1. The left vertical column $[0, 1/2) \times [0, 1)$ is stretched horizontally and contracted vertically into the rectangle $[0, 1) \times [0, 1/2)$, and the right vertical column $[1/2, 1) \times [0, 1)$ is similarly mapped onto $[0, 1) \times [1/2, 1)$. The Baker map is a chaotic bijection of the unit square $I \times I$ onto itself.

3.2. *Generalized Baker map*

The map can be generalized in the following way [Pichler & Scharinger, 1994, 1995]. Instead of dividing the square into two rectangles of the same size, the square is divided into k vertical rectangles $[F_{i-1}, F_i) \times [0, 1)$, $i = 1, \ldots, k$, $F_i = p_1 + \cdots + p_i$, $F_0 = 0$ such that $p_1 + \cdots + p_k = 1$ (see Fig. 2). The lower right corner of the ith rectangle is located at $F_i = p_1 + \cdots + p_1$. The generalized Baker map stretches each rectangle horizontally by the factor of $1/p_i$. At the same time, the rectangle is contracted vertically by the factor of p_i. Finally, all rectangles are stacked on top of each other as in Fig. 2. Formally,

$$B(x, y) = \left(\frac{1}{p_i}(x - F_i),\, p_i y + F_i\right)$$

for

$$(x, y) \in [F_i, F_i + p_i) \times [0, 1)\,,$$

It is convenient to denote the Baker map and its generalized version as $B_{(1/2,\,1/2)}$ and $B_{(p_i,\ldots,p_k)}$, respectively. The generalized map inherits all important properties of the Baker map such as sensitivity to initial conditions and parameters, mixing, and bijectiveness.

3.3. *Discretized Baker map*

3.3.1. *Version A*

Since an image is defined on a lattice of finitely many points (pixels), a correspondingly discretized

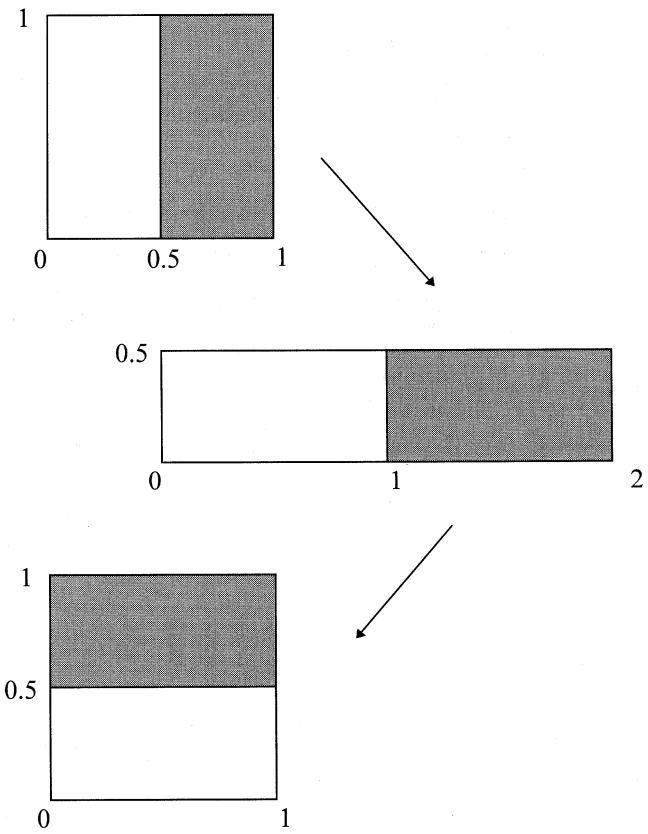


Fig. 1. Baker map.

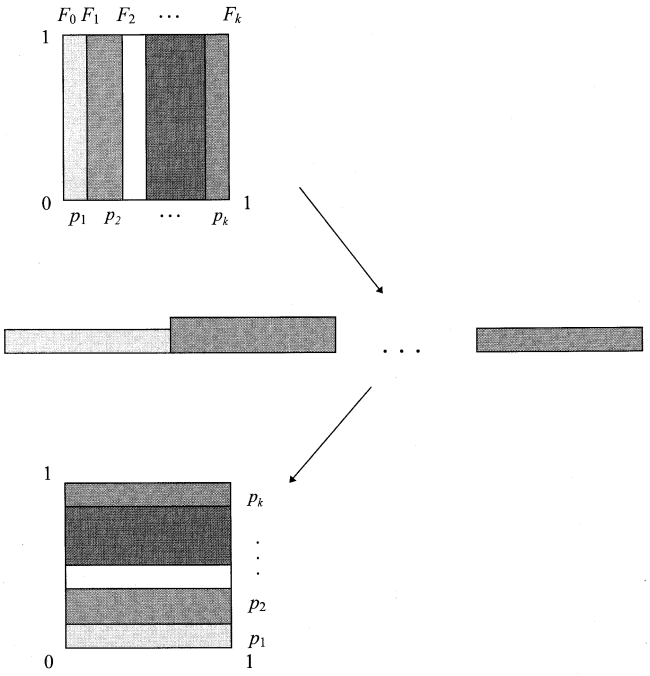


Fig. 2. Generalized Baker map.

form of the basic map needs to be derived. In particular, the discretized map is required to assign a pixel to another pixel in a bijective manner. Since the discretized map is desired to inherit the properties of the continuous basic map, the discretized map should become increasingly close to the basic map as the number of pixels tends to infinity. This requirement is expressed mathematically with Eq. (1). Following the approach suggested by Pichler and Scharinger [1994, 1995], the discretized generalized Baker map will be denoted $B_{(n_1,\ldots,n_k)}$, where the sequence of k integers, $n_1,\ldots,n_k$, is chosen such that each integer n_i divides N, and $n_1+\cdots+n_k=N$. Denoting $N_i=n_1+\cdots+n_i$, the pixel $(r,\,s)$, with $N_i\le r<N_i+n_i$, and $0\le s<N$ is mapped to

$$B_{(n_1,\ldots,n_k)}(r,\,s)=\left(\frac{N}{n_i}(r-N_i)+s \mod \frac{N}{n_i},\right.$$
$$\left.\frac{n_i}{N}\left(s-s \mod \frac{N}{n_i}\right)+N_i\right). \tag{2}$$

This formula is based on the following geometrical considerations. An $N\times N$ square is divided into vertical rectangles of height N and width n_i. Following the action of the generalized Baker map, these vertical rectangles should be stretched in the horizontal direction and contracted in the vertical direction to obtain a horizontal $n_i\times N$ rectangle. To achieve this for the discretized map, each vertical rectangle $N\times n_i$ is divided into n_i boxes $N/n_i\times n_i$ containing exactly N points (see Fig. 3, Version A). Each of these boxes is mapped to a row of pixels. Since there are n_i boxes, a horizontal rectangle $n_i\times N$ is obtained, as required. Now, how the pixels in each box are mapped to a row of pixels need to be specified. Since the original Baker map is continuous on each box, the only plausible discretization is to map the box column by column. An example for $N=16$, $n_i=2$ is shown below. The rectangle $N/n_i\times n_i=16/2\times 2=8\times 2$ is mapped to a row of 16 pixels as follows:

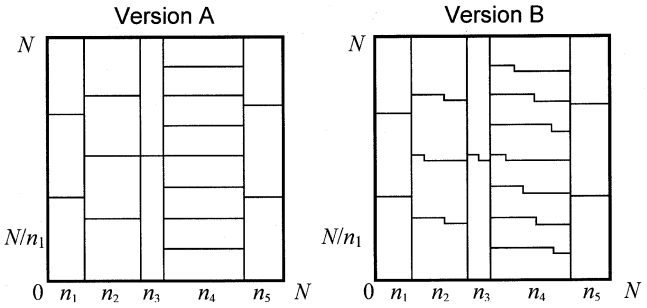


Fig. 3. Discretized versions of the Baker map.

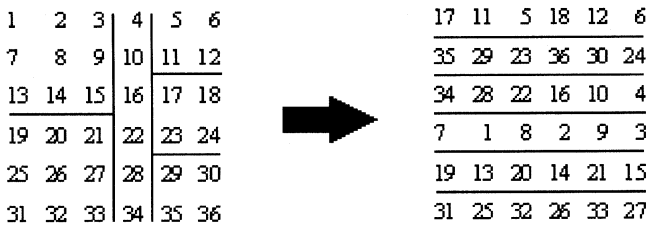
8	16
7	15
6	14
5	13
4	12
3	11
2	10
1	9

↓

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Examples of permutations for a complete 6-pixel image and an 8-pixel image are worked out in detail in Figs. 4 and 5. For the 6-pixel image, a 3-1-2 division is used, while in the 8-pixel case the division is 2-4-2.

Equation (2) is a symbolic, mathematical description of this geometric procedure. It is possible to justify the formula via symbolic dynamic. The action of the generalized Baker map can be



1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

17	11	5	18	12	6
35	29	23	36	30	24
34	28	22	16	10	4
7	1	8	2	9	3
19	13	20	14	21	15
31	25	32	26	33	27

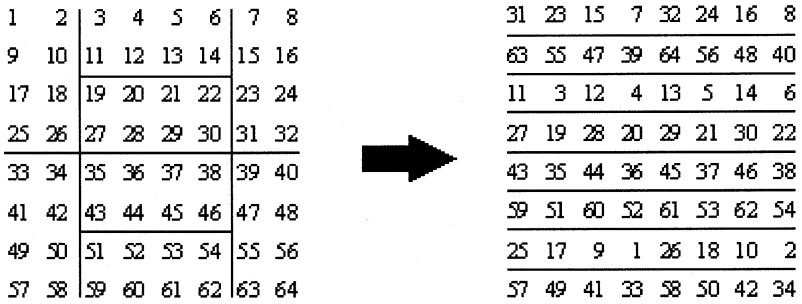
Fig. 4. The permutation induced by the discretized Baker map for a 6-pixel image (division 3, 1, 2).

described with Bernoulli shifts on double-infinite sequences. Similarly, it can be shown that the action of the discretized map defined by Eq. (2) can be represented by Bernoulli shifts in finite Abelian groups [Pichler & Scharinger, 1994, 1995]. This representation enables an elegant description of the dynamic in a symbolic form. This form could also be utilized for an efficient software and hardware implementation.

The application of the Baker map to a gray scale test image 472×472 shown in Fig. 6 produces encrypted images as demonstrated with Figs. 7 and 8. The ciphering key was randomly generated and consists of the following sequence of 17 divisors of 472:

$$(8\ 8\ 8\ 59\ 59\ 4\ 4\ 118\ 118\ 4\ 2\ 4\ 4\ 59\ 8\ 4\ 1) \quad (3)$$

Figure 6 shows the original image, and Figs. 7 and 8 show the results of applying the generalized discretized Baker map once and nine times, respectively.



1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

31	23	15	7	32	24	16	8
63	55	47	39	64	56	48	40
11	3	12	4	13	5	14	6
27	19	28	20	29	21	30	22
43	35	44	36	45	37	46	38
59	51	60	52	61	53	62	54
25	17	9	1	26	18	10	2
57	49	41	33	58	50	42	34

Fig. 5. The permutation induced by the discretized Baker map for an 8-pixel image (division 2, 4, 2).

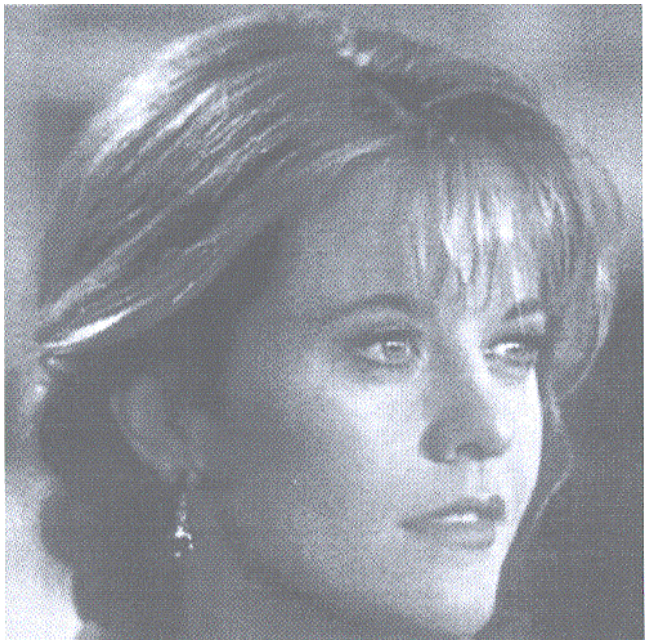


Fig. 6. The test image 472×472 pixels with 256 gray levels.

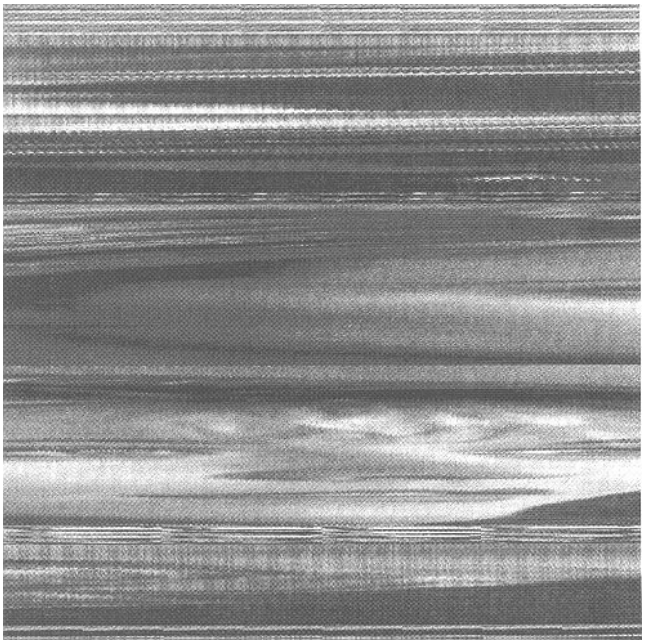


Fig. 7. The test image after applying the Baker map once.

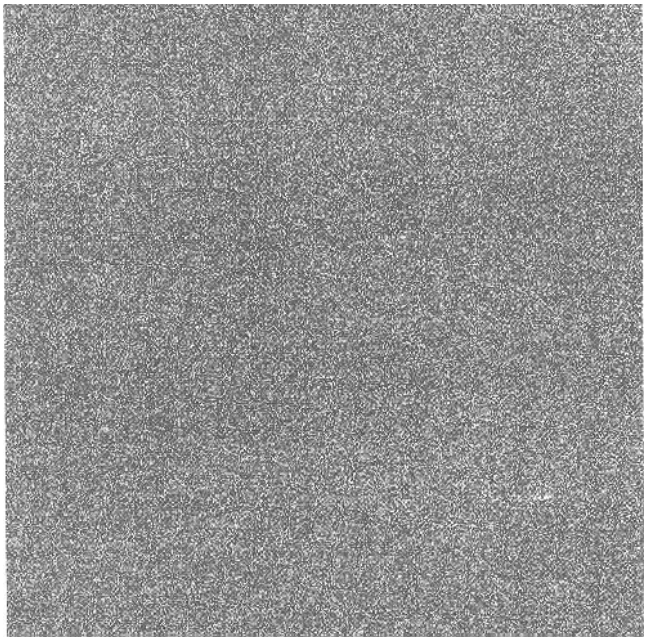


Fig. 8. The test image after applying the Baker map nine times.

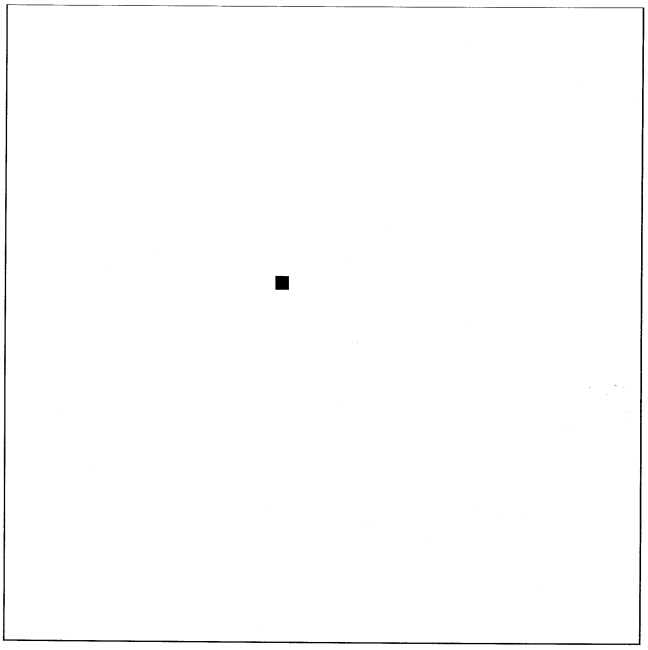


Fig. 9. An image consisting of a 10×10 black square on a white background.

The fact that spatially localized information in the original image becomes nonlocal and uncorrelated in the encrypted image can be illustrated with the following example. The original image consists of a 10×10 black square on a white background of a 472×472 image (see Fig. 9). A randomly generated ciphering key (3) was used to iterate the discretized generalized Baker map nine times. The result is shown in Fig. 10. The black pixels are scattered all over the image in an apparently random manner.

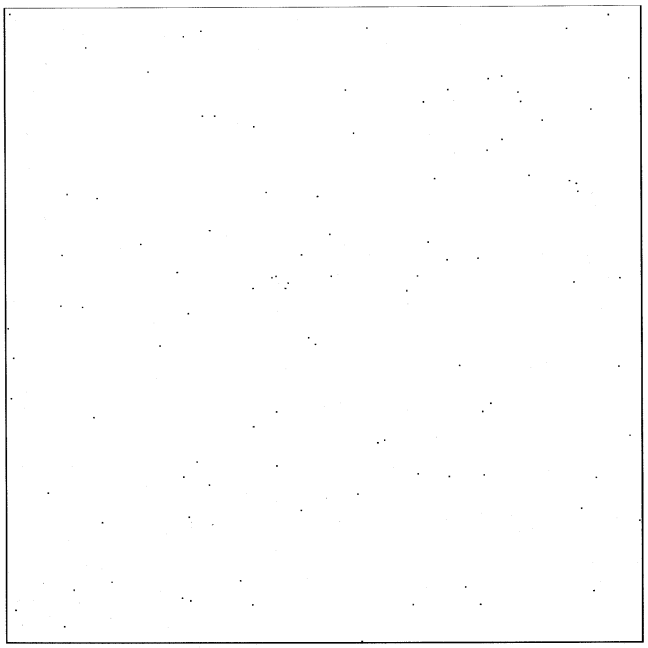


Fig. 10. Figure 9 encrypted after nine iterations.

3.3.2. *Version B*

It is possible to generalize the geometric procedure to an arbitrary combination of integers (e.g. not only divisors of N are considered) $n_1, \ldots, n_k$ which add up to N. This is important for several reasons:

- By constraining n_i to divisors of N, certain values of N may produce relatively small number of ciphering keys (for example, when N has only small number of divisors). When the encryption method is applied to the raw image data, some images would have to be slightly enlarged to the nearest integer N with a large number of divisors.
- Even though the cipher is used as a block cipher with a fixed block size which could be chosen at our leisure, the constraint of n_i being divisors of N can be rather limiting.
- The randomness properties of the permutations are much better for keys with general n_i than for keys consisting of divisors only.

In order to generalize the procedure to an arbitrary sequence of numbers $n_1, \ldots, n_k$, similar

geometric arguments as before are used. The image is again divided into vertical rectangles $N \times n_i$. Each rectangle is divided into n_i boxes containing exactly N points. Now these boxes do not necessarily have to have a rectangular shape (see Fig. 3, Version B). The top and bottom row of each box may exhibit a one-pixel "step". However, it is still possible to map the pixels column by column to a row of pixels. The procedure is illustrated with $N = 16$, $n_i = 6$ below. The first 4 columns will be by one pixel longer than

3	6	9	12		
2	5	8	11	14	16
1	4	7	10	13	15

↓

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

the remaining $n_i - 4$ columns. In particular, the first 4 columns will consist of $\lceil N/n_i \rceil$ pixels, where $\lceil x \rceil$ denotes the smallest integer greater or equal to x. An example of a permutation for an 8-pixel image is worked out in detail in Fig. 11. The only small inconvenience is that there is no simple formula similar to Eq. (2). Also sacrificed is the advantage of having a Bernoulli shift-based description of the discretized map. However, as discussed in [Fridrich, 1997b] on fast implementations of the ciphering technique on sequential computers, the implementation of the encryption algorithm for an arbitrary combination of n_i is no more complicated than in the previous case.

3.4. *Extension to rectangular images*

It is always recommended that an image be compressed before it is encrypted. From this point of view, it does not make much sense to further generalize the discretized Baker map to rectangular images since one will have to pad the compressed image to the block size anyway. Nevertheless, the continuous Baker map can be readily applied to both squares and rectangles without any changes. It is interesting to attempt the same feat in the discretized case. The problem with the discretized version is that given an $M \times N$ image with $M \neq N$, the number of pixels in each vertical rectangle, $n_i N$, may not be a multiple of M. In order to keep the number of pixels in each vertical rectangle at some multiple of M, the rectangles are modified to allow a one-pixel step in the vertical sides of each rectangle. This is a similar modification as in the description of Version B. Consequently, some boxes may have a unit step in pixels not only at the top or the bottom but also at the sides. Of course, when $n_i N$ is divisible by M, there will be no steps in the vertical sides of the rectangle. This slight modification of the Baker map preserves the geometric characteristics of the original continuous Baker map. The discretization is also consistent with Eq. (1). As explained in [Fridrich, 1997b] since it is possible

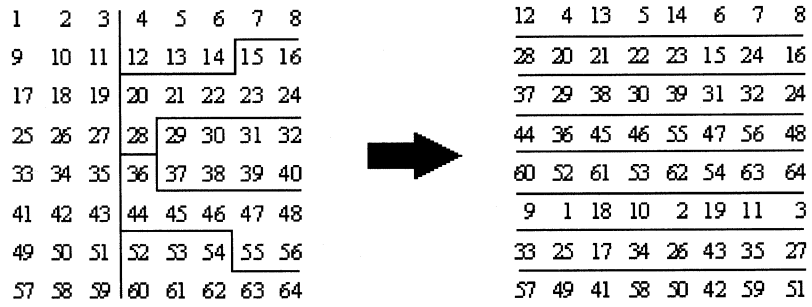


Fig. 11. Permutation induced by the discretized Baker map for an 8-pixel image when the integers n_i are not divisors of 8 (division 3, 5).

to implement the scheme using transfer matrices, there is no additional complication of the practical implementation.

3.5. *Extension to three dimensions*

The method described in this section is general and can be applied not only to the Baker map, but to any two-dimensional discretized map. It can be used to modify *any* one-to-one two-dimensional mapping to a three-dimensional mapping which acts both on the pixels and on their gray levels. The extension can be achieved by a slight modification of the chaotic map and it significantly contributes to the security of the whole cipher. The resulting substitution cipher can create a random looking image with uniform histogram in only a few iterations.

Consider a $N \times N$ square image with L gray levels. Let B be any discretized two-dimensional chaotic map. Let g_{ij} denote the gray level of the pixel (i, j),

$$g_{ij} \in \{0, \ldots, L-1\} = \mathbf{N}_0^L\,. \quad \text{A } \textit{map}$$

$$h : \mathbf{N}_0^N \times \mathbf{N}_0^N \times \mathbf{N}_0^L \to \mathbf{N}_0^L$$

needs to be found such that the pixel (i, j) with gray level g_{ij} is mapped to $B(i, j)$ with a gray level $h(i, j, g_{ij})$. The three-dimensional map

$$B3 : \mathbf{N}_0^N \times \mathbf{N}_0^N \times \mathbf{N}_0^L \to \mathbf{N}_0^N \times \mathbf{N}_0^N \times \mathbf{N}_0^L$$

should be invertible to make deciphering possible. This means that $B3(i, j, g) \neq B3(i, j, g')$ for each $(i, j) \in \mathbf{N}_0^N \times \mathbf{N}_0^N$ and $g, g' \in \mathbf{N}_0^L$. This is possible if and only if h is one-to-one for each i, j. This requirement is not restrictive at all and enables one to construct a large variety of gray level permutations. For example,

$$h(i, j, g_{ij}) = g_{ij} + \overline{h}(i, j) \mod L\,, \qquad (4)$$

where $\overline{h}$ is *any* (possibly not one-to-one) function of i and j, produces an acceptable map h. In this case, h can be interpreted as a simple shift cipher with the shift size $\overline{h}(i, j)$ dependent on the position of the pixel i, j. The shift size $\overline{h}(i, j)$ could be computed quickly using some bit operations on i and j, or it could be stored in a look up table. For a fast encryption, h should be chosen so that it can be performed quickly or, preferably, hard-wired into an encrypting hardware, while maintaining the security of the cipher. Future work includes the investigation into the performance of the encryption scheme for various choices of h obtained using XOR operation with i and/or j.

To explain how the 3D chaotic map works, let us represent the permutation induced by the discretized two-dimensional chaotic map B using two transfer matrices $t1$ and $t2$. The discretized map B transforms a pixel with coordinates (i, j) to a new position $(t1(i, j), t2(i, j))$. Storing the pixel values of the original image in a two-dimensional integer array $pixel[0 \ldots N-1][0 \ldots N-1]$, the new, enciphered image is stored in the integer array $new\_pixel[0 \ldots N-1][0 \ldots N-1]$. The following code fraction in C explains how the 3D map transforms the pixel values.

Code fraction 1.

```
for(i = 0; i < N; i++)
{
        for(j = 0; j < N; j++)
        {
                new_pixel[t1[i][j]] [t2[i][j]]
                        = (pixel[i][j] + h(i, j))%L;
        }
}
```

The deciphering process can be implemented using transfer matrices in a similar way. In fact, having computed the transfer matrices $t1$ and $t2$, the inverse transfer matrices $t1^{-1}$, $t2^{-2}$ can be obtained directly from $t1$, $t2$ without having to perform any integer arithmetic operations:

$$t1^{-1}(t1(r, s), t2(r, s)) = r$$

$$t2^{-1}(t1(r, s), t2(r, s)) = s\,.$$

The three-dimensional chaotic map leads to a substitution cipher which can create a random image with uniform histogram in a few iterations. The randomness properties of the image enciphered with a three-dimensional Baker map are studied in Sec. 7. The result of enciphering the test image from Fig. 6 with the key (3) using $\overline{h}(i, j) = i \,.\, j$ is shown in Figs. 12 and 13. Figure 12 shows the enciphered image after one iteration. The result after 9 iterations is shown in Fig. 13. *Even one iteration of the 3D chaotic Baker map makes the histogram uniform, although the histogram of the original image was highly nonuniform.*

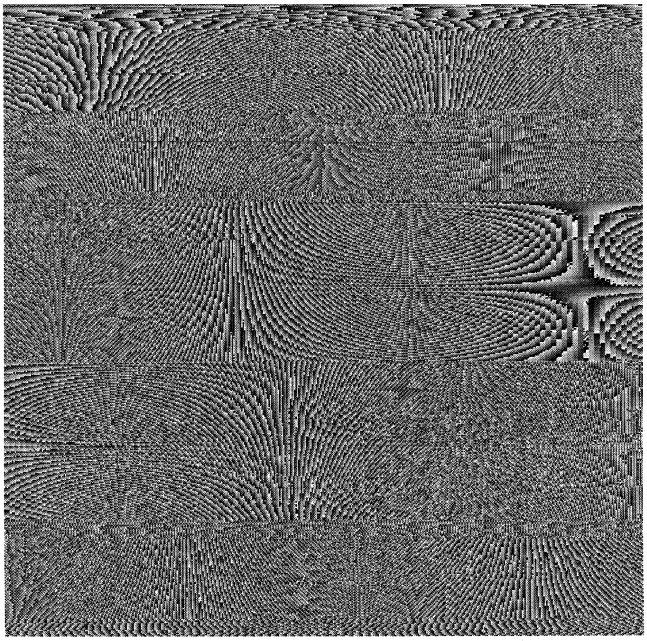


Fig. 12. The test image after one iteration of a 3D chaotic map. The histogram of the image is uniform.

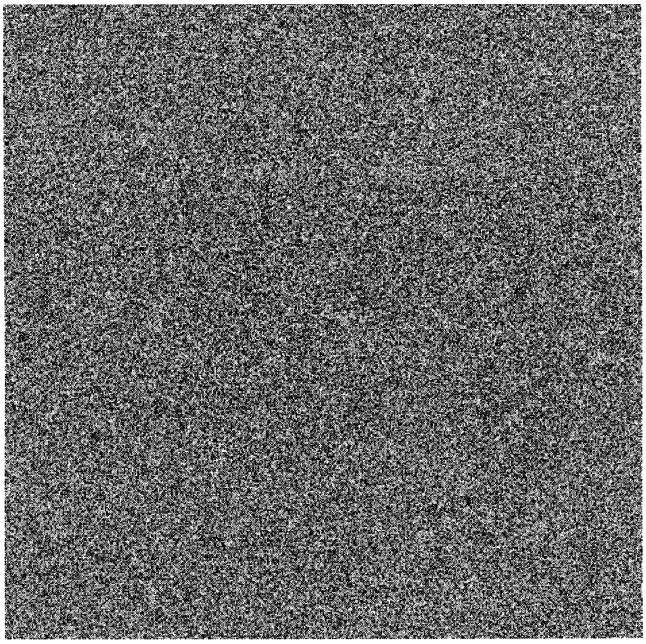


Fig. 13. The test image after nine iterations of a 3D chaotic map. The histogram of the image is uniform.

Figures 14 and 15 show the result of enciphering a black square using one and two iterations, respectively. As can be seen from the figures, starting with a black square only two iterations of the 3D chaotic map create an image with a uniform histogram!

We note that our encryption scheme in the current stage has zero diffusion, which is, of course,

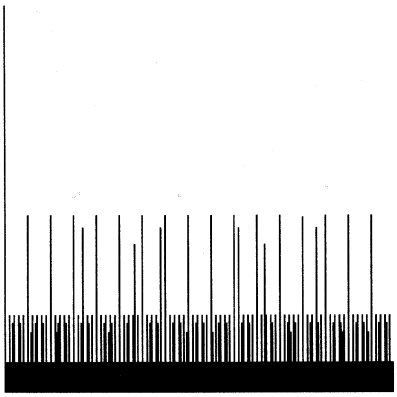


Fig. 14. Histogram of a black square after one iteration of the Baker map with gray-level mixing function $h(i, j) = i, j$.

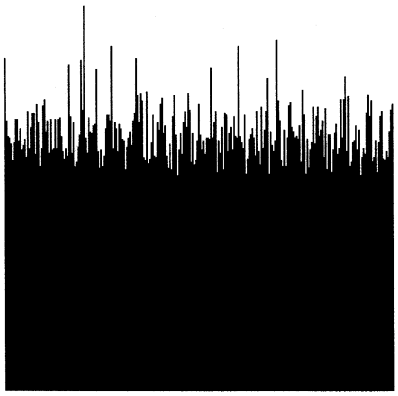


Fig. 15. Histogram of a black square after two iterations with gray-level mixing function $h(i, j) = i, j$.

undesirable from the security point of view. For example, the scheme could be broken with a chosen plain-text attack with only $N \times M$ chosen plain-texts. We introduce diffusion in the next section.

3.6. *Composing with a diffusion mechanism*

The diffusion methods described in this section are not constrained to the Baker map and can be applied to any block-substitution cipher including the 3D Baker map from the previous section. Any three-dimensional discretized chaotic map has a robustness property in the sense that flipping one pixel in the encrypted image influences one pixel in the decrypted image. The error does not diffuse through the image. This error robustness is potentially dangerous and makes the method vulnerable to a chosen plain-text type of attack. If an eavesdropper can choose the images to be encrypted, he could encrypt two images which differ in one pixel,

and then compare the encrypted versions. This way, the eavesdropper can learn to which pixel is the modified pixel mapped to. Repeating this procedure for each pixel in the image can reveal the permutation of pixels. It is then possible to reconstruct the parameters of the chaotic map. The only thing left would be the crypt-analysis of function h. Clearly, a diffusion mechanism must be present in any secure cipher [Shannon, 1949].

We suggest inserting a diffusion step into the encryption scheme after the permutation and gray-level mixing. Therefore, one complete encryption step consists of (i) one permutation with simultaneous gray-level mixing, (ii) one diffusion step during which information is spread over the image. This way, an error will diffuse to neighboring pixels each time the chaotic map is iterated.

There are many ways one can implement a diffusion step into the scheme. Pichler and Scharinger [1994, 1995] take a product of the Bernoulli cipher with a maximal length linear feedback shift register with carry over. Their method also generates uniform histograms and, at the same time, achieves complete diffusion with respect to plain-text. We have experimented with two different methods.

Method 1. Assuming the dimensions of the image are even numbers, one can divide the image into a regular tessellation of 2×2 squares. The new gray levels of each pixel within each 2×2 group depend on all four gray levels in the group. This way, after one iteration, the number of influenced pixels approximately quadruples. By adjusting the number of iterations, one can achieve complete diffusion, or partial diffusion which will enable partial recovery of the image, while keeping the chosen plain-text attacks unfeasible. It has been observed that this modification can not only achieve complete diffusion with respect to changes in pixels, but also complete diffusion with respect to changes to the ciphering key. Modifying the ciphering key by one bit or modifying the plain-text by one bit changes completely the cipher-text. This was tested by comparing the two cipher-texts and evaluating the correlation between these two. The results of many computer tests showed that only statistical correlation corresponding to two random images occurred.

Method 2. In the second method, the diffusion was obtained by scanning the image by rows (starting, for example, in the upper left corner), and changing the gray levels according to the formula

$$v_k^* = v_k + G(v_{k-1}^*) \mod L$$
$$v_{-1} = \text{initial value}\,,$$

where v_k denotes the gray levels arranged in a one-dimensional vector obtained by scanning the image by rows. The function G is some arbitrary function of the gray level, and it was chosen as a fixed random permutation implemented using a lookup table in our tests. The corresponding inverse step used in the deciphering procedure is

$$v_k = v_k^* - G(v_{k-1}^*) \mod L\,.$$

The implementation of the diffusion Method 2 is explained below using a code fraction written in C.

Code fraction 2.

```
previous = initial_value;
for(i = 0; i < N; i++)
{
        for(j = 0; j < N; j++)
        {
                pixel[i][j] = pixel[i][j] + G(previous);
                previous = pixel[i][j];
        }
}
```

This procedure achieves complete mixing very quickly and also produces a complete diffusion with respect to the plain-text. Both methods also make our ciphering technique sensitive to the key in the sense that small changes to the key produce statistically independent cipher-texts. In other words, there are no clusters of similar keys, which would weaken the scheme. The complete encryption scheme is schematically explained in Fig. 16. The enciphering process consists of several iterations of the 3D substitution step composed with the diffusion step. Detailed study of the cipher with diffusion will be the subject of further research.

4. The Number of Ciphering Keys for the Baker Map

The ciphering key is formed by the parameters of the chaotic map, the number of applications of the

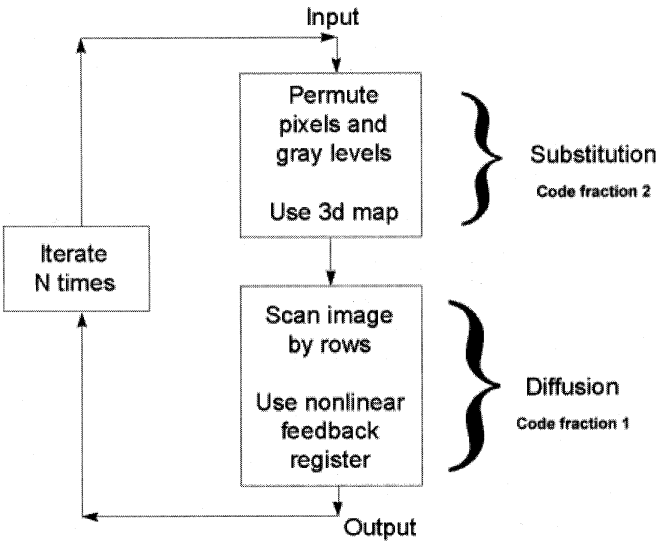


Fig. 16. A diagram of the product encryption scheme. One iteration is composed of one permutation step and one diffusion step.

map, the parameters of the gray-level transformation h, and parameters of the diffusion part. For both versions of the Baker map, the key K_B is

$$K_B = \{C, n_1, \ldots, n_k, p_1, \ldots, p_h, D\},$$

where C is the number of iterations of the Baker map, $n_1, \ldots, n_k$ are the parameters of the Baker map, $p_1, \ldots, p_h$ are the parameters necessary to describe h, and D is the set of parameters needed for the diffusion step. Experiments with real images and the cipher security analysis in Secs. 5 and 6 suggest that $C < 15$ produces safe encryption.

The number of possible keys grows very rapidly with the number of pixels in the image. Since the map (4) and the diffusion contribute the same way for any chaos-based cipher, we exclude their contribution to the number of ciphering keys and calculate the number of ciphering keys due to parameters of the chaotic maps. So, let us calculate the number of ciphering keys produced only by $n_1, \ldots, n_k$. To estimate the total number of different ciphering keys, consider Version A of the Baker map. The task is to estimate how many times an integer N can be written as an ordered sum of its divisors. The number of divisors, k, ranges from some small number, such as 2 if N is even, to N for a key consisting of N ones. The total number of ciphering keys, $K(N)$, depends on N and on how many different divisors exist for N. Trivially, when N is a prime number, $K(N) = 1$. For $N = 2^m$, $K(2^{m+1}) \approx [K(2^m)]^2$. For $N = 64$, 128, 256, 512, $K(N) \approx 10^{15}, 10^{31}, 10^{63}, 10^{126}$, respectively. The number $K(N)$ tends to be higher for N with a large number of different divisors, such as $N = 30, 60, 120, 240$. The table below shows $K(N)$ as a function of N for selected values of N.

N	$K(N)$	N	$K(N)$	N	$K(N)$
4	5	27	26425	50	2.6e11
6	24	28	5e6	51	1.8e8
8	55	30	1.5e8	52	3.1e12
9	19	32	4.7e7	54	8.0e13
10	128	33	2.2e5	55	6.6e6
12	1627	34	9.2e6	56	6.8e13
14	741	35	51885	57	1.8e9
15	449	36	1.5e10	58	9.5e11
16	5271	38	6.3e7	60	3.8e17
18	45315	39	2.0e6	62	6.5e12
20	83343	40	1.4e10	63	2.5e11
21	3320	42	1.8e11	64	3.8e15
22	29966	44	3.5e10	128	e31
24	5.1e6	45	4.7e8	256	e63
25	571	46	2.9e9	512	e126
26	2.0e5	48	4.8e13	1024	e255

In the table, the following notation is used: 2.0e5 means 2.0×10^5, etc. For Version B of the Baker map, it is easy to see that the total number of ciphering keys $K(N) = 2^{N-1}$ for all N. This number monotonically increases with N and is consistently larger by several orders of magnitude compared to Version A. The number of keys, $K(N, m)$, of length equal to m is

$$K(N, m) = \binom{N}{m}.$$

5. Structure of the Permutations Induced by the Baker Map

Each permutation can be uniquely represented as a collection of cycles. The structure of the permutations induced by the generalized discretized Baker map is studied. The average length of a cycle, and the average number of cycles were used to compare the permutations with a typical random permutation.

The average length of a cycle is defined as the expected value of the number of iterations of the map necessary to bring a randomly chosen pixel back to its original position. Given N^2 pixels, if a permutation consists of k cycles of length $c_1, \ldots, c_k$, such that $c_1 + \cdots + c_k = N^2$, the expected value, $C(N)$, of the cycle for a randomly chosen pixel is

$$C(N) = \frac{\sum_{i=1}^{k} c^2 i}{N^2} .$$

The average number of cycles, $\mathrm{cyc}(N)$, is defined as the expected value of k. It can be shown [Feller, 1957], that for a random permutation,

$$C(N) = \frac{N^2 + 1}{2} ,$$

$$\mathrm{cyc}(N) = 2\ln(N) \pm 2\ln(N) .$$

A set of computer experiments with 1000 randomly chosen keys of length between 10 and 15 for a square image 472×472 pixels gives $C(N) = 111,115.7$, $\mathrm{cyc}(N) = 33.5$. These values should be compared to random permutations with $C(N) = 111,392.5$, $\mathrm{cyc}(N) = 24.6$. Since the standard deviation for $\mathrm{cyc}(N)$ is 24.6, we can conclude that the permutations induced by the Baker map behave as typical random permutations.

Another aspect of the permutations important for the security of the cipher is how the permutations depend on the key $n_1, \ldots, n_k$. Do similar keys generate similar permutations? How can one measure similarity for permutations? In order to answer these questions, we studied a *simplified version of our encryption scheme with gray-level mixing and diffusion removed* and studied this weakened permutation cipher.

Figures 17–20 show the results of deciphering an image using wrong ciphering keys. The original image shown in Fig. 6 consists of 472×472 pixels with 256 gray levels. When the key (3)

(8 8 8 59 59 4 4 118 118 4 2 4 4 59 8 4 1)

was changed by replacing the parameter 4 on the sixteenth place by two parameters 2,

(8 8 8 59 59 4 4 118 118 4 2 4 4 59 8 2 2 1)

Fig. 17. Test image enciphered with (8 8 8 59 59 4 4 118 118 4 2 4 4 59 8 4 1) and deciphered with (8 8 8 59 59 4 4 118 118 4 2 4 4 59 4 4 2 2 1).

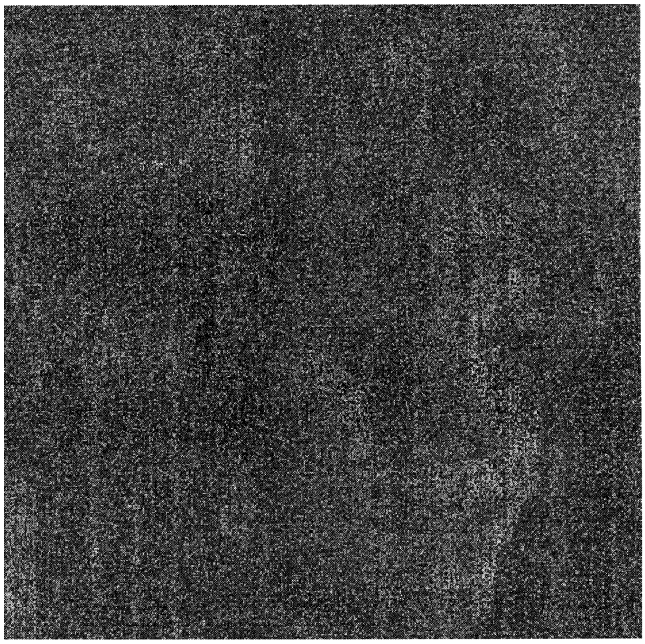


Fig. 18. Test image enciphered with (8 8 8 59 59 4 4 118 118 4 2 4 4 59 8 4 1) and deciphered with (8 8 8 118 4 4 118 118 4 2 4 4 59 8 4 1).

the deciphered image shown in Fig. 17 is clearly recognizable, although it does contain some "noise". While replacing the two parameters 59 on the fourth and the fifth place by their sum results in a much more noisy deciphered image (Fig. 18), one can

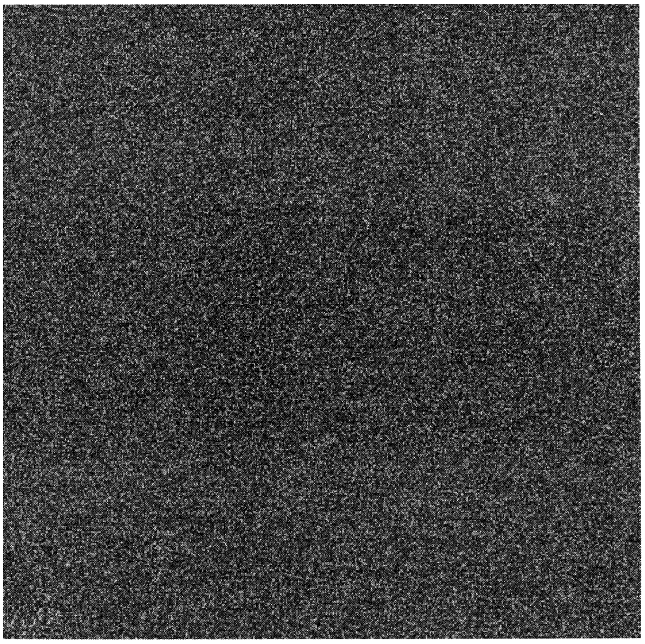


Fig. 19. Test image enciphered with (8 8 8 59 59 4 4 118 118 4 2 4 4 59 8 4 1) and deciphered with (8 8 8 59 59 4 118 118 4 4 2 4 4 59 8 4 1).

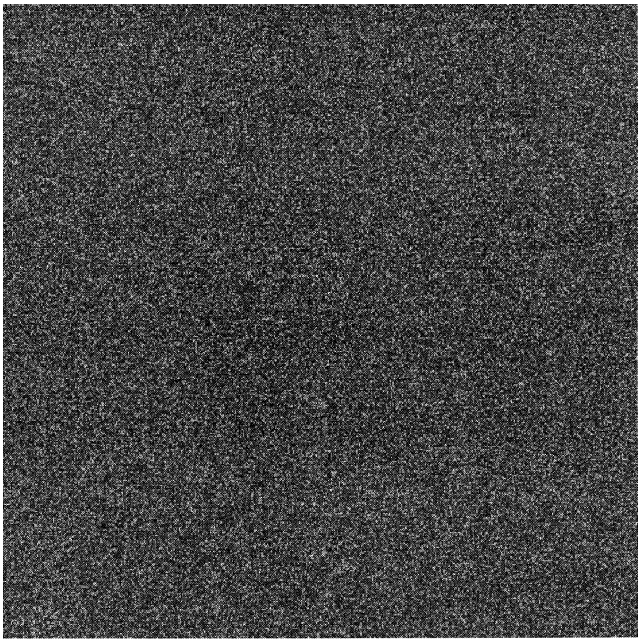


Fig. 20. Test image enciphered with (8 8 8 59 59 4 4 118 118 4 2 4 4 59 8 4 1) and deciphered with (8 8 8 59 59 4 4 59 59 59 59 4 2 4 4 59 8 4 1).

however, still find some traces of the original image (e.g. the edge on the right side of the face). Figure 19 shows the result of deciphering using a key in which the parameter 4 on the seventh place in (3) was moved after the second parameter 118. Clearly, the attempt to decipher the image has been unsuccessful. Figure 20 was obtained using a modification of the key (3) in which the two parameters 118 were replaced by four parameters 59. Again, the deciphered image does not bear any resemblance to the original.

The keys in which one parameter is replaced by a sequence of smaller parameters, or in which small parameters are merged into a larger one, are "close to each other". This observation suggests that each ciphering key is surrounded by a cluster of keys which are close to it. Below, we estimate the size of these clusters.

The similarity between two keys, K_1 and K_2, should be measured by the difference in the performance of their ciphering maps B_1 and B_2 on a typical image.

Let D_k denote the number of pixels (i, j) in the original image I which are mapped to different pixels after k applications of the maps B_1 and B_2, i.e. $B_1^k(i, j) \neq B_2^k(i, j)$, Clearly, when D_k becomes comparable to the total number of pixels in the image, N^2, the images $B_1^k(I)$ and $B_2^k(I)$ will bear no resemblance. An important issue from the point of view of the security of the cipher is how fast D_k increases with k, and how D_k depends on the similarity between the ciphering keys K_1 and K_2. Both questions are answered below.

It will be useful to represent ciphering keys as disjoint unions of subintervals of the interval $[0, N)$. Let

$$K_1 = (n_{i(1)}, n_{i(2)}, \ldots, n_{i(k)})$$

denote a ciphering key with $n_{i(1)}, \ldots, n_{i(k)}$ being the parameters, such that $n_{i(1)} + \cdots + n_{i(k)} = N$. Denoting $N_j = n_{i(1)} + \cdots + n_{i(j)}$, $N_0 = 0$, $N_k = N$, the key K_1 can be represented by a union U_1 of disjoint subintervals of $[0, N)$

$$U_1 = [0, N_1^{(1)}) \cup [N_1^{(1)}, N_2^{(1)}) \cup \cdots \cup [N_{k-1}^{(1)}, N_k^{(1)}) .$$

The similarity between two keys, K_1 and K_2, will be measured by the total length of intervals from $U_1 \cap U_2$. Denoting

$$U_2 = [0, N_1^{(2)}) \cup [N_1^{(2)}, N_2^{(2)}) \cup \cdots \cup [N_{m-1}^{(2)}, N_m^{(2)}) ,$$

assume that the two keys are the same with the exception of one interval in U_1, $[N_s^{(1)}, N_{s+1}^{(1)})$, which

corresponds to a union of several smaller consecutive intervals in U_2. This can be best visualized graphically as:

$$U_1 = |.....|.....|...|.....|.....|.....|.....|.....|.....|$$
$$U_2 = |.....|.....|...|.....|.....|.....|....|.....|.....|.|.....|.....|$$

In this particular example, the seventh interval in U_1 has been replaced by a union of four smaller intervals in U_2. From the graphical interpretation of the Baker map it is easy to see that the vertical rectangle with pixels with indices (i, j), $N_s^{(1)} \le i < N_{s+1}^{(1)}$, $0 \le j < N$, will be placed differently under application of B_1 and B_2. Therefore, $D_1 = (N_{s+1}^{(1)} - N_s^{(1)}) \times N = n_{i(s)} \times N$. When applying the Baker map the second time, the number of newly misplaced pixels increases by $n_{i(s)} \times N$ minus the relative portion of already misplaced pixels from the previous step

$$\frac{n_{i(s)}}{N} \times n_{i(s)} \times N = n_{i(s)}^2 .$$

This argument can be repeated to obtain an *approximate* recurrent formula

$$D_k = D_{k-1} + n_{i(s)} \times N - \frac{n_{i(s)}}{N} D_{k-1} ,$$
$$k = 1, 2, \ldots .$$

By dividing the last expression by the total number of pixels, N^2, the following formula for the relative count of misplaced pixels is obtained:

$$r_k = D_k / N^2$$
$$r_k = (1 - \varepsilon) r_{k-1} + \varepsilon$$

where $\varepsilon = n_{i(s)}/N$, $0 \le \varepsilon \le 1$. It is easy to solve this recurrent expression for r_k to obtain

$$r_k = 1 - (1 - \varepsilon)^{k-1}$$
$$D_k = r_k N^2 . \qquad (5)$$

Since Eq. (5) is the basis of further cipher security considerations, it is important to study its validity for real images. In order to test the formula, an image was encrypted using two different keys with an increasing number of iterations. The images were compared and the pixels with different gray levels were counted. This number, which is a function of the number of iterations k, is denoted as Dg_k. Finally, Dg_k was compared to D_k based on Eq. (5).

Set #1.
In the first set of experiments, it is assumed that each pixel in the image has a different gray level. The ciphering keys used in simulations are:

$$K_1 = (8\,8\,8\,59\,59\,4\,4\,118\,118\,4\,2\,4\,2\,2\,59\,8\,4\,1) ,$$
$$K_2 = (8\,8\,8\,59\,59\,4\,4\,118\,118\,4\,2\,4\,2\,2\,59\,4\,4\,4\,1) ,$$
$$K_3 = (8\,8\,8\,59\,59\,4\,4\,118\,118\,4\,2\,4\,2\,1\,1\,59\,8\,4\,1) .$$

The 16th interval of K_1 of length 8 was replaced by two subintervals of half the length in K_2 and the 14th interval in K_1 was replaced by two subintervals of length 1 in K_3. The following table summarizes the number of differently placed pixels, Dg_k, under iterations with K_1 compared to iterations with K_2 and K_3.

	K_1K_2		K_1K_3	
k	Dg_k/N^2	$r_k = D_k/N^2$	Dg_k/N^2	$r_k = D_k/N^2$
1	1.695e-2	1.695e-2	4.237e-3	4.237e-3
2	3.361e-2	3.361e-2	8.457e-3	8.457e-3
3	4.960e-2	4.999e-2	1.252e-2	1.266e-2
5	8.020e-2	8.192e-2	2.054e-2	2.100e-2
10	0.1481	0.1571	3.871e-2	4.157e-2
20	0.2532	0.2896	6.842e-2	8.142e-2
40	0.3817	0.4953	0.1108	0.1562
80	0.4906	0.7453	0.1537	0.2880
160	0.5354	0.9351	0.1841	0.4931
320	0.5405	0.9958	0.1915	0.7430

As can be seen from the table, the accuracy of Eq. (5) is better than 7% when the number of iterations is less than 10. The accuracy decreases to about 15% when the number of iterations reaches 20. The main reason why Dg_k/N^2 and r_k deviate for $k > 30$ is the discrete nature of the Baker map. The ciphers K_1K_2 and K_1K_3 differ in an interval $[N_s^{(1)}, N_{s+1}^{(1)}]$ of lengths 8 and 2, respectively. In the course of iterations, the pixels from the vertical rectangle

$$\{(i, j) | N_s^{(1)} \le i < N_{s+1}^{(1)}, \quad 0 \le j < N\}$$

will not be mapped to *all* pixels in the image but to a smaller set.[1] This is caused by periodicities which are present in the discretized map. Therefore, Dg_k saturates at a different number than $D_k = N^2 r_k$, which always reaches the total number of pixels, N^2, as $k \to \infty$.

The second set of experiments was performed with the same ciphering keys, K_1, K_2, K_3, for a real test image shown in Fig. 6. Note that the number of misplaced pixels with different gray levels, Dg_k, is slightly lower compared to the previous case, which somewhat decreases the accuracy of Eq. (5). This is caused by the finite number of gray levels (256) in the image. Some misplaced pixels may accidentally land on pixels with the same gray level. The frequency of this happening depends on the histogram of the image and on the total number of gray levels. The more uniform the histogram is, or, the more gray levels are in the image, the better the accuracy of Eq. (5).

	K_1K_2		K_1K_3	
k	Dg_k/N^2	$r_k = D_k/N^2$	Dg_k/N^2	$r_k = D_k/N^2$
1	1.581e-2	1.695e-2	2.110e-3	4.237e-3
2	3.183e-2	3.361e-2	4.224e-3	8.457e-3
3	4.706e-2	4.999e-2	6.239e-2	1.266e-2
5	7.632e-2	8.192e-2	1.020e-2	2.100e-2
10	0.1410	0.1571	1.925e-2	4.157e-2
20	0.2408	0.2896	3.468e-2	8.142e-2
40	0.3619	0.4953	5.710e-2	0.1562
80	0.4628	0.7453	8.019e-2	0.2880
160	0.5035	0.9351	9.726e-2	0.4931
320	0.5067	0.9958	0.1003	0.7430

Set #2.

Both sets of experiments suggest that D_k forms an *upper bound* on the number of pixels in which two encrypted images differ. For a small number of iterations $k < 30$, Eq. (5) gives an accurate estimate of Dg_k. The difference between Dg_k and D_k is caused by a small number of gray levels in the image (256) compared to the total number of pixels (472^2). D_k will be a better estimate of Dg_k for color images which may have up to 256^3 different colors. The accuracy of D_k will clearly depend on the histogram of the image. The disagreement between Dg_k and D_k in the second experiment of Set #2 for K_1 K_3 is caused by the fact that in the first iteration a large portion of the misplaced pixels was accidentally mapped to pixels with the same gray levels. For the next iterations up to 30 Eq. (5) predicted the correct trend, i.e. $Dg_k/Dg_{k-1} \approx D_k/D_{k-1}$, $k = 1, \ldots, 30$. As discussed before, the main reason why Dg_k/N^2 and r_k deviate for $k > 30$ is the discrete nature and periodicities of the discretized Baker map.

The analysis above suggests that the formula for D_k can be safely used for typical images for $k < 30$. This value of k is more than enough to guarantee a safe ciphering method.

6. Cryptographic Strength of the Permutation Cipher Generated by the Baker Map

In this section we study the cryptographic strength of the permutations generated by the Baker map for both known plain-text and cipher-text only attacks. We emphasize that here we are investigating the cipher based on the Baker map with the three-dimensional mixing and the diffusion removed. The most important question for the security of our simplified cipher is: What size of d will guarantee that two keys actually represent different keys and cannot be used interchangeably to decipher a typical image enciphered by the other key?

Equation (5) can be used for *measuring the similarity between keys*. Suppose that two keys represented by unions of intervals, U_1, U_2 differ in intervals of total length d. Formally, let

$$U_1 = [0,\, N_1^{(1)}) \cup [N_1^{(1)},\, N_2^{(1)}) \cup \cdots \cup [N_{k-1}^{(1)},\, N_k^{(1)})\,.$$

$$U_2 = [0,\, N_1^{(2)}) \cup [N_1^{(2)},\, N_2^{(2)}) \cup \cdots \cup [N_{m-1}^{(2)},\, N_m^{(2)})\,.$$

Then

$$d = N - \sum_{[N_{j-1}^{(1)},\, N_j^{(1)}] \in U_1 \cap U_2} (N_j^{(1)} - N_{j-1}^{(1)})\,.$$

If we want to break the permutation cipher using an intelligent direct search for key, we need to

[1] This set is shown after 320 iterations in Figs. 21 and 22 for K_1 K_2 and K_1 K_3, respectively.

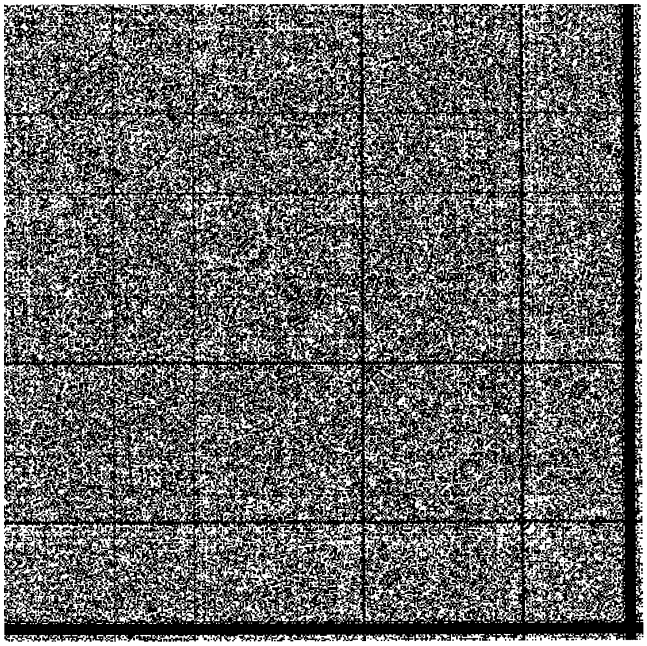


Fig. 21. Pixels with different gray levels after 320 iterations of the Baker map with keys K_1 and K_2.

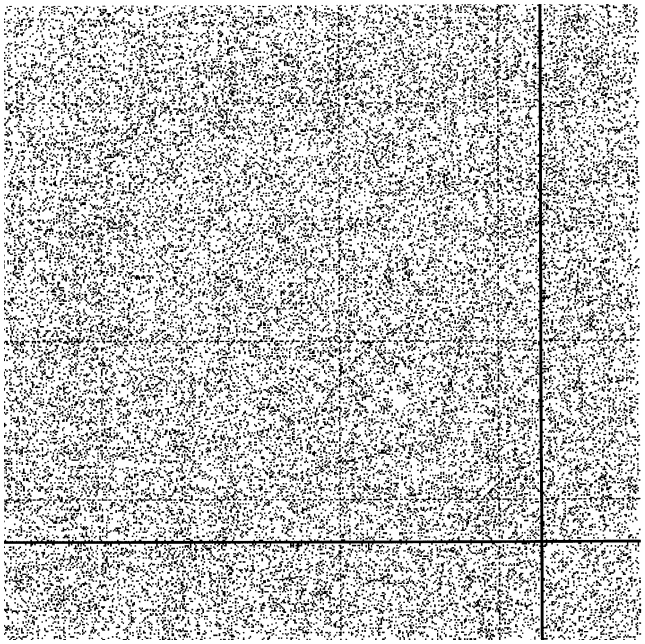


Fig. 22. Pixels with different gray levels after 320 iterations of the Baker map with keys K_1 and K_3.

distinguish two cases: (a) The plain-text is available to the cryptanalyst, and (b) the plain-text is not available.

6.1. *Known plain-text type of attack*

In this case, two keys as said to be different, or dissimilar, if after 30 iterations the encrypted images differ by the same margin as two *randomly chosen* images. Since the expected number of the same pixels for two random images is N^2/L with the standard deviation equal to $N/\sqrt{L}$, two ciphers will be considered different if at least $(1 - 1/L \pm 1/N\sqrt{L}) \times 100\%$ of pixels are misplaced. Taking the midpoint of this interval, the following equation for d needs to be satisfied (for 256 gray levels)

$$1 - (1 - d/N)^{30} = 1 - 1/256\,.$$

By solving the equation, $d \approx 0.17 \times N$. In other words, when two ciphering keys differ by more than 17%, they are considered to be dissimilar for a known plain-text type of attack. The ratio d/N can be taken as a measure of similarity between two keys. When $d/N < 0.17$, the two keys are said to be similar and can be used for ciphering/deciphering phase interchangeably.

Now that a criteria for determining whether or not two keys are similar has been developed, it is possible to estimate the maximal number of keys which are similar to a given key. Two similar keys may differ in intervals of length at most 17%. Therefore, the number of keys similar to the key $K = (n_1, \ldots, n_k)$ is equal to

$$K_{\max} = \sum_{n_i \le N \times 0.17} K(n_i)\,,$$

where $K(n_i)$ is the number of all ciphering keys of length n_i. The summation will be maximal when all n_i are equal to $0.17 \times N$. Consequently, $K_{\max} \le 100/17 \times 2^{0.17 \times N}$, and the total number of different clusters for an $N \times N$ image is at least

$$\frac{K(N)}{100/17 \times 2^{0.17 \times N}} = \frac{17}{100} 2^{N-1-0.17 \times N}\,.$$

This lower estimate for the number of different clusters for Version B of the Baker map is calculated in the table below.

# of keys		# of clusters	
N	$k(N)$	Cipher-text only	Known plain-text
64	1.8e19	1.0922e16	8.3202e14
128	3.4e38	2.3858e33	8.1441e30
256	1.2e77	1.1384e68	7.8032e62
512	1.4e154	2.5917e137	7.1635e126
1024	1.8e308	1.3434e276	6.0372e254

The results clearly suggest that the number of dissimilar ciphering keys is still very large in spite of the presence of clusters of similar keys. Therefore, it can be concluded that the proposed ciphering technique is secure with respect to a known plain-text type of attack.

6.2. *Cipher-text only type of attack*

In the second case, when the plain-text is not available, pattern recognition analysis and high level image understanding tasks have to be performed to determine if there is some pattern in the decrypted image. For this problem, a similarity between keys can be defined with less severe restrictions. Namely, one could probably safely assume that just 90% of differently placed pixels would prevent one from being able to recognize a pattern in the decrypted image. Similar arguments as before will demonstrate that two keys can be considered dissimilar (for the cipher-text only type of attack) if they differ in just 10% of their length. The number of clusters in this case increases as $0.1 \times 2^{N-1-0.1\times 10}$. This calculation is based on performing just 20 iterations. Specific numbers for selected values of N are shown in the table above.

We close this section devoted to security issues by a few remarks on diffusion. The clusters of similar keys are caused by nondiffusive properties of the cipher. This means that changes to one pixel level do not spread to neighboring pixels, and small changes in encryption keys do not drastically change the cipher-text. This is highly undesirable from the security point of view. The cipher would fall to a chosen plain-text type of attack. As indicated by computer experiments, when the diffusion is brought in, there are no clusters of similar keys — the "clusters" will contain just one key.

7. Pseudo-Random Number Generator Based on Chaotic Maps

Since an encrypted image resembles an uncorrelated random static on a TV monitor, one may try to generate a sequence of pseudo-random numbers $\{x_i\}_i$ by reading the gray levels of pixels in a row-by-row manner or some other scanning pattern. Starting with an $M \times N$ image with L gray levels (one could start, for example, with the image consisting of a black square) after performing k iterations, one obtains $M \times N$ pseudo-random integers in the range $[0, L-1]$. Majority of traditional pseudo-random number generators (PRNG) generate the next number in the sequence by following certain deterministic rule, i.e. there is a deterministic relationship between x_i and x_{i+1}. The PRNG based on three-dimensional maps is nontraditional because it does not have this property. If more than $M \times N$ pseudo-random numbers are needed, another k iterations of the chaotic map can be performed to obtain another set of $M \times N$ pseudo-random numbers, etc.

The basic two requirements for any PRNG concern the statistical properties of the stream of numbers produced. First, the pseudo-random sequence should satisfy all known statistical tests for randomness. Second, the period of the PRNG should be as large as possible. We subjected our PRNG based on discretized chaotic maps to several statistical tests including the uniformity of distribution test, the coupon collector's test, the permutation test, the poker test, and the serial pairs test. Detailed description of these statistical tests can be found in [Karian, 1991]. The tests were performed on a 472×472 image with 256 gray levels. All five tests were satisfied by the sequence of pseudo-random numbers obtained from an image of a black square encrypted using the Baker map with 9 iterations. The numbers were read in a row-by-row manner. Computer experiments done with other scanning patterns suggest that the properties of the pseudo-random sequence do not depend on the scanning pattern.

To calculate the period of the PRNG, we write the permutation P induced by the Baker map as a collection of r cycles $P = \{C_1, C_2, \ldots, C_r\}$. The length of the ith cycle will be denoted as $|C_i|$. Recalling Eq. (4) for the three-dimensional chaotic map, we further define a phase of the ith cycle as

$$Ph(C_i) = \sum_{(p,q)\in C_i} \overline{h}(p, q) \mod L\,.$$

The gray level increases by $Ph(C_i)$ in each cycle during which a pixel from that cycle returns to its original position. Therefore, the ith cycle will start repeating after $w_i|C_i|$ iterations, where w_i is the smallest number such that $w_i Ph(C_i) \equiv 0 \mod L$, or

$$w_i = \frac{LCM(Ph(C_i), L)}{Ph(C_i)}\,.$$

The order of the permutation P (or the period of the random number generator) is

$$\text{Period} = M \times N \times LCM\{w_1|C_1|, w_2|C_2|, \ldots, w_r|C_r|\}\,.$$

When the complete cipher with diffusion is used instead of Eq. (4), the period is expected to be much larger. Estimates of the period for this case are subject of future research.

Not every PRNG is suitable for encryption purposes in spite of the fact that it satisfies all known statistical tests. The third important requirement is that it should be computationally hard to determine the key and the seed based on the knowledge of a finite segment of pseudo-random numbers. This is equivalent to breaking the cipher using cipher-text only type of attack. As described in Sec. 6, the complexity of a direct key search increases exponentially with N as $2^{0.9\times N-1}$. This indicates that the seed and the key cannot be recovered by an exhaustive search.

8. Examples of Other Chaotic Maps

8.1. *The Cat map*

One very well-studied two-dimensional chaotic map is the **Cat map** introduced by Arnold and Avez [1968]. The action of the map on the unit square is often explained with a picture of a cat, which gave the map its name. The mathematical formula is:

$$C(x,\, y) = (x + y \mod 1, \quad x + 2y \mod 1)$$
$$= \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \mod 1\,,$$

where $a \bmod 1$ means the fractional part of a for any real a. Denoting the square 2×2 matrix as A, the map can be written simply as $C(x,\, y) = A(x,\, y)^T \bmod 1$, where $(\ )^T$ stands for a vector transpose. The Cat map is most easily described in geometric terms. The unit square $ABCD$, is linearly stretched so that the point $C = (1,\, 1)$ is moved to $(2,\, 3)$, and B is moved to $(1,\, 1)$. This stretching phase is described by the matrix A. After applying the mod operator, the pieces of the image lying in squares other than the unit square are cut and shifted back to the unit square (see Fig. 23). Similar to the Baker map, the Cat map is discontinuous along the lines of cutting.

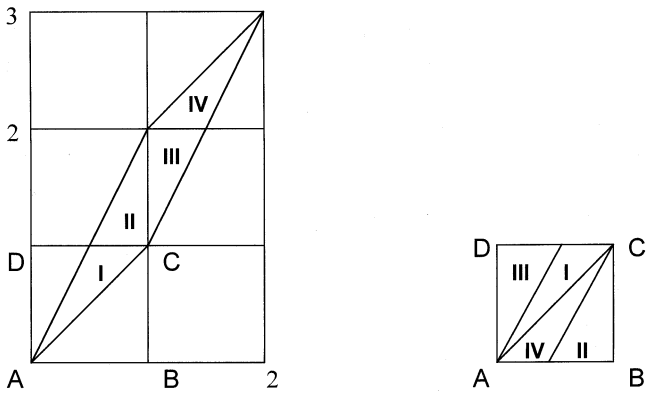


Fig. 23. The Cat map.

It seems natural to use the elements of the matrix A as the parameters for the generalized version of the Cat map. A general matrix A,

$$A = \begin{pmatrix} t & u \\ v & w \end{pmatrix}$$

with integer elements will be denoted $A_{(t,\, u,\, v,\, w)}$. Not all choices of the parameters will produce a correct generalization of the Cat map. In particular, to make sure that the map is one-to-one, the determinant of A, $|A| = tw - uv$, has to be equal to 1. We note that the four tuple $(t,\, u,\, v,\, w)$ produces the same cipher as the four-tuple ($t \bmod N$, $u \bmod N$, $v \bmod N$, $w \bmod N$).

The discretized version of the Cat map is obtained simply by changing the range of $(x,\, y)$ from the unit square $I \times I$ to the discrete lattice $\mathbf{N}_0^N \times \mathbf{N}_0^N$

$$A_{(t,\, u,\, v,\, w)}(r,\, s)^T \mod N\,.$$

The map $A_{(t,\, u,\, v,\, w)}$ transforms the square lattice of points $\mathbf{N}_0^N \times \mathbf{N}_0^N$ onto itself in a one-to-one manner. The results of applying the discretized Cat map with the matrix $A_{(1,\, 1,\, 1,\, 2)}$ to the same test image once and nine times are shown in Figs. 24 and 25.

To complete the construction of a symmetric cipher based on the Cat map, the extension method of Sec. 3.5 and methods of Sec. 3.6 can be directly applied to the generalized discretized Cat map.

The study of the cipher based on the Cat map is not pursued further in this paper because the map does not lead to ciphers with sufficiently many keys (permutations). Since the four-tuple $(t + k_1 N,\, u + k_2 N,\, v + k_3 N, w + k_4 N)$ generates the same cipher as the four-tuple $(t,\, u,\, v,\, w)$ for any $k_1,\, k_2,\, k_3,\, k_4 \in \mathbf{Z}$, the values of $t,\, u,\, v,\, w$ can be restricted to the set $\mathbf{N}_0^N$. The total number of

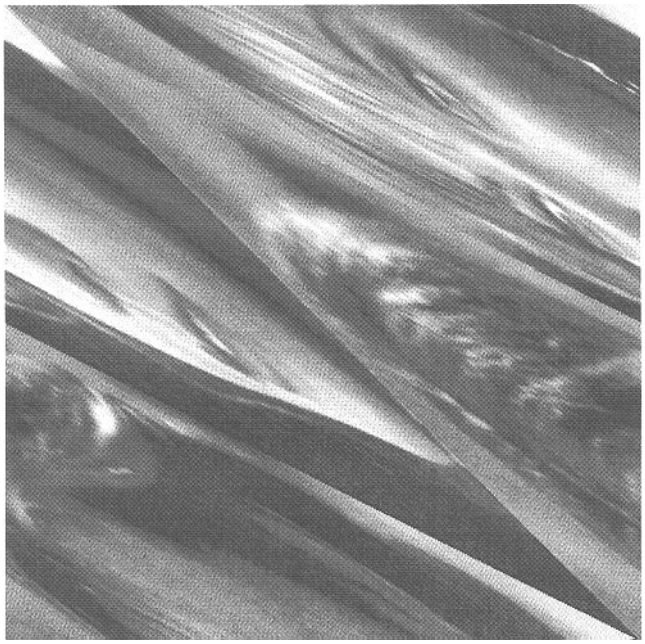


Fig. 24. The test image after applying the Cat map once.

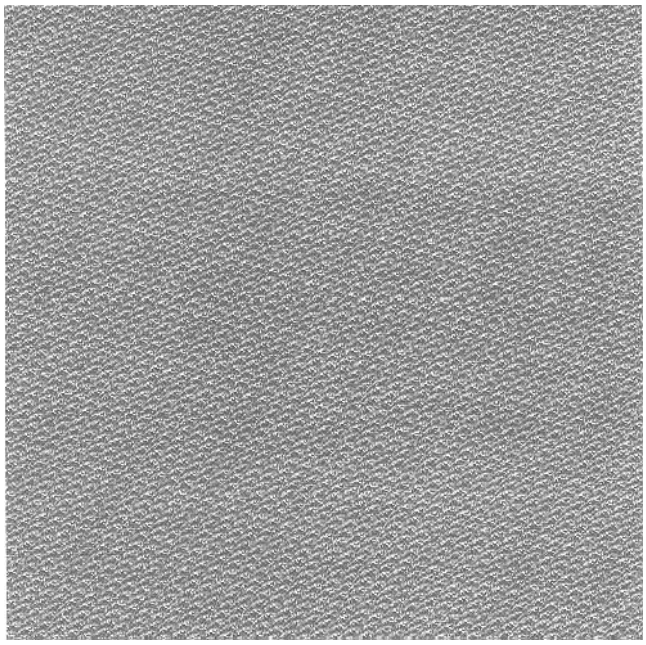


Fig. 25. The test image after applying the Cat map nine times.

ciphering keys for the Cat map is therefore smaller[2] than N^4. For a 512×512 image, this number is approximately 6×10^{10}. This number is unacceptably low and makes the direct search for the key to a viable attack.

8.2. *The Standard map*

The **Standard map** is described with the following formula:

$$S(x, y) = (S_1, S_2) = (x + y \mod 2\pi, \; y - k \sin(x + y) \mod 2\pi), \tag{6}$$

where k is a positive constant. The Standard map can be easily generalized to the following form while keeping its invertibility:

$$S_1(x, y) = x + F(y) \mod 2\pi$$
$$S_2(x, y) = y + G(S_1) \mod 2\pi$$
and
$$y = S_2 - G(S_1) \mod 2\pi$$
$$x = S_1 - F(y) \mod 2\pi,$$

where F and G are arbitrary functions containing parameters (the key). The structure of the Standard map strongly resembles Feistel networks [Schneier, 1996] so commonly used in the design of almost all block symmetric ciphers. The Standard map can be discretized in a straightforward manner by substituting $S_1 = xN/2\pi$, $S_2 = yN/2\pi$, $K = kN/2\pi$ into Eq. (6) for the Standard map

$$(i, j) \to S(i, j) = (S_1(i, j), S_2(i, j))$$
$$S_1(i, j) = i + j \mod N$$
$$S_2(i, j) = j + K \sin\left(\frac{S_1 N}{2\pi}\right) \mod N,$$

where K is a positive constant, and N is the number of pixels in each row of a square image. An example of the test image encrypted using this map is shown in Figs. 26 and 27. The Standard map can be easily generalized to the following form while keeping its invertibility:

$$S_1(i, j) = i + \Psi(j) \mod N$$
$$S_2(i, j) = j + \Phi(S_1) \mod N$$
and
$$j = S_2 - \Phi(S_1) \mod N$$
$$i = S_1 - \Psi(j) \mod N,$$

[2]Since a four tuple is a cipher if it satisfies the additional condition $|tw - uv| = 1$, the total number of keys is expected to grow with the third power only.

Fig. 26. The test image after one iteration of the Standard map.

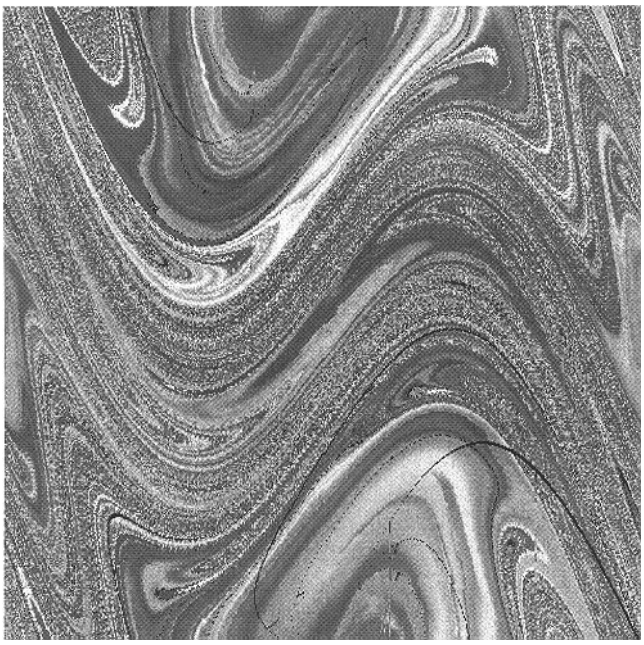


Fig. 27. The test image after nine iterations of the Standard map.

where Φ and Ψ are arbitrary functions containing parameters (the key). It has been shown by Rannou [1974] that the discretized Standard map does not behave as a typical random permutation. She attributed this to the symmetries of the Standard map (6). Once those symmetries were broken the average length cycles and the average number of cycles were close to those of a typical random map. Detailed study of the applicability of the generalized Standard map to encryption will be a part of the future research. In the remainder of the paper, we concentrate on important security issues and on the structure of the permutations induced by the Baker map.

9. Relationship Between Chaos and Cryptosystems

Chua and Brown [1996] pointed out a similarity between pseudo-random number generators used in stream ciphers[3] and one-dimensional chaotic systems constrained to periodic invariant sets. In this paper, we argue that there is a relationship between two-dimensional chaotic systems and symmetric block cryptosystems. Among other things, any good cryptosystem should:

1. Map plain-text to a random cipher-text. There should not be any patterns in the cipher-text, if the cryptosystem is good.
2. Be sensitive with respect to plain-text. This means that flipping one bit in the plain-text creates completely different cipher-text.
3. Be sensitive with respect to keys. This means that flipping one bit in a key creates completely different cipher-text when applied to the same plain-text.

In addition to these requirements, it is a well-known fact that virtually all symmetric block encryption methods are *iterative schemes* and work by iterating some basic encryption function several times. DES has 16 rounds, IDEA 8 rounds, LOKI 16 rounds, Blowfish 16 rounds, GOST 32, Khufu and Khafre 24. A large number of symmetric ciphers are based on an iterative scheme called Feistel network. Feistel network transforms a block (L, R) according to the following invertible formula

$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus f(R_i, K_i)\,,$$

where $f(.\ ,\ .)$ is an arbitrary function of two bit strings and K_i is the subkey derived from a passphrase for the ith iterative step. The

[3]Example of such ciphers are the Blum-Micali encryption scheme or the BBS scheme [Schneier, 1996]

operation $\oplus$ is usually the bitwise XOR but could be some other operation. It is worth mentioning that many chaotic maps expressed in the form of a discrete mapping have the same structure as the Feistel network. For example, the Standard map and the Hénon map can be put into this form.

Although the notion of chaos has not been satisfactorily defined (see the discussion by Brown and Chua [1996]), on a heuristic level some properties of chaotic system can be related to the basic properties of ciphering systems. Two important properties of chaotic systems are defined below.

Topological transitivity. Given a metric space X and a mapping $f : X \to X$, we say that f is topologically transitive on X if for any two open sets U, $V \subset X$, there is $n \geq 0$ such that $f^n(U) \cap V \neq \varnothing$.

Sensitivity to initial conditions. The map f is said to be sensitive to initial conditions if there is a $\delta > 0$ such that for any $x \in X$ and for any neighborhood H_x of x there is $y \in H_x$ such that $|f^n(x) - f^n(y)| > \delta$.

Devaney [1989] defined chaos as a system satisfying both properties above together with the requirement that the set of periodic points of f be dense in X. Banks *et al.* [1992] have shown that the topological transitivity combined with the density of periodic points already implies sensitivity to initial conditions. Consequently, Ingraham [1992] requires topological transitivity and sensitivity to initial conditions as the defining properties of chaos.

Topological transitivity together with the sensitivity to initial conditions cause that the state space X is "mixed" with the action of the map f. This can be related to the requirement No. 1 for ciphering systems. Sensitivity to initial conditions heuristically corresponds to the sensitivity of ciphering systems with respect to the plain-text if we substitute the plain-text for the initial condition. Finally, most chaotic systems exhibit sensitive dependency on control parameters. This in turn nicely corresponds to the requirement of sensitivity with respect to the key if we consider the key as a parameter for the ciphering transformation.

However, there is one important difference between chaos and encryption. Cryptosystems work on finite sets, while chaotic systems have meaning only on a continuum, an infinite set. One of the goals of our future research is to establish a formal relationship between chaos and cryptosystems, and use this connection to enrich both fields. Encryption would readily benefit because one could use a large number of powerful mathematical tools previously developed for nonlinear dynamic systems. For example, we could use the concept of Lyapunov number to quantify diffusion in cryptosystems. The minimal number of iterations for any given cryptosystem is usually estimated by the designers and there is no general method which would guide us as to how many iterations are actually needed to guarantee a secure cipher. For example, IDEA has 8 rounds, but it is generally accepted that as few as 6 or even 4 produce a safe cipher as well. Why the designers have chosen 8 and not 6? How can we justify the number of iterations in an encryption scheme? How many iterations are necessary for our chaos-based encryption method? The concept of Kolmogoroff entropy from the theory of dynamic systems might help us to answer these questions. Kolmogoroff entropy measures the rate with which information about initial conditions is lost in the course of iterations. In addition to the applications stated above, we expect that a successful connection between encryption and discretized chaos would lead to new attacks for breaking symmetric ciphers and to new cryptanalytic techniques.

On the other hand, the impact of cryptanalytic theoretical tools in chaos theory can only be guessed right now. It seems that symbolic dynamics [Hao, 1990] would be the candidate number one for this type of application.

The main problem that needs to be solved is a correct generalization of chaos from a continuum to finite sets. Although the size of the sets will usually be of the order of 2^{64} or 2^{128} (the typical size of all possible encryption blocks), the sets are nevertheless finite. Any definition of chaos on finite sets should merge with the classical definition as the number of elements tends to infinity. This correspondence principle will be the guide of our research. One possible approach towards the definition of chaos on finite sets is using symbolic dynamics and the concept of randomness on finite sets. The latter topic has been extensively and successfully studied in the past [Knuth, 1991].

10. Conclusion and Future Directions

In this paper, it is shown how to adapt invertible chaotic two-dimensional maps on a torus or on a

rectangle for the purpose of encryption. The map is first generalized by introducing parameters and then discretized to a finite rectangular lattice of points. Then the map is extended to three dimensions to obtain a more complicated substitution cipher. This cipher alone can turn an arbitrary plain-text into random looking cipher-text. This is utilized for constructing a nontraditional random number generator. Since the substitution cipher has no diffusion properties with respect to plain-text, it is finally composed with a simple diffusion mechanism. The resulting cipher appears to have good diffusion properties with respect to both the key and the plain-text. The properties of the permutations induced by the Baker map are shown to correspond to a typical random permutation. In particular, computer experiments done for the Baker map with many different ciphering keys demonstrate that the average length of cycles and the average number of different cycles have values similar to those for random permutation. The paper closes with some general remarks on the similarity between discretized chaos and encryption schemes.

The main features of the encryption scheme studied in this paper are a variable key length, a relatively large block size (several kB or more), and a high encryption rate (1 Mb unoptimized C code on a 60 MHz Pentium). The cipher is based on two-dimensional chaotic maps, which are used for creating complex, key-dependent permutations. Unlike most of today's symmetric encryption schemes, which rely on complex substitution rules while somewhat neglecting the role of permutations, the new cipher is based on complex permutations composed with a relatively simple diffusion mechanism.

Advantages

- The method is a private key, symmetric block product cipher.
- It is simple, fast, and lends itself for efficient software implementation.
- Unoptimized C code achieved an encryption rate of 1 Mbit per second on a 66 MHz Pentium computer.
- Variable key length and variable block size.
- The encrypted file has the same size as the original file.
- The time to perform encryption and decryption is the same.

Disadvantages

- The choices for the ciphering key depend on the block size. Files smaller than 10 kB would have to be padded a lot to guarantee sufficiently many encryption keys. This will, however, increase the size of the data to be transmitted.

The future research directions will be directed to a more detailed study of security analysis of the proposed cipher. We plan to use standard cryptanalytic tools, such as differential and linear cryptography to further assure the safety and robustness of the cipher. Also, we intend to study other maps and their discretized forms. The generalized Standard map, for example, provides a general framework for a whole new class of encryption schemes resembling in structure Feistel networks. One of the major goals of our future effort is establishing a connection between discretized chaotic systems and encryption schemes. This would enable us to quantify diffusion and sensitivity with respect to key and the plain-text using concepts, such as entropy or Lyapunov exponents. In order to do that, an appropriate framework and definition of chaos on finite metric spaces needs to be established.

Acknowledgments

The work on this paper was supported by Rome Laboratory, Air Force Material Command, USAF, under grant number F30602-96-1-0047. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Rome Laboratory, or the U.S. Government.

References

Arnold, E. A. & Avez, A. [1968] *Ergodic Problems of Classical Mechanics* (Benjamin, W. A., New Jersey) Chap. 1, p. 6.

Banks, J., Brooks, J., Cairns G., Davis, G. & Stacy P. [1992] "On Devaney's definition of chaos," *Amer. Math. Monthly* **99**, April '92, 332–334.

Bernstein, G. M. & Lieberman, M. A. [1991] *Method and Apparatus for Generating Secure Random Numbers Using Chaos*, US Patent No. 5007087, Apr. 9.

Bianco, M. E. & Mayhew, G. L. [1994] *High Speed Encryption System and Method*, US Patent No. 5365588, Nov. 15.

Bianco, M. E. & Reed, D. A. [1991] *Encryption System Based on Chaos Theory*, US Patent No. 5048086, Sept. 10.

Biham, E. [1991] "Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT '91," in *Advances in Cryptology — EUROCRYPT '91*, ed. Davies, D. W., LNCS 547 (Springer-Verlag, Berlin), pp. 532–534.

Brassard, G. [1988] *Modern Cryptography* (Springer-Verlag, New York), Chap. 6, p. 79.

Brassard, G. [1997] "Searching a quantum phone book," *Science* **275**, 627.

Brown, R. & Chua, L. O. [1996] "Clarifying chaos: Examples and counterexamples," *Int. J. Bifurcation and Chaos* **6**(2), 219–249.

Caroll, T. & Pecora, L. M. [1990] "Synchronization in chaotic systems," *Phys. Rev. Lett.* **64**(8), 821–824.

Caroll, T. & Pecora, L. M. [1991] "Driving systems with chaotic signals," *Phys. Rev.* **A44**(4), 2374–2383.

Caroll, T. & Pecora, L. M. [1992] "A circuit for studying the synchronization of chaotic systems," *Int. J. Bifurcation and Chaos* **2**(3), 659–667.

Caroll, T. & Pecora, L. M. [1993a] "Cascading synchronized chaotic systems," *Physica* **D67**, 126–140.

Caroll, T. & Pecora, L. M. [1993b] *System for Producing Synchronized Signals*, US Patent No. 5245660, Sept. 14.

Caroll, T. & Pecora, L. M. [1995a] *Cascading Synchronized Chaotic Systems*, US Patent No. 5379346, Jan. 3.

Caroll, T. & Pecora, L. M. [1995b] *Synchronization of Nonautonomous Chaotic Systems*, US Patent No. 5473694, Dec. 5.

Cuomo, K. M. & Oppenheim, A. V. [1995] "Circuit implementation of synchronized chaos with applications to communications," *Phys. Rev. Lett.* **71**(1), 65–68.

Cuomo, K. M. & Oppenheim, A. V. [1994] *Communication Using Synchronized Chaotic Systems*, US Patent No. 5291555, Mar. 1.

Cuomo, K. M., Oppenheim, A. V. & Isabelle, S. H. "Spread spectrum modulation and signal masking using synchronized chaotic systems," RLE Technical Report No. 570, Research Laboratory of Electronics, MIT, Mass., pp. i–iii and 1–37.

Deffeyes, K. S. [1991] *Encryption System and Method*, US Patent No. 5001754, Mar. 19.

Devaney, R. L. [1989] *An Introduction to Chaotic Dynamical Systems* (Addison-Wesley, Redwood, California), p. 50.

Feller, W. [1957] *An Introduction to Probability Theory and Its Applications* (John Wiley, New York), pp. 242–243.

Fridrich, J. & Geer, J. [1995a] "Reconstruction of blurred orbits under finite resolution," *J. Appl. Math. Comp.* **71**, 227–245.

Fridrich, J. & Geer, J. [1995b] "Reconstruction of chaotic orbits under finite resolution," *J. Appl. Math. Comp.* **80**, 129–159.

Fridrich, J. [1997a] "Discrete-time dynamical systems under observational uncertainty," *J. Appl. Math. Comp.* **83**, 181–207.

Fridrich J. [1997b] *Secure Image Ciphering Based on Chaos*, Final Technical Report (USAF, Rome Laboratory, New York).

Gutowicz, H. A. [1993] "Cryptography with dynamical systems," in *Cellular Automata and Cooperative Systems*, eds. Boccara, N., Goles, E., Martinez, S. & Picco, P. (Kluwer Acad. Publ., Boston), pp. 237–274.

Gutowicz, H. A. [1994] *Method and Apparatus for Encryption, Decryption and Authentication Using Dynamical Systems*, US Patent No. 5365589, Nov. 15.

Habutsu, T., Nishio, Y., Sasase, I. & Mori, S. [1991] "A secret cryptosystem by iterating a chaotic map," in *Advances in Cryptology — EUROCRYPT '91*, ed. Davies, D. W., LNCS 547 (Springer-Verlag, Berlin), pp. 127–140.

Hao B.-L., [1990] *Chaos II* (World Scientific, Singapore), p. 27.

Ingraham, R. L. [1992] *A Survey of Nonlinear Dynamics* (*Chaos Theory*) (World Scientific, Singapore).

Jackson, E. A. [1991] *Perspectives in Nonlinear Dynamics*, Vol. 2 (Cambridge Univ. Press, Cambridge), p. 33.

Karian, Z. A. & Dudewicz, E. J. [1991] *Modern Statistical Systems, and GPSS Simulation* (Computer Science Press, New York), pp. 95–129.

Knuth, D. E. [1991] *The Art of Computer Programming* (*Seminumerical Algorithms*), Vol. 2 (Addison-Wesley, Reading, MA), p. 142.

Kocarev, L. J., Halle, K. S., Eckert, K. & Chua, L. O. [1992] "Experimental demonstration of secure communications via chaotic synchronization," *Int. J. Bifurcation and Chaos* **2**(3), 709–713.

Matthews, R. [1989] "On the derivation of a 'chaotic' encryption algorithm," *Cryptologia* **XIII**(1), 29–42.

Murali, K. & Lakshmanan, M. [1993] "Transmission of signals by synchronization in a chaotic Van der Pol-Duffing oscillator," *Phys. Rev.* **E48**(3), R1624–R1626.

Papadimitriou, S., Bezerianos A. & Bountis, T. [1997] "Secure communication with chaotic systems of difference equations," *IEEE Trans. Comp.* **46**(1), 27–38.

Parlitz, D., Chua, L. O., Kocarev, L., Halle, K. S. & Shang, A. [1992] "Transmission of digital signals by chaotic synchronization," *Int. J. Bifurcation and Chaos* **2**(3), 973–977.

Pichler, F. & Scharinger, J. [1994] "Ciphering by Bernoulli shifts in finite Abelian groups," in *Contributions to General Algebra, Proc. Linz-Conference*, pp. 465–476.

Pichler, F. & Scharinger, J. [1995] "Ciphering by Bernoulli-shifts in finite Abelian groups," Johannes Kepler University, Linz, Austria, May 10, preprint, 1995.

Protopopescu, V. A., Santoro, R. T. & Tolliver, J. S. [1995] *Fast and Secure Encryption-Decryption Method Based on Chaotic Dynamics*, US Patent No. 5479513, Dec. 26.

Rannou, F. [1974] "Numerical study of discrete plane area-preserving map," *Astron. Astrophys.* **31**, 289–301.

Schneier, B. [1996] *Applied Cryptography* (Wiley, New York).

Sloane, N. J. [1983] "Encrypting by random rotations," in *Proc. Workshop on Cryptography*, eds. Goos G., Hartmanis, J. & Feuerstein, B. (LNCS, Germany), pp. 71–128.

Shannon, C. [1949] "Communication theory of secrecy systems," *Bell System Tech. J.* **28**, 656–715.

Wheeler, D. D. [1989] "Problems with chaotic cryptosystems," *Cryptologia* **XII**(3), 243–250.