An Excellent Permutation Operator for Cryptographic Applications

Josef Scharinger

Johannes Kepler University,
Institute of Computational Perception, 4040 Linz, Austria
Josef.Scharinger@jku.at

Abstract. Permutations are a core component of almost every cipher. No matter if we consider the DES, AES or most of the other encryption algorithms relevant nowadays, we always find permutation operators as essential building blocks inside. In this contribution we will introduce key-dependent permutation operators of provably excellent quality inspired by chaotic Kolmogorov flows.

From chaotic systems theory it is known that the class of Kolmogorov flows exhibits the highest degree of instability among all dynamical systems. As will be derived and proven in detail in this paper, these outstanding properties make them a perfect inspiration for developing a novel class of strong cryptographic permutation operators.

1 Introduction

In recent years chaos theory has definitely been among the hot topics in systems theory. Remarkable progress has been made in the analysis of chaotic systems and up to a certain extent also in their application. Nevertheless, success in applications lags progress in analysis. This may well be related to the fact that many of the promising systems are defined to act on a continuous phase space and it is often quite difficult or even impossible to find discrete counterparts that preserve all the nice features present in the continuous case.

In this contribution we focus on chaotic Kolmogorov systems [6]. In the well-established continuous form they provide a family of highly unstable systems where it has been proven [1,3] that every member of this family provides perfect mixing of the underlying phase space which makes them a tempting choice for realizing excellent permutation operators. However, for practical computer applications such a permutation operator is only useful if it can be applied to mixing elements arranged on a discrete grid where we have to deal with integer grid positions.

It is the main purpose of this paper to show that it is possible to derive adequate discrete counterparts for classical continuous Kolmogorov systems. Additionally we provide a detailed analysis under which criteria these novel discrete Kolmogorov systems offer high-quality permutation operators. Availability of such discrete permutation systems will finally be utilized to sketch several examples of potential applications for important tasks in information security such as symmetric block ciphering, message digest computation, or copyright protection via digital watermarking.

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 317–326, 2005.

2 Chaotic Kolmogorov Systems

2.1 Continuous Kolmogorov Systems

Continuous Kolmogorov systems [1,3,6] act as permutation operators upon the unit square. Figure 1 is intended to give a notion of the dynamics associated with a specific Kolmogorov system parameterized by the partition $\pi = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$. As can be seen, the unit square is first partitioned into vertical strips which are then stretched in the horizontal and squeezed in the vertical direction and finally stacked atop of each other. Just after a few applications (see Fig. 1 from top left to bottom right) this iterated stretching, squeezing and folding achieves perfect mixing of the elements within the state space.

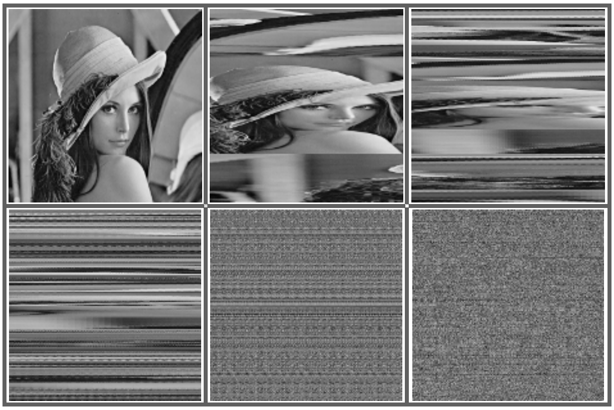

Fig. 1. Illustrating the chaotic and mixing dynamics associated when iterating a Kolmogorov system

Formally this process of stretching, squeezing and folding is specified as follows. Given a partition $\pi = (p_1, p_2, \ldots, p_k)$, $0 < p_i < 1$ and $\sum_{i=1}^{k} p_i = 1$ of the unit interval $\mathbb{U}$ and stretching and squeezing factors defined by $q_i = \frac{1}{p_i}$. Furthermore, let F_i defined by $F_1 = 0$ and $F_i = F_{i-1} + p_{i-1}$ denote the left border of the vertical strip containing the point $(x, y) \in \mathbb{E}$ to transform. Then the continuous Kolmogorov system T_π will move $(x, y) \in [F_i, F_i + p_i) \times [0, 1)$ to the position

$$T_\pi(x, y) = (q_i(x - F_i), \frac{y}{q_i} + F_i). \tag{1}$$

2.2 Discrete Kolmogorov Systems

In our notation a specific discrete Kolmogorov system for permuting a data block of dimensions $n \times n$ is defined by a list $\delta = (n_1, n_2, \ldots, n_k)$, $0 < n_i < n$ and $\sum_{i=1}^{k} n_i = n$ of positive integers that adhere to the restriction that all $n_i \in \delta$ must partition the side length n. Furthermore let the quantities q_i be defined by

$q_i = \frac{n}{n_i}$ and let N_i specified by $N_1 = 0$ and $N_i = N_{i-1} + n_{i-1}$ denote the left border of the vertical strip that contains the point (x, y) to transform. Then the discrete Kolmogorov system $T_{n,\delta}$ will move the point $(x, y) \in [N_i, N_i + n_i) \times [0, n)$ to the position

$$T_{n,\delta}(x, y) = (q_i(x - N_i) + (y \bmod q_i), (y \text{ div } q_i) + N_i). \tag{2}$$

The restriction to integral stretching- and squeezing factors is necessary to keep resultant points at integer positions within the $n \times n$ grid. Use of the div (division of positive integers a and b delivering $\lfloor \frac{a}{b} \rfloor$) and mod (remainder when dividing positive integers) operation ensures that points in $n \times n$ are mapped onto each other in a bijective and reversible manner.

2.3 Important Properties

Kolmogorov systems tend to permute elements of the state space in a chaotic non-linear and apparently random fashion. After a sufficient number of iterations it becomes extremely hard for an observer to deduce the initial state of a Kolmogorov system from its final state. To be more specific, Kolmogorov systems offer very unique properties that are explained in more detail in the sequel.

Ergodicity. Ergodicity is important for a system that is to be applied in cryptography because it stands as a synonym for confusion. Informally speaking and expressed in terms of permutation systems, ergodicity stands for the property that almost any initial point will move to almost any other position in state space with equal probability as the system evolves in time. In other words there is no statistical way to predict the initial from the final position or vice versa.

Ergodicity of continuous Kolmogorov systems has been proven long ago [1]. As for discrete Kolmogorov systems, we have no knowledge that anyone has succeeded in defining them in a way such that ergodicity can be shown. In the sequel we derive necessary and sufficient conditions on the number of iterations necessary to ensure ergodicity of discrete Kolmogorov systems as introduced by equation 2. Note that this way a constructive proof of ergodicity is achieved.

In the following we restrict attention to the practically most relevant case of $n = p^m$ being an integral power of a prime p. The discrete Kolmogorov system T_{n,δ_r} is defined by the list $\delta_r = (n_{1r}, n_{2r}, \ldots, n_{k_r r})$ of length k_r containing the positive integers to be used as key in round r. As mentioned before there are the restrictions $1 \leq i \leq k_r$, $0 < n_{ir} < n$, $\sum_{i=1}^{k_r} n_{ir} = n$ and the constraint that all $n_{ir} \in \delta_r$ must partition the side length n.

Furthermore let the stretching and squeezing factors q_{ir} to use for vertical strip number i in round number r be defined by $q_{ir} = \frac{n}{n_{ir}}$. This results in quantities q_{ir}, $q_{ir} \geq p$ that also have to be integral powers of p because of the divisibility assumption made.

Consider an arbitrary point $(x, y) \in [N_{ir}, N_{ir} + n_{ir}) \times [0, n)$ in vertical strip number i to be transformed in round number r under the influence of the key δ_r (see equation 2 and figure 1). Coordinates x and y can then be expressed by q_{ir}-adic representations of length $t_{ir} = \lceil \log_{q_{ir}} n \rceil$ by $x = \sum_{j=1}^{t_{ir}} x_{jr} (q_{ir})^{t_{ir}-j}$ and $y = \sum_{j=1}^{t_{ir}} y_{jr} (q_{ir})^{t_{ir}-j}$. Similarly N_{ir} can be expanded according to $N_{ir} = \sum_{j=1}^{t_{ir}} Ni_{jr} (q_{ir})^{t_{ir}-j}$ and $x - N_{ir}$ may be expressed as $x - N_{ir} = \sum_{j=1}^{t_{ir}} xm_{jr}$ $(q_{ir})^{t_{ir}-j}$. Obviously x is the sum of $x - N_{ir}$ and N_{ir}.

To clarify these relations, the following illustration should be helpful. Please note that while in the representation of x the most significant position stands on the right side, the most significant position in the q_{ir}-adic representation of y is found on the left side. This arrangement has been made so that the subsequent transformation can essentially be depicted as a cyclic right shift by one position.

x

$x_{t_{ir}r}$	$\ldots$	x_{3r}	x_{2r}	x_{1r}

$xm_{t_{ir}r}$	$\ldots$	xm_{3r}	xm_{2r}	0
$Ni_{t_{ir}r}$	$\ldots$	Ni_{3r}	Ni_{2r}	Ni_{1r}

y

y_{1r}	y_{2r}	y_{3r}	$\ldots$	$y_{t_{ir}r}$

y_{1r}	y_{2r}	y_{3r}	$\ldots$	$y_{t_{ir}r}$
0	0	0	$\ldots$	0

According to equation 2 application of T_{n,δ_r} will move the point (x, y) to a new position $(x', y') = T_{n,\delta_r}(x, y)$ with coordinates $x' = q_{ir}(x - N_{ir}) + (y \bmod q_{ir})$ and $y' = (y \operatorname{div} q_{ir}) + N_{ir}$, as made clear by the subsequent figure.

x'

$y_{t_{ir}r}$	$\ldots$	xm_{4r}	xm_{3r}	xm_{2r}
0	$\ldots$	0	0	0

y'

0	y_{1r}	y_{2r}	$\ldots$	$y_{(t_{ir}-1)r}$
Ni_{1r}	Ni_{2r}	Ni_{3r}	$\ldots$	$Ni_{t_{ir}r}$

Suppose that lists δ_r are chosen independently and at random[1]. Neglecting the constraint $N_{ir} \leq x$ which follows from the fact that N_{ir} is the left border of the vertical strip containing the point (x, y) for a moment, the proof of ergodicity becomes straightforward. N_{ir} adds random q_{ir}-bits to all the q_{ir}-bits of y' yielding a random value for the new y-coordinate in one step. Cyclically shifting the least significant position of the y-coordinate to the least significant position in the x-coordinate and shifting these random q_{ir}-bits towards more significant positions in the x-coordinate ensures that after at most an additional $\max_{i=1}^{k_r} t_{ir} \leq m$ iterations the transformed point can move to almost any other position in state space with equal probability. Thus ergodicity is achieved after at most $m + 1$ iterations.

[1] This is a common assumption whenever proving specific properties of iterated cryptographic schemes. Round keys are generally supposed to be random and independent.

Now let us pay attention to the constraint $N_{ir} \leq x$. A moment of thought reveals that the worst non-trivial point that will need the largest number of rounds until being able to move to any position has a x-coordinate of 0 and a y-coordinate where just y_{1r} is different from zero. Then it takes at most $m+1$ iterations until the second-least significant q_{ir}-bit in the x-coordinate is set and the least significant q_{ir}-bit in N_{ir} (and also in the x-coordinate!) may assume any random value. By shifting q_{ir}-bits towards more significant positions in the x-coordinate every iteration causes one additional position in x to become random and by adding N_{ir} the same applies to the y-coordinate. This way it is guaranteed that after another at most $m-1$ iterations ergodicity is achieved after at most $2m$ steps in total.

Theorem 1. *Let the side-length $n = p^m$ be given as integral power of a prime p. Then the discrete Kolmogorov system T_{n,δ_r} as defined in equation 2 is ergodic provided that at least $2m$ iterations are performed and lists δ_r used in every step r are chosen independently and at random.*

In the discussion above we have noted that the restriction $N_{ir} \leq x$ to observe in every step significantly increases the number of iterations necessary until an initial point can move to any other position. Particularly points with small (zero) x-coordinate need a long time until exhibiting ergodic behaviour. However, a simple trick can help a lot in reducing the number of iterations necessary to achieve ergodicity of the underlying system: after every discrete Kolmogorov permutation round just apply a cyclic shift by $\frac{n}{2} - 1$ to the elements in the $n \times n$ array. This corresponds to adding $\frac{n}{2} - 1$ modulo n to every x-coordinate and helps points with initially small x-coordinates to move to any other position in a reduced number of rounds. Additionally this simple trick also solves the problems associated with the fixed points $(0, 0)$ and $(n-1, n-1)$ so that not just almost all points can move to almost any position but really all of the $n \times n$ points will have ergodic behaviour.

Exponential Divergence. Informally speaking and expressed in terms of permutation systems, exponential divergence implies that neighboring points contained in the same subspace of the state space (e.g. points of the same vertical strip corresponding to the same block of the defining partition) diverge at an exponential rate. This way even highly correlated points in input blocks will quickly loose correlations and structures present in input data will soon disappear.

Proving exponential divergence of specific discrete Kolmogorov systems can proceed using similar arguments as applied in proving ergodicity of discrete Kolmogorv systems. Specifically we derived the the following theorem [12].

Theorem 2. *Let the side-length $n = p^m$ be given as integral power of a prime p. Then the discrete Kolmogorov system T_{n,δ_r} as defined in equation 2 exhibits exponential divergence of points contained in the same blocks defined by partitions δ_r ensuring that after at most $2m-1$ iterations arbitrary non-zero deviations between initial points have propagated at least once to the most significant position in the x-coordinate.*

Mixing Property. Informally speaking and expressed in terms of permutation systems, fulfillment of the mixing property implies that any subspace of the state space will dissipate uniformly over the whole state space. Obviously this is an even stronger requirement than ergodicity because it does not only imply that almost any point will move to almost any position in state space with equal probability but also that distances between neighboring points within certain subspaces will become random as the system evolves in time.

Combining results derived in proving ergodicity and exponential divergence of discrete Kolmogorov systems, we have proven the following theorem [11].

Theorem 3. *Let the side-length $n = p^m$ be given as integral power of a prime p. Then the discrete Kolmogorov system T_{n,δ_r} as defined in equation 2 is mixing provided that at least $4m$ iterations are performed and lists δ_r used in every step r are chosen independently and at random.*

2.4 Analysis Summary

Summarizing the preceding discussion, a simple law on the conditions necessary to ensure that discrete Kolmogorov systems generate high-quality permutations can be stated as follows:

Theorem 4. *Let the side-length $n = p^m$ be given as integral power of a prime p. Then the discrete Kolmogorov system T_{n,δ_r} as defined in equation 2 fulfills the properties of ergodicity, exponential divergence and mixing provided that at least $4m$ iterations are performed and lists δ_r used in every step r are chosen independently and at random.*

Based on this theorem it is well justified to claim that the permutation operator developed in this contribution is indeed an excellent key-dependent permutation operator for cryptographic applications.

3 Applications

As shown in our analysis section, discrete chaotic Kolmogorov systems offer perfect permutation operators. In this section we would like to emphasize relevance of this analysis by giving several examples showing that discrete Kolmogorov systems can successfully be applied to many important problems encountered in communication security. Due to the limited space available, description must be restricted to just outlining some examples in symmetric encryption, secure hashing, password based access control, and digital image watermarking.

3.1 Efficient Block Ciphering

The structure of iterated symmetric product ciphers [13] which perform a blockwise encryption of the plaintext input to the system by repeated intertwined application of r round of permutations and substitutions can be observed from

Fig. 2. Input to the system is a block of plaintext and a pass-phrase. From this key the internal key management derives individual keys and supplies them to the various rounds. Every round applies one permutation and one substitution operation to the output of the previous round (initially the plaintext block). After r rounds, the output of the final round gives the ciphertext output by the r-round product cipher.

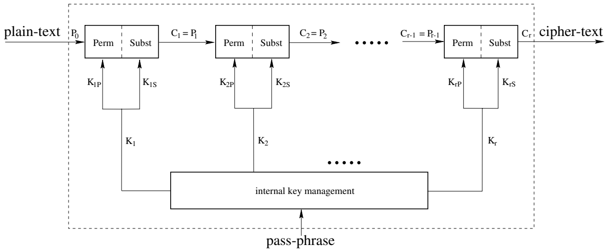

Fig. 2. Structure of an r-round product cipher

The role for discrete Kolmogorov systems within this framework is immediate to see. We use them as high-quality permutation operators for implementing the permutations needed. When complemented with an adequate substitution operator, this approach can deliver very strong and efficient ciphers. More details on that matter can e.g. be found in [10].

3.2 Cryptographic Message Digests

To provide integrity [8] and authenticity [7] in secure communications applications at reasonable computational costs, efficient and strong cryptographic hash functions are needed. Our approach to compute a message digest based on discrete chaotic Kolmogorov systems runs as follows.

First a 16×16 square array of bits is initialized with 256 pseudo-random bits (128 zeros, 128 ones) taken from the after-comma binary expansion of some "magic" constants (π, e, golden ratio ϕ, $\sqrt{2}$, $\sqrt{5}$, etc.) as done in almost any cryptographic hash function. Taken line-by-line or column-by-column, this provides the initial 256 bit message digest MD_0.

After initialization, in every step $t = 1, 2, \ldots$ the message digest MD_{t-1} is updated by processing the message in blocks W_t of 256 bit each. Since message lengths are usually not a multiple of 256, padding the last block with arbitrary constant bits may be necessary.

Now these 256 message bits are XORed with the current 256 bit message digest to obtain $X_t = W_t \oplus MD_{t-1}$. This step ensures that any block contains approximately an equal number of zeros and ones, regardless of the message block (which could be entirely zero etc.).

To maximize input avalanche effects, the 8 32-bit words $X_t(i)$ $(0 \leq i \leq 7)$ are processed according to a linear recurrence relation. First a forward dissipation

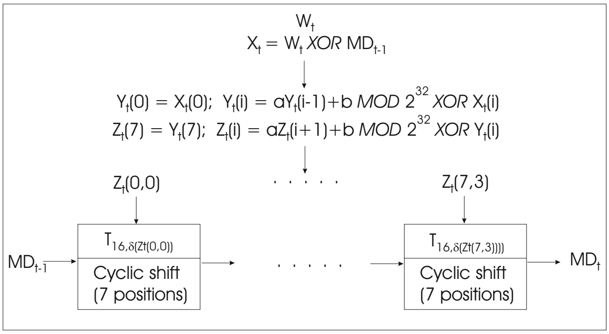
W_t
$X_t = W_t$ XOR MD_{t-1}
$Y_t(0) = X_t(0)$; $Y_t(i) = aY_t(i-1)+b$ MOD 2^{32} XOR $X_t(i)$
$Z_t(7) = Y_t(7)$; $Z_t(i) = aZ_t(i+1)+b$ MOD 2^{32} XOR $Y_t(i)$
$Z_t(0,0)$
. . . . .
$Z_t(7,3)$
MD_{t-1}
$T_{16,\delta(Zt(0,0))}$
Cyclic shift (7 positions)
. . . . . .
$T_{16,\delta(Zt(7,3))))}$
Cyclic shift (7 positions)
MD_t

Fig. 3. One step in calculating data dependent chaotic permutation hashes based on discrete Kolmogorov systems

step is done according to $Y_t(0) = X_t(0)$, $Y_t(i) = aY_t(i-1) + b \bmod 2^{32} \oplus X_t(i)$ with parameters a and b set accordingly (see e.g. [9] for a large variety of suitable parameter settings) to give pseudo-random sequences $Y_t(i)$. This is followed by a backward dissipation step (with index i decreasing) according to $Z_t(7) = Y_t(7)$, $Z_t(i) = aZ_t(i+1) + b \; mod \; 2^{32} \oplus Y_t(i)$.

After preprocessing the message block W_t to obtain the block Z_t, the actual hashing step takes place. The 256 bit of Z_t are used to provide 32 key bytes $Z_t(i,j)$ $(0 \leq i \leq 7,\ 0 \leq j \leq 3)$ to permute the message digest MD_{t-1} stored in the 16×16 array of bits using the corresponding discrete Kolmogorov system. Fig. 3 summarizes one round when calculating data dependent chaotic permutation hashes based on chaotic Kolmogorov systems. Iterating this procedure for all blocks of the input message and finally reading the 16×16 2D array line-by-line or column-by-column delivers the 256 bit message digest of the message to hash in a very efficient and elegant manner as pseudo-random message-dependent permutation of the initial message digest MD_0.

3.3 Digital Image Watermarking

Most of the commercially available systems for digital image watermarking [14] are based on ideas known from spread spectrum radio communications [2]. In spread spectrum communications, one transmits a narrowband signal over a much larger bandwidth such that the signal energy present in any single frequency is undetectable. This allows the signal reception even if there is interference on some frequencies.

Although there are many variants of spread-spectrum communications, we will focus on *Direct-Sequence Spread Spectrum* (DSSS) as the method most useful for application in digital image watermarking. A descriptive exposition how this can be achieved is found e.g. in [4] and similarly in [5]; to illustrate the principle we will closely follow along these lines.

Fig. 4 illustrates a simple, straightforward example of spread spectrum watermarking. The watermark bits (key2) to be embedded[2] are spread to fill an image of the same size as the image to be watermarked. The spread information bits are then modulated with a cryptographically secure PN signal keyed by watermarking key key1, scaled according to perceptual criteria, and added to the image in a pixel-wise fashion.

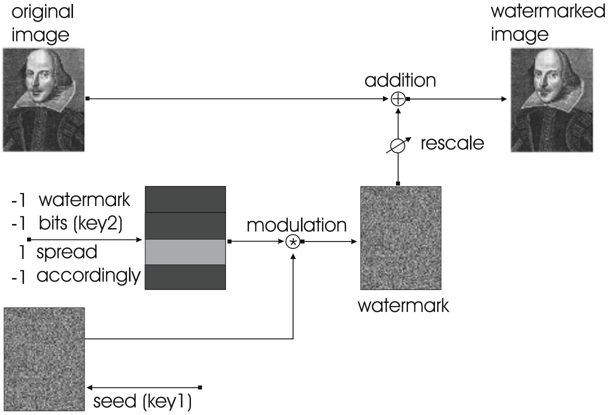

Fig. 4. Spread spectrum watermark embedding

Considering this practically most relevant approach for digital copyright protection via watermarking, an potential role for discrete Kolmogorov systems within the framework of DSSS watermarking becomes obvious. Security of any such DSSS watermarking scheme is heavily based on the cryptographically secure PN signal keyed by watermarking key key1. We implement this process as follows. Starting with a balanced initial binary image (might be a corporate logo), this image is permuted by discrete Kolmogorov systems under the influence of a key for as many rounds as are necessary to ensure that a high-quality PN signal is achieved. This PN signal is then used in the watermark embedding (and also detection) phase as depicted in Fig. 4, a fact that stresses the vital role that chaotic permutation operators can play in copyright protection via digital watermarking.

4 Conclusion

In this contribution we have shown that it is possible to derive adequate discrete counterparts for classical continuous Kolmogorov systems. Additionally we provided a detailed analysis under which criteria these novel discrete Kolmogorov systems offer high-quality permutation operators. Availability of such discrete permutation systems was finally utilized to sketch several examples of potential applications for important tasks in information security such as symmetric

[2] For simplicity, we just embed 4 bits; in real systems, 128 bit or more are used.

block ciphering, message digest computation, or copyright protection via digital watermarking. Summing up it can be concluded that our analysis performed for discrete Kolmogorov systems proves validity of specific important properties and constitutes a solid basis to apply them in many fields of secure communications.

References

1. V.I. Arnold and A. Avez. *Ergodic Problems of Classical Mechanics.* W.A. Benjamin, New York, 1968.
2. R. Dixon. *Spread Spectrum Systems.* John Wiley and Sons, New York, 1984.
3. S. Goldstein, B. Misra, and M. Courbage. On intrinsic randomness of dynamical systems. *Journal of Statistical Physics*, 25(1):111–126, 1981.
4. F. Hartung, J. Su, and B. Girod. Spread spectrum watermarking: Malicious attacks and counterattacks. In *Security and Watermarking of Multimedia Contents, Proc. SPIE 3657*, January 1999.
5. Martin Kutter. Performance improvement of spread spectrum based image watermarking schemes through m-ary modulation. In *Workshop on Information Hiding, Lecture Notes in Computer Science, volume 1768*, pages 238–250, 1999.
6. Jürgen Moser. *Stable and Random Motions in Dynamical Systems.* Princeton University Press, Princeton, 1973.
7. NIST. Digital signature standard. U. S. Department of Commerce, 1994.
8. NIST. Secure hash standard. FIPS PUB 180-1, April 1995.
9. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipies in C: The Art of Scientific Computing.* Cambridge University Press, 1988.
10. Josef Scharinger. Fast encryption of image data using chaotic Kolmogorov flows. *Journal of Electronic Imaging*, 7(2):318–325, 1998.
11. Josef Scharinger. Analysis of chaotic systems for communication security applications. In *Proceedings of the 14th International Conference on Systems Science*, volume 3, pages 78–85, 2001.
12. Josef Scharinger. Application of signed Kolmogorov hashes to provide integrity and authenticity in web-based software distribution. In *Formal Methods and Tools for Computer Science*, pages 85–88, 2001.
13. Bruce Schneier. *Applied Cryptography.* Addison-Wesley, 1996.
14. A.Z. Tirkel, G.A. Rankin, R.G. van Schyndel, W.J. Ho, N.R.A. Mee, and C.F. Osborne. Electronic watermark. In *Dicta-93*, pages 666–672, 1993.