Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps

Goce Jakimoski and Ljupčo Kocarev, *Senior Member, IEEE*

***Abstract*—This paper is devoted to the analysis of the impact of chaos-based techniques on block encryption ciphers. We present several chaos based ciphers. Using the well-known principles in the cryptanalysis we show that these ciphers do not behave worse than the standard ones, opening in this way a novel approach to the design of block encryption ciphers.**

***Index Terms*—Block encryption ciphers, chaos, cryptography, S-boxes.**

I. Introduction

IN THE last several years increasing efforts have been made to use chaotic systems for enhancing some features of communications systems. The highly unpredictable and random-look nature of chaotic signals is the most attractive feature of deterministic chaotic systems that may lead to novel (engineering) applications. Chaos and cryptography have some common features, the most prominent being sensitivity to variables' and parameters' changes. Shannon in his seminal paper [1] wrote: "In a good mixing transformation ... functions are complicated, involving all variables in a sensitive way. A small variation of any one (variable) changes (the outputs) considerably." An important difference between chaos and cryptography lies on the fact that systems used in chaos are defined only on real numbers [2], while cryptography deals with systems defined on finite number of integers [3]. Nevertheless, we believe that the two disciplines can benefit from each other. Thus, for example, as we show in this paper, new encryption algorithms can be derived from chaotic systems. On the other hand, chaos theory may also benefit from cryptography: new quantities and techniques for chaos analysis may be developed from cryptography.

The aim of this paper is to deal with chaotic systems and block encryption ciphers. Chaos has already been used to design cryptographic systems. An encryption algorithm that uses the iterations of the chaotic tent map is proposed in [4] and then generalized in [5]. Encryption algorithms based on multiple iteration of a certain dynamical chaotic system coming from gas dynamics models are presented in [6]. In [7] 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. In [8] the author encrypts each character of the message as the integer number of iterations performed in the logistic equation. While in conventional cryptographic ciphers the number of rounds (iterations) performed by an encryption transformation is usually less then 30, in [8] this number can be as large as 65536, and is always larger then 250. Another encryption algorithm based on synchronized chaotic systems is proposed in [9]. The authors suggest each byte (consists of n bits) of a message to correspond (to be encrypted) with a different chaotic attractor. In [10] the authors assume that the message to be sent is a binary file consisting of a chain of zeros and ones and the sender and the receiver has previously agreed to use the same d-dimensional chaotic dynamical rule, which generates sequences of real numbers by iterating it.

A common atribute to all chaos-based block encryption algorithms is that their security is not analyzed in terms of the techniques developed in cryptanalysis. For example, the encryption algorithm proposed in [4] is cryptanalyzed in [11], showing that the algrorithm can be broken using known-plaintext attack. We recently analyzed [12] the performance and security of chaos based encryption schemes proposed in [8]–[10]. The analysis shows that the encryption rates these algorithms offer are not competitive to the encryption rates of the standard cryptographic algorithms, and, furthermore, the algorithms can be easily broken using known-plaintext attacks.

In this paper we present several block encryption ciphers based on chaotic maps. Our approach differs from others in two ways. First, we use systematic procedure to create chaos based ciphers. Two well-known chaotic maps, exponential and logistic, defined on the unit interval by $x \rightarrow a^x \bmod 1$ and $x \rightarrow 4x(1-x)$, respectively, are used for this purpose. We show that with the proper choice of discretization and parameters, that may play role of the key, it is possible to design block encryption ciphers. Second, we cryptanalyze our ciphers, showing that they are resistant to known attacks.

This is the outline of the paper. In Section II we describe the general form of our block encryption algorithms. Section III explains some cryptographic tools that will be used in Section IV to find when a chaotic map may produce a cipher that has acceptable values of differential and linear approximation probabilities. In Section V we discuss different ways of using chaos based ciphers, and we close our paper with conclusion in Section VI.

II. Description of Block Encryption Algorithms

Recall first that the most encryption ciphers have the form

$$\begin{aligned} \boldsymbol{x}_0 &= B_0 \\ \boldsymbol{x}_i &= E_Z[\mathbf{x_{i-1}}], \qquad i = 1, \ldots, r \\ B_r &= \boldsymbol{x}_r \end{aligned} \tag{1}$$

Manuscript received December 8, 1999; revised July 26, 2000. This work was supported in part by the ARO under Grant DAAG55-98-1-0269, MURI Project "Digital Communication Devices based on Nonlinear Dynamics and Chaos," by the DOE under Grant DE-FG03-95ER14516, and by the ST Microelectronics. This paper was recommended by Associate Editor M. D. Bernardo.

The authors are with the Institute for Nonlinear Science, University of California, San Diego, La Jolla, CA 92093-0402 USA (e-mail: kocarev@heisenberg.ecsd.edu).

Publisher Item Identifier S 1057-7122(01)01397-6.

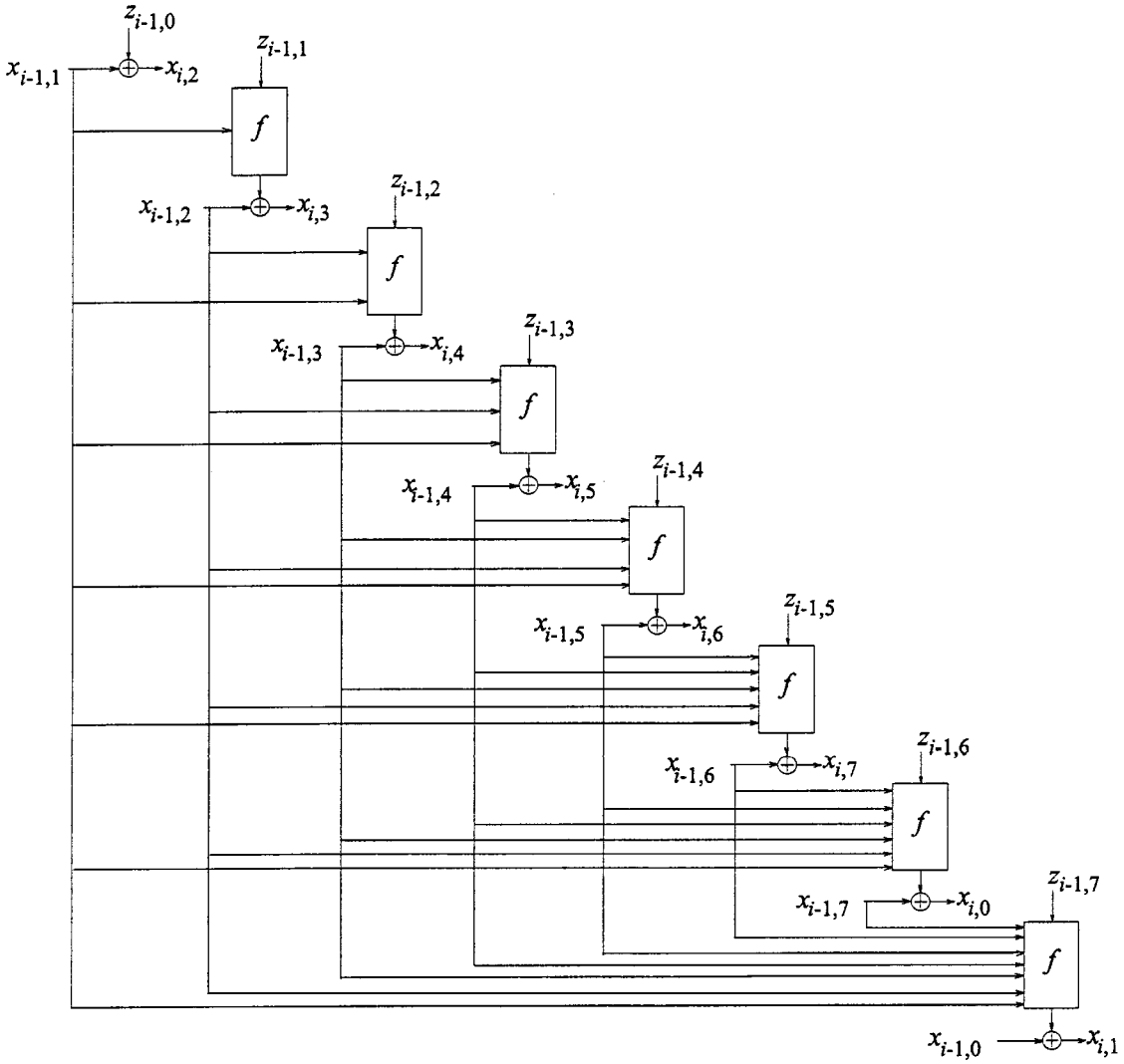

Fig. 1. Block diagram of encryption transformation (2).

where B_0, B_r are the plaintext and the cryptogram blocks with length L in bytes, respectively, $\boldsymbol{x}$ is an L dimensional vector, and E_Z is the key-dependent encryption transformation. A few classes of encryption transformations have been studied in the literature: Feistel networks [13], including DES [3], LOKI [14], CAST-128 [15], TWOFISH [16], unbalanced Feistel networks examples being MacGuffin [17] and BEAR/LION [18], and SP-networks (also called uniform transformations structures) such as IDEA [19] and SAFER [20].

In this paper we study a class of block encryption ciphers that can be described as follows. Let B_0 be a plaintext block of length 64 bits ($L = 8$ bytes). We write $x_{i,0}, \ldots, x_{i,7}$ for the eight bytes of the block B_i, $B_i = x_{i,0}, \ldots, x_{i,7}$. The cipher consists of r rounds of identical transformations applied in a sequence to the plaintext block. Encryption transformation is given with

$$x_{i,k+1} = x_{i-1,k} \oplus f_{k-1}[x_{i-1,1}, \ldots, x_{i-1,k-1}, z_{i-1,k-1}] \tag{2}$$

where $i = 1, \ldots, r$, $k = 1, \ldots, 8$, $f_0 = z_{i,0}$, $x_8 \equiv x_0$ and $x_9 \equiv x_1$, and $z_{i,0}, \ldots, z_{i,7}$ are the eight bytes of the subkey z_i which controls the ith round; see Fig. 1. The functions $f_1, \ldots, f_7$ have the following form:

$$f_j = f(x_1, \ldots, x_j, z_j)$$

where $j = 1, \ldots, 7$, and $f: M \to M, M = \{0, \ldots, 255\}$ is a map derived from a chaotic map. The output block $B_i = x_{i,0}, \ldots, x_{i,7}$ is input in the next round, except in the last round. Therefore, $B_r = x_{r,0}, \ldots, x_{r,7}$ is the ciphertext block (encrypted information). The length of the ciphertext block is 64 bits (8 bytes) and is equal to the length of the plaintext block. Each round i is controlled by one 8-byte subkey z_i. There are r subkeys totally and they are derived from the key in a procedure for generating round subkeys. In all examples we study bellow, f has the form of $f = g(x_1 \oplus x_2 \oplus \ldots \oplus x_j \oplus z_j)$ where g is obtained via discretization of a nonlinear map, with mixing property and robust chaos.

The decrypting structure undoes the transformations of the encrypting structure: r decryption rounds are applied to the ciphertext block B_r to produce the original plaintext block B_0. The round subkeys are applied now in a reverse order. The decryption round transformation is

$$x_{i-1,k} = x_{i,k+1} \oplus f_{k-1}[x_{i-1,1}, \ldots, x_{i-1,k-1}, z_{i-1,k-1}] \tag{3}$$

with $k = 1, \ldots, 8$, $f_0 = z_0$, $x_8 \equiv x_0$ and $x_9 \equiv x_1$.

III. Cryptanalysis

The central question in cryptography is what is security? This question can be answered at two different levels: theoretical and practical.

At theoretical level, the basic properties charactering a secure object are "randomness increasing" and "computationally unpredictable." By object we wean pseudo-random number generator, one-way function, or block encryption algorithm. It is well known that if one of the following objects exist: a secure pseudo-random number generator, a secure one-way function, and a secure block encryption algorithm, then all exist. Impagliazzo *et al.* [21] showed that secure pseudo random number generators (PRNG) exist if and only if secure one-way functions exist. Finally, the statement that secure PRNG's can be used to construct secure private-key crypto-systems and vice versa is proven in [22] and [23].

The rigorous definitions for "randomness-increasing" and "computationally unpredictable" are far beyond the scope of this paper and we refer the reader to [24]. The following informal definition of computationally unpredictable for pseudo-random number generators is due to Blum *et al.* [25]. We say that a pseudo-random number generator is *polynomial-time unpredictable* if and only if for every finite initial segment of sequence that has been produced by such generator, but with any element deleted from that segment, a probalistic Turing machine can, roughly speaking, do not better in guessing in polynomial time what the missing element is than by flipping a fair coin. Yao proved that a pseudo-random number generator is secure if and only if it is polynomial-time unpredictable.

The central unsolved question in the theory outlined above is whether a secure object exists. A major difficulty in settling the existence problem for this theory is summarized in the following heuristic unpredictability paradox [26]: *if a deterministic function is unpredictable, then it is difficult to prove anything about it, in particular, it is difficult to prove that is unpredictable*. Most of the results about unpredictability and cryptographic security follow from certain assumptions concerning the intractability of certain number-theoretical problems by probabilistic polynomial-time procedures. For example, the statement that the x^2 mod N generator is unpredictable is proven under so-called quadratic residuacity assumption; see [25] for details.

At the practical level cryptographic security of a cryptographic object (for example, a block encryption algorithm) can be checked up only by means of proving its resistance to various kind of known attacks. In this section we describe two basic attacks: differential [27] and linear cryptanalysis [28]. For extensions and generalizations of differential and linear cryptanalysis we refer the reader to [31]–[35].

A. Differential Cryptanalysis

Differential cryptanalysis [27]–[29] is a chosen-plaintext attack to find the secret key of an iterated cipher. It analyzes the effect of the "difference" of a pair of plaintexts on the "difference" of succeeding round outputs in an r-round iterated cipher.

An i-round differential is a couple (α, β), where α is the difference of a pair of distinct plaintexts B_0 and B_0^* and where β is a possible difference for the resulting ith outputs B_i and B_i^*. The probability of an i-round differential (α, β) is the conditional probability that β is the difference ΔB_i of the ciphertext pair after i rounds given that the plaintext pair has difference $\Delta B_0 = \alpha$ when the plaintexts and the round subkeys are independent and uniformly distributed.

The basic procedure of a differential attack on a r-round iterated cipher can be summarized as follows.

1) Find $(r-1)$-round differential (α, β) such that its probability is maximum, or nearly maximum.
2) Choose a plaintext B_0 uniformly at random and compute B_0^* so that the difference ΔB_0 is α. Submit B_0 and B_0^* for encryption under the actual key. From the resultant ciphertexts B_r and B_r^*, find every possible value (if any) of the last-round subkey z_r corresponding to the anticipated difference β. Add one to the count of the number of appearances of each such value of the last-round subkey.
3) Repeat Step 1 and Step 2 until some values of z_r are counted significantly more often than others. Take this most-often-counted subkey, or this small set of such subkeys, as the cryptanalyst's decision for the actual subkey z_r.

For the complexity (number of encryptions needed) of this attack holds

$$\mathrm{Comp}(r) \geq 2 \Big/ \left(p_{\max} - \frac{1}{2^m - 1} \right) \tag{4}$$

where $p_{\max} = \max_\alpha \max_\beta P(\Delta B_{r-1} = \beta | \Delta B_0 = \alpha)$ and m is the block length.

Usually the most difficult step in the attack procedure described above is the first step. When searching for $(r-1)$-round differential with maximum or nearly maximum probability, the attacker exploits some "weakness" of the nonlinear transformations used in the cipher. Thus the nonlinear maps should be chosen to have differential uniformity. The differential approximation probability of a given map f (DP_f for short) is a measure for differential uniformity and is defined as

$$DP_f = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X | f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right) \tag{5}$$

where X is the set of all possible input values and 2^n the number of its elements. Actually, DP_f is the maximum probability of having output difference Δy, when the input difference is Δx.

B. Linear Cryptanalysis

Linear cryptanalysis exploits a cipher's weakness expressed in terms of "linear expressions." In Matsui's terminology [30], a linear expression for one round is an "equation" for a certain modulo two sum of round input bits and round output bits as a sum of round key bits. The expression should be satisfied with probability much more (or much less) than 0.5 to be useful. A generalization of this idea [35] is using a more general notion of I/O sums.

An I/O sum $S^{(i)}$ for the ith round is a modulo-two sum of a balanced binary-valued function f_i of the round input B_{i-1} and

a balanced binary-valued function g_i of the round output B_i, that is

$$S^{(i)} := f_i(B_{i-1}) \oplus g_i(B_i) \tag{6}$$

where $\oplus = XOR$ denotes modulo-two addition and a balanced binary-valued function is defined as a function that takes on the value zero for exactly half of its arguments and the value one otherwise.

I/O sums for successive rounds are linked if the output function g_{i-1} of each round before the last coincides with the input function f_i of the following round. When ρ successive $S^{(i)}$ are linked, their sum

$$S^{(1,\dots,\rho)} := \bigoplus S^{(i)} = g_0(B_0) \oplus g_\rho(B_\rho) \tag{7}$$

is called a multi-round I/O sum.

The imbalance $I(V)$ of a binary-valued variable V is the nonnegative real number $|2P[V = 0] - 1|$. The imbalance is used as a measure for the "effectiveness" of an I/O sum. The average-key imbalance of the I/O sum $S^{(1,\dots,\rho)}$ is the expectation of the key dependent imbalances $I(S^{(1,\dots,\rho)}|z^{(1,\dots,\rho)})$ and is denoted as I(S(1, . . ., r)). An I/O sum is effective if it has a large average-key imbalance and is guaranteed if its average-key imbalance is one.

Assuming that the attacker has access to N plaintext/ciphertext pairs with uniformly randomly chosen plaintexts the basic procedure is as follows.

1) Find an effective I/O sum $S^{(1,\dots,r-1)}$.
2) Set up a counter $c[\tilde{z}_r]$ for each possible last-round key $\tilde{z}_r$ and initialize all counters to zero.
3) Choose a plaintext pair (B_0, B_r).
4) For each possible value $\tilde{z}_r$, evaluate $\tilde{B}_{r-1} = E^{-1}_{\tilde{z}_r}(B_r)$ and if $g_0(B_0) \oplus g_{r-1}(\tilde{B}_{r-1}) = 0$, increment $c[\tilde{z}_r]$ by 1.
5) Repeat Steps 3 and 4 for all N available plaintext/ciphertext pairs.
6) Output all keys $\tilde{z}_r$ that maximize $|c[\tilde{z}_r] - (N/2)|$ as candidates for the key actually used in the last round.

As in the differential cryptanalysis attack, the first step in this procedure is the most difficult one. The existence of an effective I/O sum depends on the characteristics of the nonlinear maps used in the cipher. The most commonly used characteristic, when talking about linear cryptanalysis, is the linear approximation probability (LP_f for short) and it is defined as

$$LP_f = \max_{a,b\neq 0} \left(\frac{\#\{x \in X | x \bullet a = f(x) \bullet b\} - 2^{n-1}}{2^{n-1}} \right)^2 \tag{8}$$

where $a \bullet b$ denotes the parity of bit-wise product of a and b, X is the set of all possible inputs and 2^n the number of its elements. The linear approximation probability is square of the maximal imbalance of the event: the parity of the input bits selected by the mask a is equal to the parity of the output bits selected by the mask b. Decreasing the LP_f yields to increasing the complexity of the linear cryptanalysis attack.

IV. Examples

In this section we design ciphers using chaotic maps. We choose two simple chaotic maps: quadratic (logistic)

$$\tilde{x}_{n+1} = 4\tilde{x}_n(1 - \tilde{x}_n) \tag{9}$$

and exponential

$$\tilde{x}_{n+1} = a^{\tilde{x}_n} (\mathrm{mod}\, 1) \tag{10}$$

where $\tilde{x} \in [0, 1]$ and $a > 1$. It is well known that both maps are chaotic.

A. Algorithm Based on Quadratic Function

We consider now the cipher (2) with the function f defined as

$$f(y_j) = \begin{cases} \text{floor } [y_j(256 - y_j)/64], & \text{if } \tilde{y}_j < 256 \\ 255, & \text{if } \tilde{y}_j = 256 \end{cases} \tag{11}$$

where $\tilde{y}_j = \text{floor}[y_j(256 - y_j)/64]$, and $y_j = x_1 \oplus x_2 \dots \oplus x_j \oplus z_j$. The transformation is obtained from the logistic map (9). In the first step, the logistic map is scaled so that input and output values of the new map are in the interval [0, 256]. The second step is discretization of the newly derived map.

The function f is not one-to-one mapping. There are distinct elements of the set $\{0, 1, \dots, 255\}$ that are mapped to the same value. Thus, the cardinality of the set of all possible output values is less than 256. For example, the number of elements that are mapped to the value 255 is 17. This property implies that, when the input values are uniformly distributed, the output values are not uniformly distributed, i.e., the function f, "spoils" the input uniform distribution. Actually, when all input values are equally likely, the probability of having output value 255 is 17/256. This is significantly greater than 1/256. We used this fact to amount a known plaintext attack. The complexity of the attack was not greater than 2^{29}, which is far below the complexity of the brute force attack.

The problem can be solved by using maps that produce one-to-one mappings after discretization or replacing the discretization procedure. Examples of both are given in the subsections that follow.

B. Algorithm Based on Exponential Function

Let us consider a function of the following form:

$$f(y_j) = \begin{cases} a^{y_j} \bmod 257, & \text{if } \tilde{y}_j < 256 \\ 0, & \text{if } \tilde{y}_j = 256 \end{cases} \tag{12}$$

where $\tilde{y}_j = a^{y_j} \bmod 257$, and $y_j = x_1 \oplus x_2 \oplus \dots \oplus x_j \oplus z_j$. This function is derived from (10) by extending the output range to the interval [0, 256] and discretization. a is chosen so that it is a natural number and a generator of the multiplicative group of nonzero elements of the Galois field of order 257. There are 128 different values of a. In this case the map performs one-to-one transformation.

We check the values of the differential approximation probability DP_f and the linear approximation probability LP_f for all possible values of a. The differential approximation probability

is $DP_f = 1/2$ for all a and it appears for $\Delta x = 128$ in (5). The minimal value of the linear approximation probability is $LP_f = 0.118164$. However, if we iterate the exponential function (12) two or three times, then $LP \leq 2^{-3}$ and $DP \leq 2^{-4}$ for all a.

C. Algorithm Based on Nth Iteration of the Logistic Map

In the previous example the discrete map was bijection due to the choice a to be a primitive element of the Galois field. In this example the one-to-one map is determined using discretization procedure that is different from the one used in the first example. The procedure is as follows.

1) Divide the phase space into $n+1$ equal volume regions. Assign the numbers $0, \ldots, n$ to the regions so that one number is assigned to exactly one region. If a point is in the region i we say that its magnitude is i.
2) Randomly choose one starting point from each region and determine its image after N iterations of a chaotic map.
3) Find the set S of starting points that have unique image. Choose a subset A that contains 256 elements of S and determine the set B of corresponding images.
4) Assign new magnitudes $0, \ldots, 255$ to the elements of A according to their old magnitudes. Do the same with the elements of B. If the new magnitude of the starting point in A is i and the new magnitude of its image is j, then we say that $f(i) = j$. The map f is one-to-one.

The finally constructed function depends on the way the magnitudes are assigned in the first step, the chaotic map that is iterated, the number of iterations, and the starting points. By changing any one we can change the function f. We stress that, if the cardinality of the set S is less than 256, the Step 3 is impossible. The number of regions is chosen so that the average number of starting points that have unique image is slightly greater than 256, when the chaotic map used in Step 2 is the logistic map.

Let us now assume that the chaotic map has uniformly distributed ergodic invariant measure and the number of regions in Step 1 is $n+1$. The probability that given image is an image of exactly one starting point is

$$\sum_{i=1}^{n} \frac{1}{n} \left[\frac{n}{n+1}\right]^n = \left[\frac{n}{n+1}\right]^n \to 1/e$$

when $n \to \infty$. Thus for large values of n the portion of images that correspond to exactly one starting point is $1/e$. If we want to construct a map f: $\{0, \ldots, m-1\} \to \{0, \ldots, m-1\}$ the number of regions should be slightly greater than me for large values of m.

Table I shows a function constructed using the previously described procedure. The numbering system used is hexadecimal. The chaotic map, which was used in Step 2, is the logistic map. We choose $N = 1000$ and $n = 767$. The cardinality of the set S is 259. The differential approximation probability of the function f is $2^{-5} < DP_f = 12/256 < 2^{-4}$ and the linear approximation probability is $LP_f = 2^{-4}$.

The encryption cipher (2) is a product encryption cipher, i.e., it achieves the desired confusion and diffusion through repeatedly applying the encryption round transformation to the 64-bit block of plaintext. The number of rounds needed depends on the nonlinear map used and the way it is involved in the cipher.

TABLE I
THE FUNCTION f OBTAINED FROM THE LOGISTIC MAP USING THE PROCEDURE DESCRIBED IN THE TEXT

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	60	c4	56	52	88	17	82	ac	28	96	4f	4a	ff	20	b5	6a
1	92	83	bc	a7	b2	9a	ee	70	35	e1	25	61	9d	a4	9c	47
2	b7	7d	2f	24	c7	7e	c5	c8	77	14	8d	cc	fd	8a	ef	36
3	76	2c	12	11	2a	29	a8	b8	22	84	c3	e9	e6	e2	15	57
4	e0	3c	69	ce	05	d4	cd	fa	30	f8	dd	75	cf	a0	0c	55
5	9f	41	f3	6f	ea	d2	a2	65	23	89	81	39	e4	93	ba	6b
6	a9	b0	1f	f7	34	43	1b	08	04	fc	0b	aa	73	94	eb	8e
7	c2	d6	53	48	18	27	8f	5b	5d	d0	ec	f4	f5	31	4b	ab
8	4e	97	79	bb	13	b6	5e	8b	10	50	49	1d	f6	99	00	68
9	3f	95	ad	e7	e8	87	8c	51	64	1e	d9	e5	5a	da	de	f0
a	0f	46	f1	1c	71	e3	09	a5	dc	9e	bf	40	80	3b	45	02
b	a6	42	d1	ed	d7	fe	16	9b	63	72	c0	78	b4	67	26	03
c	01	54	07	90	38	21	62	3d	d8	ca	7f	b1	0a	d5	44	a1
d	0d	c9	f2	2e	b9	59	6c	66	b3	74	32	bd	df	58	6d	37
e	3a	2d	db	6e	f9	1a	c6	06	5f	a3	2b	19	7c	fb	7b	af
f	be	0e	85	5c	33	7a	c1	4d	cb	86	91	4c	d3	ae	3e	98

The encryption round can be represented by a weighted directed graph G with set of vertices $\{0, 1, \ldots, 7\}$ corresponding to the eight input bytes. If the output byte j depends on the input byte i, then the edge (i, j) is an element of the set of edges of G. If the input byte i affects the output byte j after it is transformed by the function f, the weight of the edge (i, j) is 1. Otherwise, the weight of the edge (i, j) is 0. We define the distance $d_n(i, j)$ between the input byte i and the output byte j after n rounds as the maximal possible weight of the path with length n between the vertices i and j.

The encryption cipher (2) has the minimal distance $\min_{i,j} d_n(i, j) = n - 2$, when $n > 1$. For $n = 1$, the minimal distance is 0. We choose the number of rounds to be $r = 20$. If the attacker can unroll two rounds, the minimal distance would be 16. Thus, the imbalance of any linear expression is not greater than $LP_f^{16} = 2^{-64}$ and the linear cryptanalysis attack is impossible. Further, the encryption cipher is a Markov cipher [36] and every input bit will "pass through" at least 16 nonlinear transformations before affecting any output bit. Thus, we do not believe that differentials with high probability exist. Statistical tests showed that after $k = 4$ rounds the maximum probability $P(\Delta x_{k,j} | \Delta x_{0,i}, \Delta x_{0,m} = 0, m \in \{0, \ldots, 7\}, m \neq i)$ is $1.3 \cdot 2^{-8}$. Therefore, this probability rapidly approaches its uniform value 2^{-8}.

D. Key Schedule

The key schedule is the means by which the key bits are turned into round keys that the cipher can use. The mapping that performs each round i depends on the value of the round subkey z_i. The length of the round subkeys is 64 bits and they are derived from the 128-bit key K_0 in a procedure as follows.

We denote the bytes of the keys K_i by $K_{i,j}$, $j = 0, \ldots, 15$. The key generation procedure is given with

$$K_{i,k+1} = K_{i-1,k} \oplus f_{k-1}[K_{i-1,1}, \ldots, K_{i-1,k-1}, c_{k-1}]$$
$$z_i = RH(K_i) \tag{13}$$

where $i = 1, \ldots, r$, $k = 1, \ldots, 16$, $f_0 = c_0$, $K_{i,16} \equiv K_{i,0}$ and $K_{i,17} \equiv K_{i,1}$. $c_0, \ldots, c_{15}$ are 16 bytes of the constant c. The function RH assigns the 64-bit right half of the key K_i to the round subkey z_i.

The structure of the key generation procedure is similar to the encryption structure (2). The only difference is that the length of the block is 128 bits and the round subkeys are equal to the constant c. The value of the constant is $c = 45f83fd1e01a638099c1d2f74ac61d04_h$ and it is randomly chosen.

V. Using Chaos-Based Encryption Ciphers

A Feistel network is a method for transforming any function (usually called the F function) into a permutation. The fundamental building block of a Feistel network is the F function: a key-dependent mapping of an input string onto an output string. Each F function usually has two parts: linear and nonlinear. Nonlinear part of the F function is called S-box: it is a table-driven nonlinear substitution operation. Most common linear functions used in the Fiestel networks are MDS matrices [37] and/or pseudo-Hadamard transformations (PHT) [20]. A maximum distance separable (MDS) code over a field is a linear mapping from a field elements to b field elements, producing a composite vector of $a + b$ elements, with the property that the minimum number of nonzero elements in any non zero vector is at least $b+1$. Another mapping used to increase the difficulty for cryptanalysys is simple XORing the key material before the first round and after the last round (this technique is known as whitening [38]).

The ciphers we use here clearly belong to the class of Feistel networks. The function f in (2) plays role of the F function in the Fiestel networks. However, the functions f in (2) which are derived from chaotic maps, can also be used only as S boxes, nonlinear parts of the F function. In this paper we keep our presentation as much simpler as possible; thus, for example, in all examples we use $y_j = x_1 \oplus x_2 \oplus \ldots \oplus x_j \oplus z_j$. Instead, one can use, for example

$$y_j = ((x_1 \lll 3) \oplus (x_2 \lll 3) \ldots) \oplus x_j \oplus z_j$$

where $\lll 3$ denotes 3-bit left rotation. Although rotation has performance impact in software and hardware implementation of an cipher, and makes the cipher nonsymmetric, the rotation may increase the difficulty for cryptanalyses. Another extensions (generalizations) of our ciphers are also possible, including those with linear MDS and PHT functions.

We have found that for a given chaotic map and for a given discretization procedure, there exist more then one function f with good cryptographic properties (low values of DP and LP). As an example, we mention that the second or third iteration of the exponential function (12) generate 128 ciphers, for which $LP \leq 2^{-5}$ and $DP \leq 2^{-4}$. Thus, one may generalize the procedure for encryption in a way that the function f is key-dependent. For example, one can use the first seven bits of the key byte to determine the value of a in (12) while the last bit to determine how many times (two or three) the function f is iterated.

We have extensively cryptanalyzed the class of ciphers described in Sections IV-B and IV-C using second or third iteration of the exponential function (12) and $r = 20$ rounds. Conventional cryptanalysys allows an attacker to control both the plaintext and the ciphertext inputs into the cipher. Since the structure of the key generation procedure is similar to the encryption structure (2), we allow the attacker to control also the key schedule. This attack is known as related-key attack; our ciphers seem to be resistant to such attacks. Therefore, we conjecture that there exists no more efficient attack to our ciphers than brute force.

The ciphers we discuss here use blocks of length 64 bits. We also consider 128-bit block ciphers based on chaotic maps. Our preliminary results (not reported here) indicate that these ciphers have also good cryptographic properties and therefore may be used as encryption transformations.

One of the goals of the design of the block encryption cipher was its easy implementation in software and hardware. The cipher and the key schedule use only byte operations that can be implemented on various processors. These operations can be implemented in hardware as well. The map f in (2) can be realized with a byte-in byte-out look-up table. Finally we note that our ciphers can be used in all standard block-cipher chaining modes, as one-way hash functions and pseudo-random number generators.

VI. Conclusion

In this paper we have proposed a class of block encryption ciphers based on chaos, using two well-known chaotic maps: exponential and logistic. We have shown that these maps produce ciphers that have acceptable values of differntial and linear approximation probabilities. The ciphers use only byte operations that can be easily implemented on various processors and in hardware. As a result of extensive cryptanalysis we conjecture that there exists no more efficient attack to our ciphers than brute force.

The ciphers we have studied in this paper belong to the class of Feistel networks. An essential part of every Feistel network is an S-box: table-driven nonlinear substitution operation. S-boxes are created either randomly or algorithmically. Here we have proposed another way of creating S-boxes: by using chaotic maps. It turns out that very simple chaotic maps and very simple discretization procedure generate secure S-boxes, which is the opposite to the case of randomly constructed S-boxes: they are unlikely to be secure.[1]. Therefore, we suggest that maybe there exists more deeper connection between cryptography and chaos theory, yet to be discovered. This and other questions related to chaos and cryptography will be a subject to our future studies.

[1]For example, Khafre [38] uses S-boxes from the RAND tables [39] and it is vulnerable to differential cryptanalysis. Or, DES variants with random fixed S-boxes are very likely to be weak [40].

REFERENCES

[1] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656–715, 1949.

[2] J. Gickenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Berlin, Germany: Springer, 1983.

[3] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. New York: Wiley, 1996.

[4] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A secret key cryptosystem by iterating a chaotic map," in *Proc. Advances in Cryptology—EUROCRYPT'91*. Berlin, Germany: Springer-Verlag, 1991, pp. 127–140.

[5] Z. Kotulski and J. Szczepanski, "Discrete chaotic cryptography," *Ann. Phys.*, vol. 6, pp. 381–394, 1997.

[6] Z. Kotulski, J. Szczepanski, K. Grski, A. Paszkiewicz, and A. Zugaj, "Application of discrete chaotic dynamical systems in cryptography—DCC method," *Int. J. Bifurcation Chaos*, vol. 9, pp. 1121–1135, 1999.

[7] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, pp. 1259–1284, 1998.

[8] M. S. Baptista, "Cryptography with chaos," *Phys. Lett. A*, vol. 240, pp. 50–54, 1998.

[9] Y. H. Chu and S. Chang, "Dynamical cryptography based on synchronized chaotic systems," *Electron. Lett.*, vol. 35, pp. 974–975, 1999.

[10] E. Alvarez, A. Fernandez, P. Garcia, J. Jimenez, and A. Marcano, "New approach to chaotic encryption," *Phys. Lett. A*, pp. 373–375, 1999.

[11] E. Biham, "Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91," in *Proc. Advances in Cryptology—EUROCRYPT'91*. Berlin, Germany: Springer-Verlag, 1991, pp. 532–534.

[12] G. Jakimoski and L. Kocarev, "Analysis of some recently proposed chaos-based encryption algorithms," submitted for publication.

[13] H. Feistel, "Cryptography and computer privacy," *Scientific American*, vol. 228, no. 5, pp. 15–33, 1973.

[14] L. Brown, J. Pieprzyk, and J. Seberry, "LOKI: A cryptographic primitive for authentication and secrecy applications," in *Proc. Advances in Cryptology—AUSCRYPT'90*. Berlin, Germany: Springer-Verlag, 1990, pp. 229–236.

[15] C. Adams, "Constructing symmetric ciphers using the CAST design procedure," *Designs, Codes and Cryptography*, vol. 12, pp. 71–104, 1997.

[16] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Freguson. Twofish: A 128-bit block cipher. [Online]. Available: http://www.counterpane.com/twofish.html

[17] M. Blaze and B. Schneier, "The MacGuffin block cipher algorithm," in *Fast Software Encryption Second Int. Workshop Proc.*. Berlin, Germany: Springer-Verlag, 1995, pp. 97–110.

[18] R. Anderson and E. Biham, "Two practical and provably secure block ciphers: BEAR and LION," in *Fast Software Encryption, Third Int. Workshop Proc*. Berlin, Germany: Springer-Verlag, 1996, pp. 113–120.

[19] X. Lai and J. L. Massey, "A proposal for a new block encryption standard," in *Advances in Cryptology—EUROCRYPT'90*. Berlin: Springer-Verlag, 1991, pp. 389–404.

[20] J. L. Massey, "SAFER K-64: A byte oriented block-ciphering algorithm," in *Fast Software Encryption*, R. Anderson, Ed. Berlin, Germany: Springer, 1993, (LNCS 809), pp. 1–17.

[21] R. Impagliazzo, L. Levin, and M. Luby, "Pseudo-random number generation from one-way functions," in *Proc. 21st Annu. Symp. Theory Computing*, 1989, pp. 12–24.

[22] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM J. Comput.*, vol. 17, pp. 373–386, 1988.

[23] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity-based cryptography," in *Proc. 30th Annu. Symp. Foundations Computer Science*, 1989, pp. 230–235.

[24] A. Yao, "Theory and applications of trapdoor functions," in *IEEE 23rd Symp. Foundations Computer Science*, 1982, pp. 80–91.

[25] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM J. Comput.*, vol. 15, pp. 364–383, 1986.

[26] J. C. Largaris, "Pseudo-random numbers," in *Probability and Algorithms*. Washington, DC: National Academy, 1992, pp. 65–85.

[27] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," in *Advances in Cryptology—CRYPTO'90*. Berlin, Germany: Springer-Verlag, 1991, pp. 2–21.

[28] ——, "Differential cryptanalysis of FEAL and N-Hash," in *Advances in Cryptology—EUROCRYPT '91*. Berlin, Germany: Springer-Verlag, 1991, pp. 1–16.

[29] ——, "Differential cryptanalysis of the full 16-round DES," in *Advances in Cryptology—CRYPTO'92*. Berlin, Germany: Springer-Verlag, 1993.

[30] M. Matsui, "Linear cryptanalysis method for DES ciphers," in *Advances in Cryptology—EUROCRYPT'93*. Berlin, Germany: Springer-Verlag, 1994, pp. 386–397.

[31] X. Lai, "Higher order derivations and differential cryptanalysis," in *Communication and Cryptography: Two Sides of One Tapestry*. Norwell, MA: Kluwer, 1994, pp. 227–233.

[32] B. Kaliski, Jr. and M. Robshaw, "Linear cryptoanalysis using multiple approximations," in *Advances in Cryptology—CRYPTO '94*. Berlin, Germany: Springer-Verlag, 1994, pp. 26–39.

[33] L. Knudsen and M. Robshaw, "Non-linear approximations in linear cryptanalysis," in *Advances in Cryptology—EUROCRYPT '96*. Berlin, Germany: Springer-Verlag, 1996, pp. 224–236.

[34] S. Langford and M. Hellman, "Differential-linear cryptanalysis," in *Advances in Cryptology—CRYPTO '94*. Berlin, Germany: Springer-Verlag, 1994, pp. 17–26.

[35] C. Harpes, G. G. Kramer, and J. L. Massey, "A generalization of linear cryptanalysis and the applicability of Matsui's pilling-up lemma," in *Advances in Cryptology—EUROCRYPT '95*. Berlin, Germany: Springer-Verlag, 1995, pp. 24–38.

[36] X. Lai, J. L. Massey, and S. Murphy, "Markov ciphers and differential cryptanalysis," in *Advances in Cryptology—EUROCRYPT '91*. Berlin, Germany: Springer-Verlag, 1991, pp. 17–38.

[37] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[38] R. C. Merkle, "Fast software encryption functions," in *Advances in Cryptology—CRYPTO '90*. Berlin, Germany: Springer-Verlag, 1991, pp. 476–501.

[39] RAND Corporation, *A Million Random Digits with 100,000 Normal Deviates*. Glencoe, IL: Free Press, 1955.

[40] E. Biham and A. Shamir, *Differential Cryptanalysis of Data Encryption Standard*. Berlin, Germany: Springer-Verlag, 1993.

Goce Jakimoski was born in Ohrid, Macedonia, in 1971. He received the B.S. degree in electrical engineering from Sts Cyril and Methodius University, Skopje, Macedonia, in 1995, and the M.S. degree in electrical engineering from the same University in 1998.

His research interests involve symetric key encryption schemes.

Ljupco Kocarev (SM'95) is an Associate Research Scientist at the Institute for Nonlinear Science at UCSD. His scientific interests include nonlinear science and its application to physics, biology and electrical engineering. He has authored or co-authored more than 60 journal articles in various international journals, including *Chaos: An Interdisciplinary Journal of Nonlinear Science*; *Chaos, Solitons, and Fractals*; *Geophysical Research Letters*; *International Journal of Bifurcation and Chaos*; *International Journal of Circuit Theory and Application*; IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART I: FUNDAMENTAL THEORY AND APPLICATIONS; IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART II: ANALOG AND DIGITAL SIGNAL PROCESSING; IE-ICE TRANSACTIONS ON FUNDAMENTALS AND ELECTRONICS, COMMUNICATIONS AND COMPUTER SCIENCE; *Journal of Applied Mathematics and Mechanics*; *Journal of Circuits, Systems, and Computers*; *Journal of Physics A: Mathematical and General Physics*; *Journal of the Franklin Institute*; *Physica D*; *Physical Review E*; *Physical Review Letters*; and *Physics Letters A*.