Chaos, Solitons and Fractals 42 (2009) 2312–2321

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

Cryptography with chaos and shadowing

Nejib Smaoui *, Ali Kanso

Department of Mathematics and Computer Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait

ARTICLE INFO

Article history:
Accepted 30 March 2009

ABSTRACT

In this paper, we present a novel approach to encrypt a message (a text composed by some alphabets) using chaos and shadowing. First, we generate a numerical chaotic orbit based on the logistic map, and use the shadowing algorithm of Smaoui and Kostelich [Smaoui N, Kostelich E. Using chaos to shadow the quadratic map for all time. Int J Comput Math 1998;70:117–29] to show that there exists a finite number of true orbits that shadow the numerical orbit. Then, the finite number of maps generated is used in Baptista's algorithm [Baptista MS. Cryptography with chaos. Phys Lett A 1998;240:50-4] to encrypt each character of the message. It is shown that the use of chaos and shadowing in the encryption process enhances the security level.

1. Introduction

The field of modern cryptography deals with more than hiding information, it encompasses a wide range of problems such as data encryption, data integrity, entity authentication, digital signature, key distribution, etc. However, the most basic problem in the field of cryptography is the classical one of maintaining the confidentiality of data transmitted over a public channel. Before late 1970s the use of cryptography to securely communicate over a public channel was totally restricted to the military and governmental sectors. After that, cryptography has been made available to the general public and it became widely used in many commercial and financial sectors which led to a growing interest in this field. Nowadays, cryptography is widely used in many areas including data storage in disks, internet banking, shopping over the internet, protection of communications channels, etc.

A cryptosystem is a procedure that transforms a readable message M, known as plaintext, into a scrambled message symbol C, known as ciphertext, in an invertible way. Cryptosystems are divided into symmetric or private key and asymmetric or public key. In asymmetric key cryptosystems, the sender uses some publicly known information (revealed by the receiver), known as the encryption key, to transform the plaintext M into the ciphertext C. This process is known as the encryption process. The sender then transmits the ciphertext C over a public channel to the receiver. At the receiver's end, the receiver uses some secret information, known as decryption key, that is mathematically related to (but computationally infeasible to obtain from) the encryption key to recover back the plaintext M. This process is known as the decryption process. The security of most of these cryptosystems depends on various unsolved mathematical problems including, factorization and discrete logarithm problems. RSA and El-Gammal cryptosystems [1] are some well-known approaches of asymmetric key cryptosystems, they are secure as long as the factorization and discrete logarithm problems, respectively, remain unsolved. In symmetric key cryptosystems, the sender and receiver must agree prior to the exchange of the ciphertext on some information, known as private key, that it is used by the two parties for both encryption and decryption. The communicating parties must keep this key secret from potential eavesdroppers to maintain private communications. Furthermore, symmetric key cryptosystems are divided into block ciphers and stream ciphers. A block cipher operates on large blocks of plaintext and encrypt each block as a single unit. Whereas a stream cipher operates on a continuous stream, and usually involves a

* Corresponding author.
E-mail addresses: nsmaoui64@yahoo.com (N. Smaoui), akanso@hotmail.com (A. Kanso).

0960-0779/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chaos.2009.03.128

pseudo-random number generator. Symmetric key cryptosystems are much faster than the asymmetric ones, and the key distribution in symmetric key cryptosystems can be resolved by using a secure channel or an asymmetric cryptosystem to exchange secret keys between the sender and the receiver. Well-known symmetric cryptosystems include the block ciphers DES and AES, and the stream ciphers shrinking generator and alternating step generator [1]. Most of today's cryptosystems rely on tools that use algebra, number theory, combinatorics, chaos, etc.

During the last two decades, many scientist, engineers and cryptographers have been attracted by the theory of chaos to develop an efficient cryptosystem. The possibility of implementing such theory in cryptographic applications was first reported in [2–5]. Most existing chaotic cryptosystems fall into the category of symmetric key. These cryptosystems are based on the high sensitivity of chaotic maps to their initial conditions and control parameters. In recent years, several chaotic cryptosystems have appeared in the literature, including those based on chaotic synchronization [6], and those without chaotic synchronization [7–26]. Based on the insecurity of cryptosystems with chaotic synchronization [27,28], the idea of constructing cryptosystems without synchronization has been given much more attention in the past few years.

In 1998, Baptista [8] proposed an attractive chaotic cryptosystem without synchronization governed by the one-dimensional chaotic logistic map

$$x_{n+1} = bx_n(1 - x_n) \tag{1}$$

in which the initial condition and parameter of the chaotic map are used as the secret key. After its publications, several variants of Baptista's cryptosystem have been proposed [29–34]. At the same time, a number of cryptanalytic attacks on Baptista's cryptosystem and its variants have also been proposed [35–39]. In Baptista's cryptosystem, each message block (character) is encrypted as the number of iterations applied to the logistic map to reach the region (ϵ-interval) that is associated with that block. The resultant ciphertext to be transmitted over a public channel is simply a sequence of integers corresponding to the number of iteration for each character in the message. Since the initial condition and parameter are shared between the communicating parties, the receiver can recover the original message by iterating the logistic map according to the sequence of integers. But during the encryption or decryption process, the number of iteration applied to the logistic map may involve thousands if not millions of iterates, therefore, one questions whether the orbits generated by the logistic map are chaotic or noise driven by computer round-off error.

Due to the different platforms used by the sender to encrypt a plaintext and the receiver to decrypt a ciphertext, and due to the sensitivity of the initial conditions of chaotic attractors, we present the idea of shadowing [40–49] to our cryptosystems. The question whether a true orbit shadows a numerical orbit has long been posed by Anosov [45] and Bowen [46]. In [45,46], it was shown that true orbits of systems which are uniformly hyperbolic shadow noisy orbits for all time (i.e., numerical orbits will stay close to true orbits for all time). The results obtained in [45,46] cannot be applied on chaotic attractors of the logistic map since it is of a non-hyperbolic nature.

Nusse and York [48] and Hammel et al. [40] investigated the shadowing problem of non-hyperbolic systems and showed that for the logistic map and under some conditions, true orbits can shadow numerical orbits for a long time. Hammel's computer-aided method in determining how long and how close a true orbit can shadow a noisy orbit works for some parameters of the logistic map and fails for others. Smaoui and Kostelich [49] modified Hammel's method by introducing a small noise on the control parameter b to show that a true chaotic orbit of the logistic map can shadow a noisy orbit for all time.

To the best of the authors knowledge, the idea of shadowing is introduced here for the first time in the area of cryptography. First, a numerical orbit $\{p_i\}_{i=0}^{N}$ of the chaotic attractor of the logistic map for a given parameter b and a given initial condition x_0 is computed. Then, based on the shadowing method, true orbits $\{x_i\}_{i=0}^{N}$ that shadow the numerical orbit $\{p_i\}_{i=0}^{N}$ for all time are found. The true orbits are generated by a finite number of maps to shadow the numerical orbit of a given chaotic attractor of the logistic map for a given parameter b and a given initial condition x_0. That is,

$$\{x_i\}_{i=0}^{N} = \bigcup_{j=1}^{m} \{x_{ij}\}_{i=0}^{k_j-1} \approx \{p_i\}_{i=0}^{N} \quad \text{where} \quad \sum_{j=1}^{m} k_j = N.$$

The true orbits $\{x_{ij}\}_{i=0}^{k_j-1}$ are computed using slightly different maps:

$$x_{n+1} = f_{b_j}(x_n) = b_j x_n(1 - x_n), \quad \text{where} \quad \max_{1 \leqslant j \leqslant m}(b_j - b) < \sqrt{\delta_p},$$

with m being the number of maps used and $\delta_p \approx 2 \cdot 10^{-8}$.

Using the finite number of one-dimensional logistic maps as shown above, we can quickly and safely encrypt information for further transmission.

The paper is organized as follows: In Section 2, we present the method of shadowing to the logistic map. Section 3 describes Baptista's cryptosystem and how it can be connected to the idea of shadowing. Section 4 illustrates through examples Baptista's method with and without shadowing, and we conclude in Section 5.

2. Shadowing the logistic map

The dynamics of the logistic map is obtained by iterating Eq. (1) for a fixed parameter b. If we restrict the bifurcation parameter b to any value in [0,4], then one can plot the system's attractor of the logistic map as a function of b. Fig. 1 presents

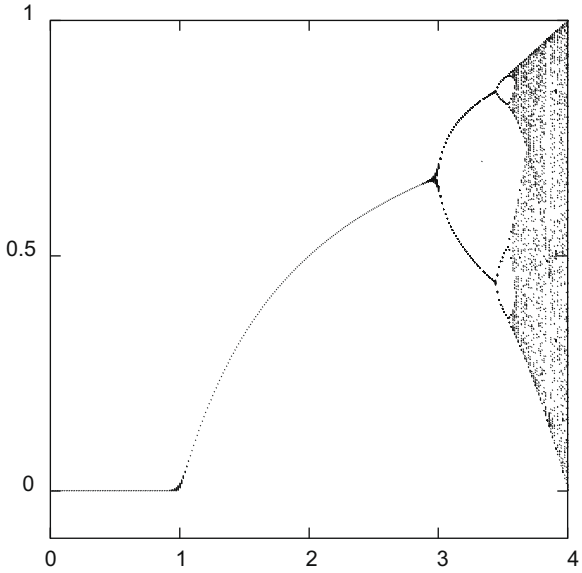

Fig. 1. The bifurcation diagram for the logistic map.

the bifurcation diagram of the logistic map for different values of b. When $0 \leqslant b \leqslant 1$, the trivial solution is the only fixed point. For $1 \leqslant b \leqslant 3$, a nontrivial fixed point appears. As b gets larger than 3, say $b = 3.2$, the attractor is a period-2 cycle which oscillates between two nontrivial fixed points. As b is increased further, both branches bifurcate, yielding a period-4 cycle, a period-8, a period-16, and a period-32 cycles. This phenomenon is known as the period-doubling bifurcation. For the values of b between 3.57 and 4, the map becomes chaotic. Fig. 2 is a blow up of Fig. 1 for the values of b in the chaotic region between 3.8 and 3.9.

Since a great deal of research on chaotic processes relies heavily on computer simulations, and since the study of these chaotic processes involve thousands if not millions of iterates, therefore one questions whether the orbit of a chaotic process is real or it is due to computer round-off error introduced as noise into the system. Suppose that the truncation error causes errors of order 10^{-8} for processes involving quantities of order 1, and assuming that the distance between two trajectories doubles at each iteration for a given chaotic process, then two trajectories starting 10^{-8} apart will be 1 unit a part in less than 50 iterations. As a result, the relation between the computer-generated chaotic trajectory (noisy orbit) and a true trajectory is no longer clear.

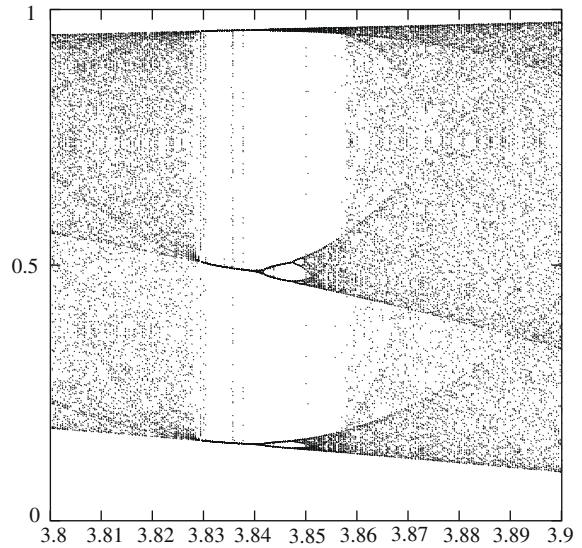

Fig. 2. A blow up of Fig. 1 for the values of $3.8 \leqslant b \leqslant 3.9$.

Fig. 3 depicts the time series plots of the first 200 iterates of the logistic map Eq. (1) corresponding to three different cases: case (A) for $x_0 = 0.1$ and $b = 3.86$; case (B) for $x_0 = 0.1$ and $b = 3.860337$, and case (C) for $x_0 = 0.100337$ and $b = 3.86$. Despite the similarities in the initial conditions, and/or in the bifurcation parameter b, a visual inspection of the time series plots in Fig. 3 indicates the difference in their qualitative behavior. Fig. 4 presents the first 50 iterates of the map as well as the 51st to 200th iterates for the three cases (A), (B), and (C) above. Points falling on the 45 deg line in these plots suggest that the generated orbits are very close. One can easily observe that quite early in time, the three series predict each other reasonably well. However, as time increases, the series deteriorates from each other.

Definition 1. A true orbit $\{x_n\}_{n=0}^{N}$ is an orbit free of noise.

Definition 2. $\{p_n\}_{n=0}^{N}$ is a δ_p-noisy orbit for the logistic map if

$$|p_{n+1} - bp_n(1 - p_n)| < \delta_p; \quad 0 \leqslant n \leqslant N.$$

A true orbit is said to shadow a noisy orbit, if each point of the noisy orbit is near a corresponding point of the true orbit.

Definition 3. The true orbit $\{x_n\}_{n=0}^{N}$ δ_x-shadows the noisy orbit $\{p_n\}_{n=0}^{N}$ if

$$|x_n - p_n| < \delta_x; \quad 0 \leqslant n \leqslant N.$$

Anosov [45] and Bowen [46] showed that if f is hyperbolic, then noisy orbits are shadowed by true orbits for all time.

Lemma 1 (Anosov, Bowen). *If f is hyperbolic, then for every $\delta_x > 0$ there is a $\delta_p > 0$ such that every δ_p-noisy orbit for f is δ_x-shadowed.*

The above shadowing Lemma of Anosov and Bowen is not valid for chaotic processes that are not hyperbolic. Coven [47] and Nusse and York [48] demonstrated that for some nonlinear maps and under some conditions true orbits of non-hyperbolic chaotic attractors can shadow numerical orbits. In 1987, Hammel et al. [40,41] showed through a computer-aided method that the numerical orbit of a chaotic attractor of the logistic map can be shadowed by a true orbit. The Cray X-MP machine was used in Hammel's numerical method.

Theorem 1 (Hammel). *For $N = 10^7$, the noisy orbit $\{p_n\}_{n=0}^{N}$ with $b = 3.8$ and $p_0 = 0.4$ is δ_x-shadowed by a true orbit $\{x_n\}_{n=0}^{N}$ within $\delta_x = 10^{-8}$.*

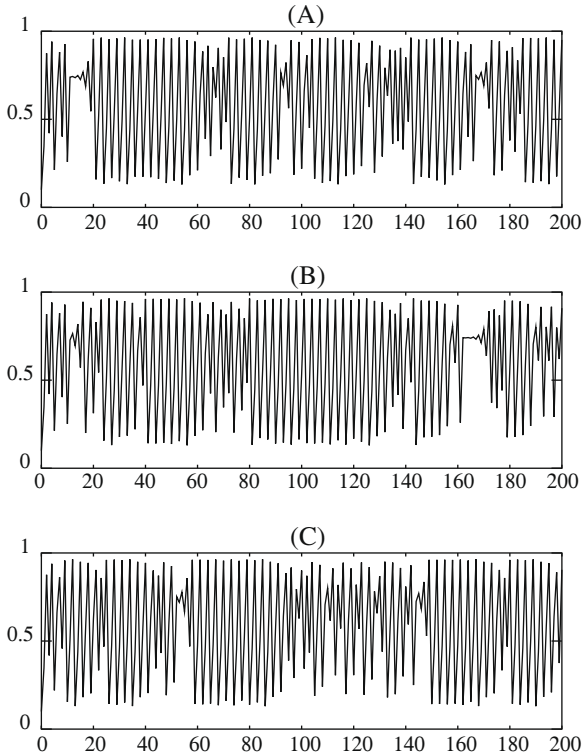

Fig. 3. Time series plots for the first 200 iterates of the logistic map for three different cases: (A) $x_0 = 0.1$ and $b = 3.86$; (B) $x_0 = 0.1$ and $b = 3.860337$; (C) $x_0 = 0.100337$ and $b = 3.86$.

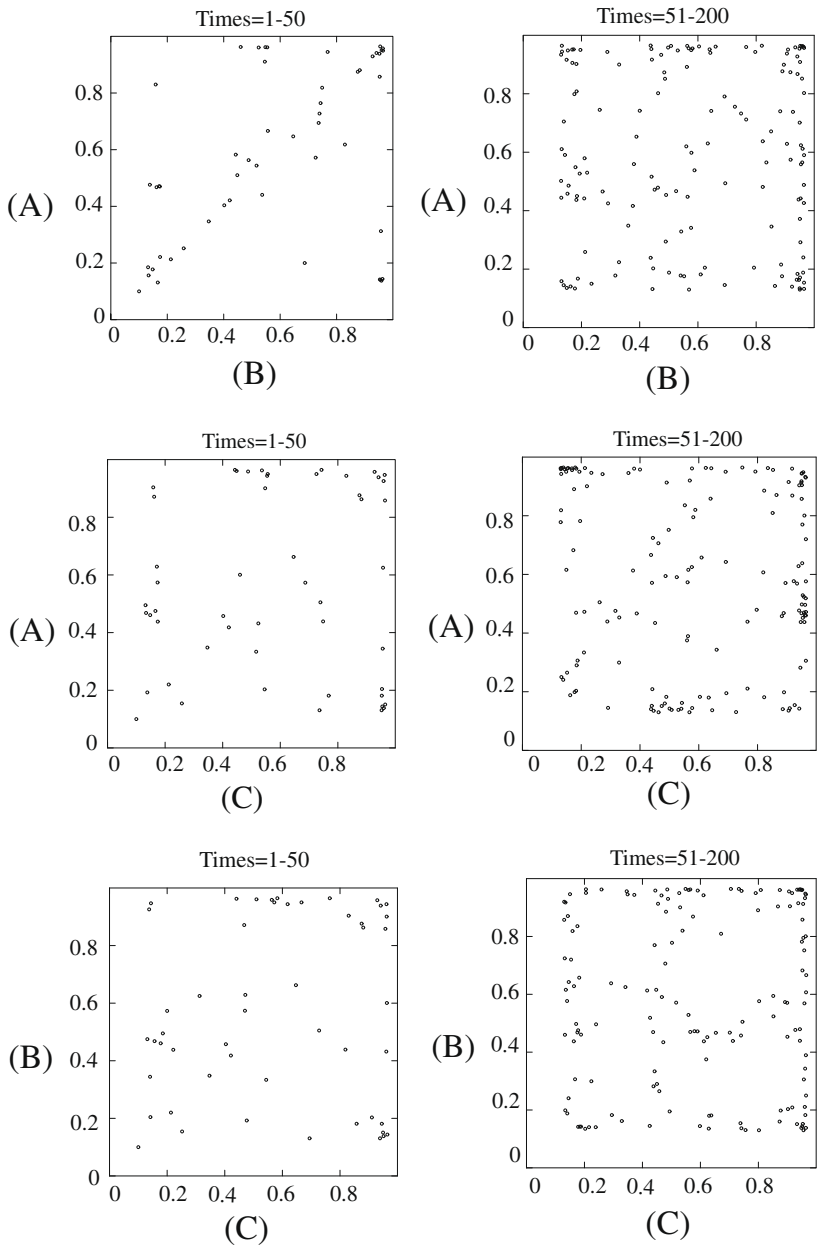

Fig. 4. Scatterplots matched by time: (A) $x_0 = 0.1$ and $b = 3.86$; (B) $x_0 = 0.1$ and $b = 3.860337$; (C) $x_0 = 0.100337$ and $b = 3.86$.

The shadowing technique used in Hammel's method to determine how long and how close can true orbits shadow numerical orbits of the non-hyperbolic attractor of the logistic map works for some parameters b of the logistic map and fails for others.

Smaoui and Kostelich [49] showed that by introducing a small noise on the control parameter b, a true orbit can shadow a noisy orbit for all time. At each time when Hammel's shadowing method fails, b is perturbed by $\sqrt{\delta_p}$ and the process is continued. In this way, the logistic map will be presented as a finite number of maps where the control parameter for each of these maps differ only by $\sqrt{\delta_p}$.

Theorem 2 (Smaoui and Kostelich). *For $N = 10^7$, the noisy orbit $\{p_n\}_{n=0}^{N}$ with $b = 3.86$ and $p_0 = 0.4$ is δ_x-shadowed by a true orbit $\{x_n\}_{n=0}^{N}$ within $\delta_x = 10^{-3}$, where*

$$\{x_n\}_{n=0}^{N} = \bigcup_{j=1}^{m=2} \{x_{ij}\}_{i=0}^{k_j-1}, \quad \text{and} \quad \sum_{j=1}^{m=2} k_j = N.$$

Table 1 shows that for the case $b = 3.86$ and for different initial conditions p_0, two maps were needed to shadow the numerical orbit $\{p_n\}_{n=0}^{N=10^7}$. The maximum variation of the parameter b (i.e., $\delta_b = b_1 - b_2$) is in the order of 10^{-4}.

Using the above shadowing technique, the simple one-dimensional logistic map for the case $b = 3.86$ in Eq. (1) can then be rewritten as follows:

$$x_{n+1} = \begin{cases} b_1 x_n(1 - x_n), & \text{if } n \leqslant k_1 \\ b_2 x_n(1 - x_n), & \text{if } k_1 < n \leqslant N. \end{cases} \tag{2}$$

Remark. In general, the simple one-dimensional logistic map can be written as follows:

$$x_{n+1} = \begin{cases} b_1 x_n(1 - x_n), & \text{if } n \leqslant k_1 \\ b_2 x_n(1 - x_n), & \text{if } k_1 < n \leqslant k_2 \\ \quad \vdots & \\ b_m x_n(1 - x_n), & \text{if } k_{m-1} < n \leqslant N. \end{cases} \tag{3}$$

In the next section, we present Baptista's encryption algorithm with and without shadowing. In Baptista's algorithm, the encryption of a character contained in a message to be transmitted is the number of iterations applied in a chaotic map to make its trajectory reaches an ϵ-interval associated with that character as described by Baptista [8]. However, instead of using the logistic Eq. (1) for iteration, we use Eq. (3) found from the shadowing technique described above.

3. Implementation of Baptista's algorithm with and without shadowing

3.1. Description of Baptista's cryptosystem

In [8] Baptista proposed a cryptosystem based on a one-dimensional chaotic map, namely, the logistic map

$$x_{n+1} = b x_n(1 - x_n) \tag{4}$$

where $x_n \in (0, 1)$ and $b \in (3.57, 4]$ are the system variable and parameter, respectively, and n is the number of iterations. Thus, given an initial condition x_0 and a parameter b, the sequence $\{x_n\}_{n=0}^{N}$ is computed.

The encryption scheme proposed in [8] is as follows: let $[x_{\min}, x_{\max})$ be some portion or the whole of the attractor of the logistic map. Divide the interval $[x_{\min}, x_{\max}) \subseteq (0, 1)$ into S ϵ-intervals $X_1, \ldots, X_S$: $X_i = [x_{\min} + (i - 1)\epsilon, x_{\min} + i\epsilon)$, where $\epsilon = \frac{x_{\max} - x_{\min}}{S}$. For each ϵ-interval an alphabet is associated. The values of x_0 and b (preferably) together with the association between the S ϵ-intervals constitute the secret key of the cryptosystem (Table 2).

Denote by $P = \{m_1, m_2, \ldots, m_n\}$ the plaintext to be encrypted which is composed of at most S different characters.

3.1.1. Encryption/decryption procedure

The encryption of the first character, m_1, is performed as follows: we start with an initial condition x_0, and iterate Eq. (1) until its trajectory reaches an ϵ-interval associated with that character. The number of iterations is the ciphertext symbol c_1. To encrypt the second plaintext character, m_2, we iterate Eq. (1) starting from x_{c_1} until x_j, for some $j > c_1$, falls within an ϵ-interval associated with m_2. In general, the ith plaintext character, m_i, for $1 < i \leqslant n$, is encrypted as the number of iterations applied to Eq. (1) to make its trajectory, departing from its current state x_l, for $l = \sum_{k=1}^{i-1} c_k$, reaches an ϵ-interval associated with that character. At the end, the ciphertext is composed by the sequence $\{c_1, c_2, \ldots, c_n\}$. At the receiver's end, the plaintext

Table 1
Number of maps needed to shadow a true trajectory for $b = 3.86$, $N = k_1 + k_2 = 10^7$ and for different initial conditions p_0.

p_0	m	δ_x	δ_b	k_1	k_2
0.00232	2	5.59×10^{-3}	4.85×10^{-4}	2088	9,997,912
0.09	2	5.57×10^{-3}	4.8×10^{-4}	1340	9,998,660
0.1	2	4.66×10^{-3}	3.37×10^{-4}	2642	9,997,358
0.15	2	4.17×10^{-3}	2.7×10^{-4}	1995	9,998,005
0.2	2	5.24×10^{-3}	4.26×10^{-4}	1651	9,998,349
0.3	2	2.16×10^{-3}	7.3×10^{-5}	728	9,999,272
0.35	2	5.52×10^{-3}	4.27×10^{-4}	1833	9,998,167
0.4	2	3.69×10^{-3}	2.12×10^{-4}	2373	9,997,627
0.45	2	4.54×10^{-3}	3.2×10^{-4}	1400	9,998,600
0.6	2	2.97×10^{-3}	1.37×10^{-4}	2332	9,997,668
0.7	2	2.16×10^{-3}	7.3×10^{-5}	718	9,999,282
0.8	2	5.59×10^{-3}	4.8×10^{-4}	3085	9,996,915
0.9	2	5.67×10^{-3}	4.98×10^{-4}	3211	9,996,789

Table 2
Schematic representation of how an attractor $[x_{\min}, x_{\max})$ is divided into $S = 256$ ϵ-intervals with size equals $\frac{x_{\max}-x_{\min}}{256}$, and where each ϵ-interval is associated with a character.

i	ϵ-interval	Possible character's association
1	$[x_{\min}, x_{\min}+\epsilon)$	a
2	$[x_{\min}+\epsilon, x_{\min}+2\epsilon)$	b
3	$[x_{\min}+2\epsilon, x_{\min}+3\epsilon)$	c
.	.	.
.	.	.
.	.	.
26	$[x_{\min}+25\epsilon, x_{\min}+26\epsilon)$	z
.	.	.
.	.	.
.	.	.
.	.	.
254	$[x_{\min}+253\epsilon, x_{\min}+254\epsilon)$	#
255	$[x_{\min}+254\epsilon, x_{\min}+255\epsilon)$	@
256	$[x_{\min}+255\epsilon, x_{\min}+256\epsilon)$	*

character m_1 can be recovered from the ciphertext symbol by iterating the logistic map (1), from its current state, as much times as denoted by the ciphertext symbol c_1. The location of the reached state, with respect to the S ϵ-intervals, determines the original plaintext character.

3.1.2. Constraint on the ciphertext symbols

Baptista emphasized that regardless of the initial state x_0, the ciphertext is a value that does not exceed 65,532. He also emphasized that each ciphertext symbol c_i should be a number that is at least 250. Thus, c_i should satisfy $N_0 \leqslant c_i \leqslant N_{\max}$, where $N_0 = 250$ and $N_{\max} = 65,532$. Since there exists many possibilities for each c_i in $[N_0, N_{\max}]$, an extra coefficient $\eta \in [0,1]$ (chosen by the sender and is known to him/her only) is used to determine the right number c_i to be transmitted: if $\eta = 0$, then c_i is chosen to be the least number of iterations required to make the trajectory departing for some current state reaches the desired ϵ-interval associated with m_i, provided that this number is greater than or equal to 250; otherwise, the sender keeps iterating Eq. (1) until the number of iterations needed to reach the desired ϵ-interval is at least 250. If $\eta \neq 0$, whenever the trajectory reaches the desired ϵ-interval, the sender obtains a number κ from a random number generator (with a normal distribution within $[0,1]$) and compares the two numbers κ and η. If $\kappa \geqslant \eta$, the plaintext character m_i is encrypted as the number of iterations it takes the trajectory to reach that ϵ-interval, provided it is at least 250; otherwise, the sender keeps iterating Eq. (1) until the number of iterations to reach the desired ϵ-interval is at least 250 and the condition $\kappa \geqslant \eta$ is satisfied.

3.2. Implementation of Baptista's algorithm without shadowing

For simplicity, we consider the coefficients $\eta = 0$ and $N_0 = 250$. Suppose that the initial condition is $x_0 = 0.1$, and the system parameter is $b = 3.86$. Suppose the message to be encrypted is the sentence "hello kuwait". This message consists of the following characters: "a", "e", "h", "i", "k", "l", "o", "t", "u", "w" and " " (space).

Let $x_{\max} = 0.8$ and $x_{\min} = 0.2$ (i.e., $[0.2, 0.8)$ is the chosen portion of the attractor of the logistic map). Divide the interval $[0.2, 0.8)$ into $S = 256$ ϵ-intervals $X_1, \ldots, X_{256} : X_i = [0.2 + (i-1)\epsilon, 0.2 + i\epsilon)$, where $\epsilon = \frac{0.8-0.2}{256} = 0.002343570$. For each ϵ-interval an alphabet is associated. Suppose that the 26 English alphabets and the space character are associated with 27 ϵ-intervals as follows (Table 3).

To encrypt the sentence "hello kuwait", one possible ciphertext is: 4078 455 1738 321 962 325 309 388 1065 633 1762 1620. The first ciphertext symbol is obtained by iterating Eq. (1), with initial state $x_0 = 0.1$, until its trajectory falls within the interval $[0.21640625, 0.21875)$, provided the number of iterations is at least 250. The second ciphertext is obtained by iterating Eq. (1), with current state x_{4078}, until its trajectory falls within the interval $[0.209375, 0.21171875)$, provided the num-

Table 3
An ϵ-interval is associated with each of the 27 characters.

a	[0.2, 0.20234375)	j	[0.22109375, 0.2234375)	s	[0.2421875, 0.24453125)
b	[0.20234375, 0.2046875)	k	[0.2234375, 0.22578125)	t	[0.24453125, 0.246875)
c	[0.2046875, 0.20703125)	l	[0.22578125, 0.228125)	u	[0.246875, 0.24921875)
d	[0.20703125, 0.209375)	m	[0.228125, 0.23046875)	v	[0.24921875, 0.2515625)
e	[0.209375, 0.21171875)	n	[0.23046875, 0.2328125)	w	[0.2515625, 0.25390625)
f	[0.21171875, 0.2140625)	o	[0.2328125, 0.23515625)	x	[0.25390625, 0.25625)
g	[0.2140625, 0.21640625)	p	[0.23515625, 0.2375)	y	[0.25625, 0.25859375)
h	[0.21640625, 0.21875)	q	[0.2375, 0.23984375)	z	[0.25859375, 0.2609375)
i	[0.21875, 0.22109375)	r	[0.23984375, 0.2421875)	" "	[0.2609375, 0.26328125)

ber of iterations is at least 250, and so on. These numbers are then transmitted to the receiver over a public channel who can recover the first message character by iterating Eq. (1) 4078 times, with initial state $x_0 = 0.1$, the pointed point $x_{4078} = 0.216503$, whose location is within the 8th ϵ-interval, associated with the letter "h". Then, the receiver iterates Eq. (1), from the current state $x_{4078} = 0.216503$, 455 more times obtaining $x_{4533} = 0.210935$, which is within the 5th ϵ-interval, associated with the letter "e", and so on. The complete message is then recovered after iterating Eq. (1) 13,656 times.

Note that, each ciphertext symbol is dependent on all preceding ciphertext symbols.

3.3. Implementation of Baptista's algorithm with shadowing

The encryption and decryption processes are similar to those of Baptista's algorithm but instead of using the chaotic map (1), we use the chaotic map (3). The encryption of a plaintext character is the number of iterations applied to the chaotic map (3) to make its trajectory reaches an ϵ-interval associated with that character as described in Section 3.2.

The secret key of this scheme is the initial condition x_0, the control parameter b (i.e., b_m in Eq. (3)), and preferably the S associations between the S ϵ-intervals and the S characters of some alphabets. However, in this algorithm, for each initial condition x_0 and control parameter b_m there are one or more maps needed to shadow the numerical orbit. These maps and the values of k_i, for $1 \leqslant i \leqslant m-1$, in Eq. (3) all depend on b_m and the initial condition x_0 (see Table 1). For fixed control parameter b_m, given an initial condition, the values of $b_1, b_2, \ldots, b_{m-1}$ and $k_1, k_2, \ldots, k_{m-1}$ can be computed. These values can be calculated using the shadowing algorithm as described in [49].

Consider the encryption of the message "hello kuwait". As in Section 3.2, let the chosen portion of the attractor be $[0.2, 0.8)$, and for simplicity, consider the coefficients $\eta = 0$ and $N_0 = 250$.

From Table 1, for the initial condition $x_0 = 0.1$, and parameter $b = 3.86$, the number of maps needed to shadow the chaotic trajectory is given by:

$$x_{n+1} = \begin{cases} b_1 x_n(1 - x_n), & \text{if } n \leqslant k_1 \\ b_2 x_n(1 - x_n), & \text{if } k_1 < n \leqslant N, \end{cases} \tag{5}$$

where $b_1 = 3.860337,\ b_2 = 3.86$ and $k_1 = 2642$.

To encrypt the first plaintext character, we iterate Eq. (5), with initial state $x_0 = 0.1$ and $b_1 = 3.860337$, until its trajectory falls within the interval $[0.21640625, 0.21875)$ or until the number of iterations reaches $k_1 = 2642$. If the trajectory falls in that interval before 2642 iterations we record the number of iterations (provided it is at least 250) as the ciphertext symbol and move on to encrypt the second plaintext character. If not, we keep iterating until x_{2642}. At this stage, we replace the parameter b_1 by b_2 and carry on the iterations. Once the trajectory falls in the required interval, we record the number of iterations (from the initial state $x_0 = 0.1$) as the first ciphertext symbol and move on to encrypt the second plaintext character.

To encrypt the sentence "hello kuwait", one possible ciphertext is: 274 1594 509 1308 287 1364 1173 4520 363 438 880 4514. The first ciphertext symbol is obtained by iterating Eq. (5), with initial state $x_0 = 0.1$, until its trajectory falls within the interval $[0.21640625, 0.21875)$. The second ciphertext symbol is obtained by iterating Eq. (5), with current state x_{274}, until its trajectory falls within the interval $[0.209375, 0.21171875)$. The third ciphertext symbol is obtained by iterating Eq. (5), with current state x_{1868}, until its trajectory falls within the interval $[0.22578125, 0.228125)$. The fourth ciphertext symbol is obtained by iterating Eq. (5), with current state x_{2377}, 265 times (until x_{2642}). At this stage, we replace b_1 by b_2 and carry on iterating until the trajectory falls within the interval $[0.22578125, 0.228125)$. The fifth ciphertext symbol is obtained by iterating Eq. (5), with current state x_{3685}, until its trajectory falls within the interval $[0.2328125, 0.23515625)$, and so on. The complete message is encrypted after iterating Eq. (5) 17,224 times.

The decryption of the ciphertext is done by iterating the chaotic map (5) as much times as indicated by the ciphertext. The position of the final point, with respect to the S ϵ-intervals determines the plaintext character to the receiver.

4. Comparison of the two algorithms

In this section, we perform a comparison of both algorithms based on the number of times each ϵ-interval is reached departing from any initial condition in Eqs. (1) and (3). We also compare the security level of the proposed algorithm with Baptista's one.

We start by investigating the least number of times an ϵ-interval is visited. Similar ergodicity principles to those invoked in Baptista's work, which assure that, departing from almost any initial condition in Eq. (1), each of the ϵ-intervals is reached several number of times, provided that this interval is a portion of the attractor, also apply to our implementation with shadowing.

Tables 4 and 5 below show a comparison between both algorithms when considering a portion of the attractor $[0.2, 0.8)$, for $N = 65,532$ and $N = 10^7$, respectively. This comparison is based on the least number of times an ϵ-interval is reached.

For the reason that different initial conditions results in the same natural invariant density [8], a 65,532-iteration sized trajectory departing from any initial condition visits any ϵ-interval at least approximately 25 times in the case of unshadowed logistic map and 26 times in the case of shadowed logistic map.

Table 4
Comparison between the two algorithms based on the minimum number of times an ϵ-interval is reached for $N = 65,532$.

Initial condition	Least number of times an ϵ-interval is reached	
	Unshadowed map	Shadowed map
x_0		
0.00232	30	19
0.09	22	25
0.1	24	28
0.15	20	23
0.2	27	33
0.25	24	22
0.3	26	25
0.35	26	26
0.4	27	30
0.45	26	30
0.5	24	30
0.6	27	29
0.7	27	19
0.8	25	30
0.9	19	22

Table 5
Comparison between the two algorithms based on the minimum number of times an ϵ-interval is reached for $N = 10^7$.

Initial condition	Least number of times an ϵ-interval is reached	
	Unshadowed map	Shadowed map
x_0		
0.00232	5464	5445
0.09	5410	5385
0.1	5400	5443
0.15	5502	5352
0.2	5404	5361
0.25	5398	5461
0.3	5433	5411
0.35	5417	5371
0.4	5437	5433
0.45	5355	5437
0.6	5437	5354
0.7	5391	5485
0.8	5428	5478
0.9	5399	5345

Thus, a 10^7-iteration sized trajectory departing from any initial condition goes towards any ϵ-interval at least approximately 5420 times in the case of unshadowed logistic map and 5411 times in the case of shadowed logistic map.

Therefore, we conclude that each of the 256 ϵ-interval is visited approximately the same number of times in the case of unshadowed and shadowed logistic maps.

Remark 1. Experimental simulations have shown that regardless of the value of the system parameter b, the number of times an ϵ-interval is reached in the case of unshadowed logistic map is more or less the same in the case of shadowed logistic map.

In what follows, we consider the security issue of the two algorithms. In both algorithms, the secret key consists of the initial condition x_0, the control parameter b (b_m in Eq. (3)), and preferably the S associations between the S ϵ-intervals and the S characters of some alphabets. However, for a practical system the key should only consist of x_0 and b. Suppose that the key is $K = (x_0, b)$. Baptista's algorithm without shadowing involves only these two parameters in the encryption and decryption processes. However using the method of shadowing, for each x_0 and b_m in Eq. (3) there are one or more maps with control parameters $b_1, b_2, \ldots, b_{m-1}$ needed to shadow the numerical orbit. Thus, the shadowing method requires the generation of further $2(m-1)$ parameters, namely, $b_1, b_2, \ldots, b_{m-1}$ and $k_1, k_2, \ldots, k_{m-1}$, where these parameters are not part of the secret key but are related to it. One way for the sender and the receiver to separately compute these values is by using the shadowing algorithm proposed in [49]. Hence, Baptista's algorithm with shadowing involves a further $2(m-1)$ parameters making the total number of parameters $2m$. Therefore, a cryptanalyst trying to break the modified algorithm is faced with further challenges (directly related to the individual initial condition and control parameter). Thus, the problem of having to guess correctly the initial condition and control parameter becomes a must in this case as without the complete knowledge of these two parameters a cryptanalyst cannot compute the maps $b_1, b_2, \ldots, b_{m-1}$ which are updated throughout the encryption and decryption processes at the stages $k_1, k_2, \ldots, k_{m-1}$ as denoted in Eq. (3). Hence, this modification results in improving the security level of the cryptosystem against cryptanalytic attacks such as the those proposed on Baptista's algorithm in [35,38].

5. Concluding remarks

We have proposed a modified approach of Baptista's algorithm based on the shadowing method. The main advantage of our algorithm is to enhance the security level of the system leaving a cryptanalyst with no choice but to guess correctly the initial condition and the control parameter of the map through an exhaustive search. Moreover, when encrypting a reasonable sized message, the number of iteration applied to the logistic map may involve thousands if not millions of iterates, therefore, one should be very careful to ensure that the orbits generated by the logistic map are chaotic and not just noise driven by computer round-off error.

Acknowledgment

This work was supported by Kuwait University Research Grant No. [SM04/07].

References

[1] Menezes AJ, van Oorschot PC, Vanstone SA. Handbook of applied cryptography. CRC Press; 1997.
[2] Matthews R. On the derivation of a chaotic encryption algorithm. Cryptologia 1989;XIII(1):29–42.
[3] Pecora LM, Carroll TL. Synchronization in chaotic systems. Phys Rev Lett 1990;64:821–4.
[4] Habutsu T, Nishio Y, Sasase I, Mori S. A secret key cryptosystem by iterating a chaotic map. In: Advances in cryptology: EUROCRYPT 91. Lecture notes in computer science, vol. 0547. Berlin: Springer-Verlag; 1991. p. 127–40.
[5] Hayes S, Grebogi C, Ott E. Communicating with chaos. Phys Rev Lett 1993;70(20):3031–4.
[6] Alvarez G, Monotoya G, Romera M. Chaotic cryptosystems. In: Proceedings of the IEEE 33rd annual international carnahan conference on security technology, IEEE; 1999. p. 332–8.
[7] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos 1996;6(2):219–49.
[8] Baptista MS. Cryptography with chaos. Phys Lett A 1998;240:50–4.
[9] Alvarez E, Fernandez A, Garcia P, Jimenez J, Marcano A. New approach to chaotic encryption. Phys Lett A 1999;263:373–5.
[10] Kotulski Z, Szczepanski J. Application of discrete chaotic dynamical systems in cryptography – DCC method. Int J Bifurcat Chaos 1999;9(6):1121–35.
[11] Dachselt F, Schwarz W. Chaos and cryptography. IEEE Trans Circ Syst I 2001;48(12):1498–509.
[12] Kocarev L. Chaos-based cryptography: a brief overview. IEEE Circ Syst Mag 2001;1(3):6–21.
[13] Schmitz R. Use of chaotic dynamical systems in cryptography. J Franklin Inst 2001;338(4):429–41.
[14] Shujun L, Xuanqin M, Yuanlong C. Pseudo-random bit generator based on couple chaotic systems and its application in stream-ciphers cryptography. In: Advances in cryptology: INDOCRYPT 01. Lecture notes in computer science, vol. 2247. Berlin: Springer-Verlag; 2001. p. 316-29.
[15] Papadimitriou S, Bountis T, Mavaroudi S, Bezerianos A. A probabilistic symmetric encryption scheme for very fast secure communications based on chaotic systems of difference equations. Int J Bifurcat Chaos 2001;11(12):3107–15.
[16] Wong W, Lee L, Wong K. A modified chaotic cryptographic method. Comput Phys Commun 2001;138:234–6.
[17] Li S, Mou X, Cai Y. Improving security of a chaotic encryption approach. Phys Lett A 2001;290(3–4):127–33.
[18] Jakimoski G, Kocarev L. Chaos and cryptography: block encryption ciphers based on chaotic maps. IEEE Trans Circ Syst I 2001;48(2):163–9.
[19] Li S, Zheng X, Mou X, Cai Y. Chaotic encryption scheme for real-time digital video. In: Real-time imaging VI. Proceedings of SPIE, vol. 4666; 2002. p. 149–60.
[20] Masuda N, Aihara K. Cryptosystems with discretized chaotic maps. IEEE Trans Circ Syst I 2002;49(1):28–40.
[21] Machado R, Baptista M, Grebogi C. Cryptography with chaos at the physical level. Chaos, Solitons & Fractals 2004;21(5):1265–9.
[22] Wei J, Liao X, Wong K, Xiang T. A new chaotic cryptosystem. Chaos, Solitons & Fractals 2006;30(5):1143–52.
[23] Mi B, Liao X, Chen Y. A novel chaotic encryption scheme based on arithmetic coding. Chaos, Solitons & Fractals 2008;38(5):1523–31.
[24] Behnia S, Akhshani A, Mahmodi H, Akhavan A. A novel algorithm for image encryption based on mixture of chaotic maps. Chaos, Solitons & Fractals 2008;35(2):408–19.
[25] Yang H, Liao X, Wong K, Zhang W, Wei P. A new block cipher based on chaotic map and group theory. Chaos, Solitons & Fractals 2008;35(2):408–19.
[26] Kanso A, Smaoui N. Logistic chaotic maps for binary numbers generations. Chaos, Solitons & Fractals 2009;40(5):2557–68.
[27] Ogorzatek MJ, Dedieu H. Some tools for attacking secure communication systems employing chaotic carriers. In: Proceedings of the IEEE international symposium circuits and systems, vol. 4, IEEE; 1998. p. 522–5.
[28] Short KM. Signal extraction from chaotic communications. Int J Bifurcat Chaos 1997;7(7):1579–97.
[29] Wong KW, Lee L, Wong K. A modified chaotic cryptographic method. Comput Phys Commun 2001(3):234–6.
[30] Wong KW. A fast chaotic cryptographic scheme with dynamic look-up table. Phys Lett A 2002;298(4):238–42.
[31] Wong KW. A combined chaotic cryptographic and hashing scheme. Phys Lett A 2003;307(5–6):292.
[32] Wong KW, Ho SW, Yung CK. A chaotic cryptography scheme for generating short ciphertext. Phys Lett A 2003;310:67–73.
[33] Palcois A, Juarez H. Cryptography with cycling chaos. Phys Lett A 2002;303(5–6):345–51.
[34] Li S, Mou X, Ji Z, Zhang J, Cai Y. Performance analysis of Jakimoski–Kocarev attack on a class of chaotic cryptosystems. Phys Lett A 2003;307(1):22.
[35] Alvarez G, Montoya F, Romera M, Pastor G. Cryptanalysis of an ergodic chaotic cipher. Phys Lett A 2003;311(23):172–9.
[36] Alvarez G, Montoya F, Romera M, Pastor G. Keystream cryptanalysis of a chaotic cryptographic method. Comput Phys Commun 2004;156(2):205–7.
[37] Alvarez G, Montoya F, Romera M, Pastor G. Cryptanalysis of dynamic look-up table based chaotic cryptosystems. Phys Lett A 2004;326(34):211–8.
[38] Jakimoski G, Kocarev L. Analysis of some recently proposed chaos-based encryption algorithms. Phys Lett A 2001;291(6):381–4.
[39] Li Shujun, Mou Xuanqin, Ji Zhen, Zhang Jihong, Cai Yuanlong. Performance analysis of Jakimoski–Kocarev attack on a class of chaotic cryptosystems. Phys Lett A 2003;307(1):22–8.
[40] Hammel S, Yorke J, Grebogy C. Do numerical orbits of chaotic dynamical processes represent true orbits? J Complexity 1987;3:136–45.
[41] Hammel S, Yorke J, Grebogy C. Numerical orbits of chaotic processes represent true orbits. Bull Am Math Soc (New Series) 1988;19(2).
[42] Sauer T, Yorke J. Rigorous verification of trajectories for the computer simulation of dynamical system. Nonlinearity 1991;4(3):961–79.
[43] Kostelich E, Grebogi C, Ott E, Yorke J. Higher-dimensional targeting. Phys Rev E 1993;47(1).
[44] Shinbrot T, Ott E, Grebogi C, Yorke J. Using chaos to direct orbits to targets in systems describable by a one-dimensional map. Phys Rev A 1992;45(6).
[45] Anosov D. Geodesic flows and closed Riemannian manifolds with negative curvature. Proc Steklov Inst Math 1967;90.
[46] Bowen R. *w*-limit sets for axiom A diffeomorphisms. J Diff Equat 1975;18.
[47] Coven E, Kan I, Yorke J. Pseudo-orbit shadowing in the family of tent maps. Trans Am Math Soc 1988;308(1).
[48] Nusse H, Yorke J. Is every approximate trajectory of some process near an exact trajectory of a nearby process? Commun Math Phys 1988;114:363–79.
[49] Smaoui N, Kostelich E. Using chaos to shadow the quadratic map for all time. Int J Comput Math 1998;70:117–29.