Cryptanalytic methods in chaotic cryptosystems

G. Álvarez, F. Montoya, M. Romera, and G. Pastor

Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas

Serrano, 144 — 28006 Madrid, Spain

ABSTRACT

In recent years, telecommunications networks have undergone an explosive growth. As a consequence, there has been a strong demand of information protection mechanisms. Many cryptosystems based on chaos have been proposed, although little or no critical analysis has been made about the security and cryptographic robustness of these algorithms. In this paper we present our tools to examine some of these algorithms from a cryptographic perspective, showing many vulnerabilities that can be exploited to successfully break them. We conclude that most of the chaotic cryptosystems are very insecure and cumbersome, thus, unreliable and impractical for real applications.

Keywords: Chaotic Cryptosystems, Cryptanalysis, Gray Codes.

1. INTRODUCTION

Modern telecommunications networks, and specially Internet, have increased the possibilities of user communications and information transmission to limits unimaginable a short time ago. There is a parallel growing cryptographic techniques demand, which has originated an intense research activity and the search of new directions in cryptography.

As a result, a rich variety of chaotic cryptosystems for end to end communications have been put forward, whose robustness and privacy are equally diverse [1-9].

Up to date, little or no critical analysis has been made about the security and cryptographic robustness of these algorithms [10-16]. We have detected that a systematic approach to cryptanalysis and security evaluation is missing. To fill this void, in this paper we examine some of these algorithms from a cryptographic perspective.

First, in section 2, we propose some new analysis tools based on the theory of 1D quadratic maps, such as Gray codes [17], an extension of the Myrberg method [18] or the well known bifurcation diagrams and histograms.

Second, in section 3, we make use of these tools to successfully attack the proposed cryptosystems. Depending on the cipher under study and its parameter configuration, some or all of the following attacks prove to be successful, usually with a surprisingly low number of texts: ciphertext-only, known-plaintext, chosen plaintext, and chosen ciphertext.

After our cryptanalysis, we conclude that most of the chaotic cryptosystems are very insecure and, thus, unreliable for critical applications.

2. CRYPTANALYSIS TOOLS

Chaotic cryptosystems, as any other cryptosystem, seek to offer three important properties to frustrate cryptanalytic efforts, namely [6]:

i) *Be sensitive with respect to keys*: flipping one bit in a key creates completely different ciphertext when applied to the same plaintext.

ii) *Be sensitive with respect to plaintext*: flipping one bit in the plaintext creates completely different ciphertext.

iii) *Map plaintext to random ciphertext*: there should not be any patterns in the ciphertext, if the cryptosystem is good.

These three properties can be easily related to three characteristics of chaotic systems, respectively:

i) *Parameter sensitivity*: small variation in one of the system parameters is enough to make two trajectories, starting at the same initial point, separate at exponential rate.

ii) *Initial condition sensitivity*: two trajectories starting at two different, though arbitrarily close, initial points separate from each other exponentially.

iii) *Ergodicity*: the trajectories followed by points belonging to the phase space travel through the space with uniform distribution.

Although chaotic systems satisfy all these properties, they are deterministic in nature after all. As a consequence, it is possible to detect patterns in their behaviour, which can be readily used by the cryptanalyst to find order within the apparent chaos.

To serve this purpose, we make use of the following three tools, adapted from the well known chaos theory background: Gray codes [17], hyperbolic components centres determination using our extension of the Myrberg method [18], and bifurcation diagrams and histogramas. The field of application of these tools is restricted to unimodal maps, with one critical point.

Gray codes

A Gray code is a function $G(i)$ of the integers i, that for each integer $N \geq 0$ is one-to-one for $0 \leq i \leq 2^{N-1}$, and that has the following remarkable property: The binary representation of $G(i)$ and $G(i+1)$ differ in exactly one bit.

Let $x \alpha \ f_c(x)$ be a family of 1-D quadratic maps, of parameter c, which transforms an interval I into itself. To represent symbolically the dynamics of the orbit followed by an initial point x_0 for a given parameter value c, we do not record the exact value of each iterate, but consider simply if it falls to the left (L), to the right (R), or on the critical point (C) of the map. Thus, from the orbit x_0, $x_1 = f_k(x_0)$, $x_2 = f_k(x_1)$,…, $x_n = f_k(x_n-1)$,… one gets a symbolic sequence $S = s_0 s_1 s_2 \ldots s_n \ldots$ in one-to-one correspondence, where

$$s_i = \begin{cases} \mathrm{L} & \text{if} \quad x_i < \mathrm{C} \\ \mathrm{C} & \text{if} \quad x_i = \mathrm{C} \\ \mathrm{R} & \text{if} \quad x_i > \mathrm{C} \end{cases}.$$

In Fig. 1 we plotted the graph of the real Mandelbrot map, $x_{n+1} = x_n^2 + c$, with $c = -2$. In the low part appears the order

corresponding to the open intervals on the *x*-axis when considering until three initial symbols. If *n* initial symbols are considered, the points separating the intervals have the property that orbits whose initial point corresponds to an interval separator are preperiodic. These interval separators, which will be denoted as $I_{n+1,1}^{(i)}$, with $1 \le i \le 2^{n-1}$, the first subindex being the preperiod and the second subindex being the period, are calculated as the zeroes of $f_c^{n-1}(x) = 0$. For example, for three initial symbols (n = 3) and $c = -2$, the family of interval separators $I_{4,1}^{(i)}$ are the four zeroes of $f_{-2}^{2}(x) = (x^2 - 2)^2 - 2 = 0$, namely, $\pm\sqrt{2+\sqrt{2}}$ and $\pm\sqrt{2-\sqrt{2}}$. Of course, the previous level interval separators $I_{2,1}$, $I_{3,1}^{(1)}$ and $I_{3,1}^{(2)}$ are still acting as separators in between next level separators. Let us note that when substituting L's for 1's and R's for 0's, the sequences are ordered in Gray code.

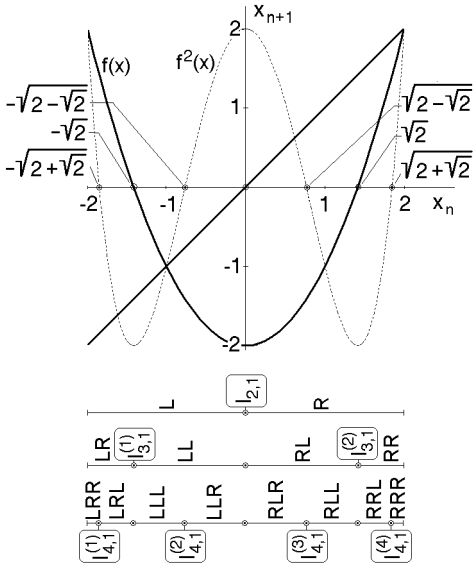

***Fig. 1**. Symbolic sequences for the real Mandelbrot map with c = –2.*

Hyperbolic components centres

The *c* value of the real Mandelbrot map's parameter originating a real superstable orbit of a given symbolic sequence CXX…X can be obtained by repeatedly iterating Myrberg's modified formula:

$$c_{n+1} = \pm\sqrt{-c_n \pm \sqrt{-c_n \pm \ldots \pm \sqrt{-c_n}}} \qquad (1)$$

where the symbol ± stands for a + sign or a – sign according to the expression of the orbit's symbolic dynamics, assigning the + sign to the letter R and the – sign to the letter L.

Bifurcation diagrams and histograms

A bifurcation diagram represents the position of the fixed points of a map as a function of the parameter value, usually showing a number of bifurcations. They constitute an invaluable tool for the study of 1D maps.

Another way to obtain information about a map for a given parameter value is through its histogram. After dividing up the attractor in subintervals, the histogram represents in normalized form the relative frequency of visits to each interval when an initial point is iterated under a parameter value.

4. ATTACK METHODS

In this section we describe how these tools help us to find order underlying the apparent randomness in the proposed chaotic systems, which is enough to open vulnerabilities for attack.

Gray codes

Whenever a cryptographic algorithm uses a unimodal map and a succession of iterates to code a binary sequence, according to whether they fall to the left (say "1") or to the right (say "0") of the critical point, Gray codes turn out to be of great assistance to the cryptanalyst.

As a first example, let us consider the cryptosystem proposed by Alvarez *et al*. [1], based on the tent map:

$$f(x) = \begin{cases} rx & \text{if } x \le 0.5 \\ r(1-x) & \text{if } x \ge 0.5 \end{cases} \qquad (2)$$

The encryption process can be described in the following way: choose a suitable real number *r*, the parameter of the dynamical system (2), as the key of the system. Next, consider the first block of information bits to be transmitted, of length b_1, and start iterating Eq. (2) from an arbitrary initial condition x_0. Construct a chain C_1 of 0's and 1's according to the convention: $x_n \le 1/2 \to 0$ and $x_n > 1/2 \to 1$. As this chain is being generated, keep looking for the repetition in it of the bits of the first block b_1. When this pattern appears, record the value of x_{n_1} at which this pattern began and stop iterating. The vector (b_1, x_{n_1}) constitutes the ciphertext of the first block of b_1 bits of the plaintext. The encryption process continues by selecting the next new b_2-bit length block and iterating from a new arbitrary initial value until the same pattern is generated by the orbit of the dynamical system (2). The next ciphertext unit would be made up by the block length and the value of the iterate at which the pattern appeared: (b_2, x_{n_2}). This process goes ahead until the plaintext is exhausted. The ciphertext units are decrypted by iterating b_i times the initial condition x_{n_i}, and using the threshold 1/2 to convert the sequence of real numbers thus obtained into the correct sequence of bits.

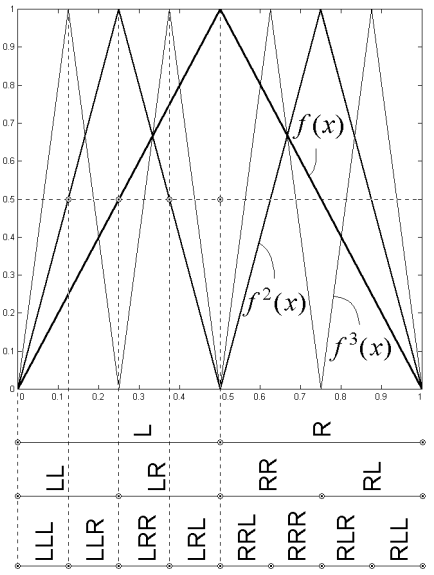

***Fig. 2**. Graphic of tent map for r = 2.0: upper part, f(x), $f^2(x)$, and $f^3(x)$; lower part, first letters of the symbolic sequences followed by initial points within the indicated intervals.*

In Fig. 2, we show a representation of the first, second and third order iteration of *f*(*x*). Below, we represent the first letters of the symbolic sequence of the orbit described by any initial point within the interval thus delimited. It is easy to observe that the leftmost interval for the *i*-th iteration is delimited by the origin and the first peak of $f^i(x)$, whose coordinate can be calculated as $x_p = 1/(2r^{i-1})$. Hence, for $i = 1$, the interval for which all

initial points give rise to symbolic sequences of the form L… is (0, 1/2); for $i = 2$, symbolic sequences LL… are originated by initial points in (0, 1/(2r)); for $i = 3$, the interval corresponding to symbolic sequences LLL… is (0,1/(2r^2)); and so on. As a matter of fact, these sequences are ordered according to a Gray code.

The following chosen ciphertext attack finds r, looking for the value of the first peak of the b-th iteration of $f(x)$:

1. Choose a ciphertext (b, x_0), with x_0 sufficiently close to the origin.
2. Request the decrypted plaintext.
3. Check the plaintext sequence: if it is made of all 0's, then choose a new initial point slightly bigger; if it is all 0's but the last bit, then choose a new initial point slightly smaller.
4. Repeat until the value of x_p has been obtained with the desired precision and then calculate the parameter value as $r = \sqrt[b]{1/(2x_p)}$.

For instance, as in [1], let the secret key be r = 1.99 and $b_{\max}$ = 16. To begin with, we try the following ciphertext: (16, 10^{-5}), obtaining the sequence 00…0. We try next (16, 2×10^{-5}), obtaining 00…01. Therefore, we know that the correct value must lie in between 10^{-5} and 2×10^{-5}. We perform a binary search, trying (16, 1.5×10^{-5}), from which we obtain 00…0. Next we try (16, 1.75×10^{-5}), whose plaintext is 00…01. Continuing with this process we reach the exact value of the secret key r = 1.99 after having used 18 units of chosen ciphertext. As a result from our several tests, we have checked that our method of attack successfully retrieves the exact key in less than 20 steps.

Gray codes can be further used to exploit another vulnerability in this cryptosystem. When we consider the dynamical system (2) when r = 2, there is a uniformity in the lengths of subintervals corresponding to a given symbolic sequence (see Fig. 2). The set of points having sequence $s_1s_2…s_k$ has length 2^{-k}, independent of the sequence. As r departs from 2, the length of the subintervals starts varying slightly, but still remains close enough to the uniform distribution as to give a good hint on the orbit followed by initial points within those subintervals. Under these circumstances, the following ciphertext-only attack, the most difficult of all, succeeds in finding the plaintext by making simple guesses about the sequence of symbols originated by those initial points:

1. Given the first ciphertext unit, (b, x_0), divide up the unit interval in 2^b subintervals of equal length 2^{-b}.
2. Find in which of these subintervals the initial point x_0 is located.
3. The plaintext will be the symbolic sequence associated to that interval (see Fig. 2), changing L's into 0's and R's into 1's.
4. Proceed with the next ciphertext unit in the same way.

For instance, considering an 8-bit block size, the initial value $x_{n_1} = 0.492690$ of the first ciphertext unit, lies in the 126-th subinterval. Any initial point in this subinterval gives rise to a symbolic sequence LRLLLLLR… Simply translating into binary code, we get the guess 01000001. Following with this process, we construct Table 1, where we have listed the results of such guesses for a sequence of 15 ciphertext units. It can be seen that almost all the bits are guessed correctly, without any knowledge of the secret key! In the example, our method of attack is able to recover correctly 117 bits out of 120. The closer the value of the secret key r is to 2.0, the better this method works.

Plaintext (binary)	Ciphertext	Guess (binary)
C(0100001**1**)	0.492690	A(0100000**1**)
r(0111001**0**)	0.363853	s(0111001**1**)
i(01101001)	0.305905	i(01101001)
p(01110000)	0.374380	p(01110000)
t(01110100)	0.345842	t(01110100)
o(01101111)	0.292097	o(01101111)
l(0110110**0**)	0.285359	m(0110110**1**)
o(01101111)	0.290362	o(01101111)
g(01100111)	0.272265	g(01100111)
y(01111001)	0.318392	y(01111001)
i(01101001)	0.305906	i(01101001)
s(01110011)	0.365439	s(01110011)
t(01110100)	0.345434	t(01110100)
h(01101000)	0.311260	h(01101000)
e(01100101)	0.276883	e(01100101)

***Table 1**. Message recovered in a ciphertext-only attack. Differences appear in bold face.*

As a second example, let us consider the algorithm proposed by Habutsu *et al*. [7]. To encrypt a plaintext P into C, it uses the inverse of another version of the tent map:

$$F^{-1}:\begin{cases} x_{n-1} = \alpha x_n & r_i = 0 \\ x_{n-1} = (\alpha - 1)x_n + 1 & r_i = 1 \end{cases}$$

iterating n = 75 times. To resolve the ambiguity of which expression to use when going backwards, a random value r_i is used. This means that for each plaintext there can exist 2^{75} different ciphertexts. Thus, $C = f^{-75}(P)$. However, when decrypting the ciphertext C into P, it uses the direct form of the map, i.e.:

$$F:\begin{cases} x_{n+1} = \dfrac{x_n}{\alpha} & (0 \le x_n \le \alpha) \\ x_{n+1} = \dfrac{x_n - 1}{\alpha - 1} & (\alpha < x_n \le 1) \end{cases}$$

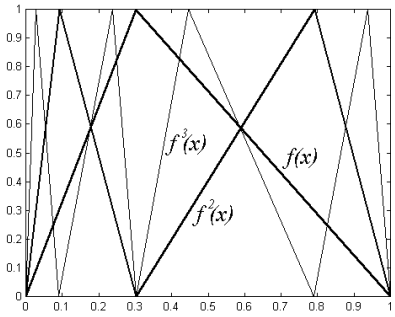

***Fig. 3**. Graphic of Habutsu's tent map [7] for α = 0.3: f(x), $f^2(x)$, and $f^3(x)$.*

Hence, $P = f^{75}(C)$. It is precisely during the decryption process that this algorithm is totally vulnerable to a chosen ciphertext attack. In Fig. 3, we show a representation of the first, second and third order iteration of $f(x)$. The leftmost interval for the i-th iteration is delimited by the origin and the first peak of $f^i(x)$, whose coordinate can be calculated as $x_p = \alpha^i$. Thus, if we choose a ciphertext C sufficiently small, then we can obtain the key value as $\alpha = \sqrt[75]{C/P}$. This attack recovers the exact key with just one ciphertext!

As a last example, let us consider Baptista's cryptosystem [4], based on the property of ergodicity of chaotic systems, i.e., the eventual visit of the trajectory to all partitions in the phase space as the number of iterations grows. His cryptosystem exploits this property by using the logistic map

$$x_{n+1} = bx_n(1-x_n) \tag{3}$$

where $x_n \in (0,1)$ and the parameter b is chosen so that Eq. (3) behaves chaotically.

The message to be transmitted is considered to be coded in a s-symbol alphabet. The interval (0,1) is thus divided up into s sub-intervals of length ε, in a one-to-one association with the s symbols. In Fig. 4 we have represented a schematic view of how the association between the s ε-intervals and the s symbols takes place. Each interval, or site, is in the range $[x_{\min}+(s-1)\varepsilon, x_{\min}+s\varepsilon]$, where s can take any value, v.g. in [BAP 98] s = 256. It is clear that $\varepsilon = (x_{\max}-x_{\min})/s$ and $[x_{\min}, x_{\max}]$ is a portion of the attractor or the whole one.

The ciphertext generated as the number of iterations needed by the orbit to land on the interval which corresponds to the given plaintext symbol starting from an initial condition x_0. To recover the original text, Eq. (3) is iterated from x_0 as many times as indicated by the ciphertext. After this number of iterations is reached, the orbit will have landed on an interval which corresponds to the plaintext symbol.

The system key is formed by the possible associations between the s intervals and symbols, the value of x_0 and the value of the parameter b. For every new symbol to be encrypted, a new x_0 is chosen equal to the value of the last iterate, corresponding to the previous symbol. For example, if the symbol s_1 is encrypted as c_1, corresponding to the number of iterations needed to land on the interval s_1 starting from x_0, then the next initial value will be $x_0' = f^{c_1}(x_0)$, where f^{c_1} represents the c_1-th iteration of Eq. (3), and so on.

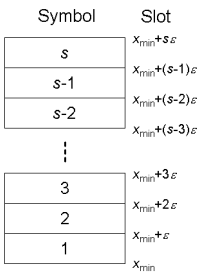

***Fig. 4**. Schematic representation of the attractor partitioned in slots of size ε and their association with the language of s symbols.*

Let us consider a known plaintext attack, in which we know the first pairs of plain and ciphertext of a message and that we know the parameter value, which stands for half of the key. For the sake of simplicity, we will consider a source S_2, emitting only two symbols; the attractor defined by the real Mandelbrot map, instead of the logistic map; and the interval $(x_{\min}=-2, x_{\max}=2)$, without loss of generality. We set the initial point to x_0 = 0.232323, to the right of the critical point, and c = –1.8. Under these conditions, the number of iterations needed to encrypt each symbol equals the number of times that the orbit remains to the left or to the right of the critical point. Let us agree that s_1 corresponds to the left region, while s_2 corresponds to the right region. Therefore, if the plaintext is $s_1s_2s_2s_1s_1s_1s_1$…, then the ciphertext (1 1 3 1 1 2 1 …) allows us to reconstruct the symbolic sequence of the orbit followed: O = RL R LLR L L RL L … Next, we need to narrow the interval where the initial value x_0 lies in, or equivalently, we have to work out the value of the interval separators delimiting the interval which corresponds to that symbolic sequence O. If O is translated into Gray code, by simply substituting R's by 0's and L's by 1's, then we obtain g_0 = 01011011011. Next, we add and subtract one, still operating in Gray code, obtaining: g_1 = 01011011010 and g_{-1} = 01011011001. Last, we take the common part of g_0 and g_1 and of g_0 and g_{-1}. Substituting 1's by '–' and 0's by '+' in Eq. (1), we obtain finally:

$$\left.\begin{matrix} g_1 = \underline{01011011010} \\ g_0 = \underline{01011011011} \end{matrix}\right\} 0101101101 \rightarrow +\sqrt{-c-\sqrt{-c+\sqrt{-c-\sqrt{-c-\sqrt{-c+\sqrt{-c-\sqrt{-c-\sqrt{-c+\sqrt{-c-\sqrt{-c}}}}}}}}}}$$

$$\left.\begin{matrix} g_{-1} = \underline{01011011001} \\ g_0 = \underline{01011011011} \end{matrix}\right\} 0101101101 \rightarrow +\sqrt{-c-\sqrt{-c+\sqrt{-c-\sqrt{-c-\sqrt{-c+\sqrt{-c-\sqrt{-c-\sqrt{-c+\sqrt{-c}}}}}}}}}$$

which evaluated at c = –1.8 leads to a lower and an upper bound of x_0, (0.2254207616, 0.2434413280). The more pairs of clear and ciphertext we have, the greater the precision with which x_0 can be bounded. Continuing with this example, if more than 7 plain and ciphertext units were known, say 20, the bounding interval would be narrowed to (0.23232170,0.23232332). This method allows us to drastically decrease the key space for an exhaustive key search, from 10^{16} a 4.05×10^{9}. Obviously, given more units of plain and ciphertext, the key space can be further reduced.

Let us consider now a higher order source, S_4, emitting four different symbols. In this case, both symbols s_1 and s_2 correspond to letter L, whereas s_3 and s_4 correspond to letter R. Let us consider a chosen plaintext attack, in which we can request the ciphertext of a message consisting of all its units set to s_1. The number of iterations used to encrypt each symbol indicates the number of sites different from s_1 that are visited before landing on s_1. However, we can not assume that this simply means an R, because the site corresponding to s_2 lies also to the left of the critical point. Therefore, we construct a sequence of 1's and 0's from the ciphertext, where the 1 represents the symbol s_1 and the 0, any other symbol. Next, we repeat the process with a chosen plaintext with all units set to s_2. Another sequence of 1's and 0's is thus generated and we XOR both sequences. Let us see how this work with an example. Again, x_0 = 0.232323 y c = –1.8. The result of encrypting the plaintext ($s_1s_1s_1s_1s_1s_1s_1$) is (1 3 3 3 2 2 2). Its associated Gray sequence is (1001001001010101). When the plaintext ($s_2s_2s_2s_2s_2s_2s_2$) is encrypted, we obtain (3 3 3 4 7 6 4), whose associated Gray sequence is (0010010010001000000010000010001). XORing both sequences we obtain 101101101101110101… Finally, we add a 0 at the beginning of the sequence, assuming with 50% probability that x_0 lies to the right. Taking this sequence as g_0, we construct g_1 and g_{-1} and from them we can work out a lower and an upper bound of x_0 as before. In this case, we obtain (0.2322592037, 0.2323430867). Once again, the interval where x_0 lies will be increasingly narrowed as more clear and ciphertext units are taken into account.

This process can be repeated for any other higher order source S_i. In each case, a chosen plaintext of S_i /2 symbols has to be used, obtaining similar results.

Hyperbolic components centres

Myrberg equation (1) can still be used for another attack on Baptista's algorithm. Let us suppose now that we know the value of the initial point x_0. Under this circumstance, it is possible to estimate the value of the parameter and recover the complete key from a few pairs of known plain and ciphertexts.

To begin with, we shall use the S_2 source, the initial point x_0 = 0.232323 and c = –1.8. Given a known plain and ciphertext pair, say $s_1s_2s_2s_1s_1s_1s_1$…, and (1 1 3 1 1 2 1 …), respectively, we construct the symbolic sequence followed by the orbit: O = RL R LLR L L RL L … Next, taking the letters in

this sequence one by one, we construct a series of discrete dynamical systems which will get closer and closer to the parameter value which originated the given sequence. For O = RL…, we have that

$$+\sqrt{-c-\sqrt{-c}} = x_0 \,;\, -c = \sqrt{-c} + x_0^2$$

which allows us to construct the dynamical system $-c_{n+1} = \sqrt{-c_n} + x_0^2$. Iterating from $c_0 = 0$, we obtain the fixed point $c^* = -1.105$, our first estimate of c. For O = RLR…, we have the dynamical system $-c_{n+1} = \sqrt{-c_n + \sqrt{-c_n}} + x_0^2$, which converges to the value $c^* = -1.843$. If we continue adding more letters to the orbit, we obtain the following succession of estimates for c: 1.447, –1.7123, –1.8106, –1.7568, –1.7853, –1.8039, –1.7928, –1.7981, –1.8022, –1.7996, –1.8008, –1.8001, –1.7998, … Hence, with a few pairs of plain and ciphertext units we have estimated the parameter value with an error of 10^{-4}. Of course, given more text units, the estimation will be better. The application of this method to higher order sources is immediate.

Bifurcation diagrams and histograms

Although well known in the study of 1D maps, bifurcation diagrams and histograms are valuable tools for the cryptanalyst, and should be routinely used by the cryptosystem designer too, as they can pinpoint weaknesses in an algorithm.

First, we shall use the bifurcation diagram of the tent map, depicted in Fig. 5, to identify a vulnerability in Alvarez's algorithm [1]. When moving to the left from $r = 2$, the interval visited by the orbits of Eq. (2) shrinks steadily, getting smaller as r decreases. The shape of the curves in Fig. 5 can be computed from what we call the critical polynomials of Eq. (2), defined as $P_{n+1} = f(r, P_n)$. Starting from $P_0 = x_0 = 1/2$, the tent map's critical point, we obtain:

$$P_0 = 1/2$$
$$P_1 = r/2$$
$$P_2 = r(1 - r/2)$$
$$\dots$$

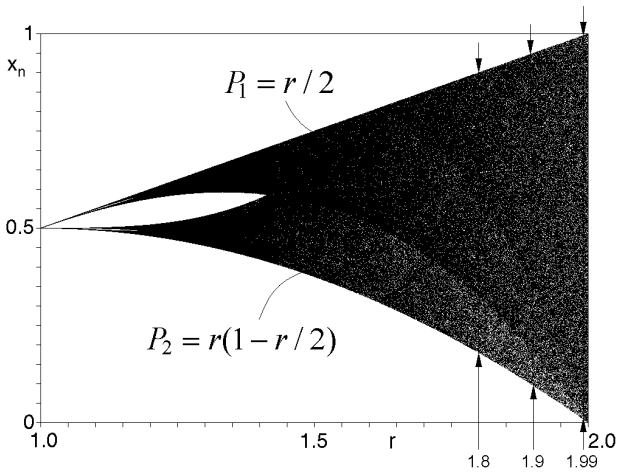

Fig. 5*. Bifurcation diagram of the tent map. The arrows indicate the intervals visited by the orbits for different values of the parameter r.*

Thus, the upper bound of the visited interval for any parameter value is $r/2$, whereas the lower bound is $r(1 - r/2)$. Now we have the necessary tools to carry out the following chosen plaintext attack which again finds the exact value of the secret key r:

1. Starting with the maximum block size, request 1000 times the ciphertext of the word 00…0, of length $b_{\max}$ bits. It will be of the form (b_i, x_{n_i}). In fact, the value of b_i will be much lower, as r departs from 2.0.
2. Using the following formula, compute the corresponding values of the estimation of r:

$$\tilde{r}_i = 1 + \sqrt{1 - 2x_{n_i}}$$

   The maximum value of all the $\tilde{r}_i$'s thus computed corresponds to the exact key.

Histograms can be further used for a chosen plaintext attack on Habutsu's cryptosystem. Simply request 1000 times the ciphertext C_i of the same plaintext P. Due to the use of the random parameter r_i, 1000 different values of C_i will be returned. But as depicted in Fig. 6, there is an accumulation of points towards the exact value of the parameter. The more ciphertexts requested, the more accurate the estimation of α will be. For values of the parameter $0.4 \le \alpha \le 0.6$, this attack method fails to give a good estimation of α, being worse as α gets closer to 0.5, because the points are more uniformly distributed.

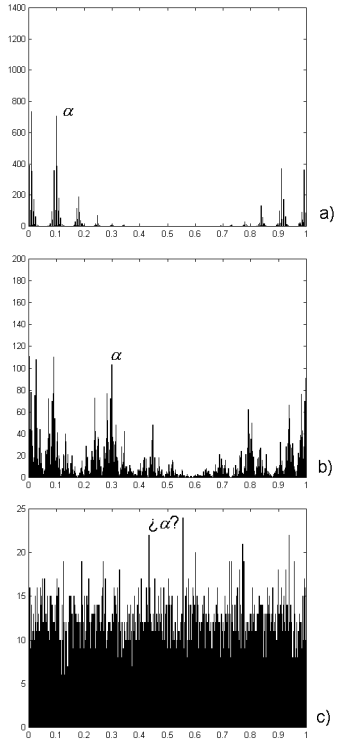

Fig. 6*. Three histograms of C_i for n = 10000 and 1000-point bins. a) $\alpha = 0.1$, b) α = 0.3, and c) α = 0.5.*

5. CONCLUSIONS

We have experimentally proved in this paper that using cryptanalysis tools developed from chaos theory it is possible to break different chaotic cryptosystems based on unimodal maps iteration.

Apart from the weaknesses showed by our attack methods, all these cryptosystems are computationally heavy. For the encryption of a single unit of plaintext, they require times which are some orders of magnitude longer than it would take for a classical algorithm.

When using a non linear dynamical system exhibiting sensitivity to initial conditions and to parameter mismatch, both transmitter and receiver need to use the same machine precision

if the correct plaintext is to be recovered. This requirement implies that, unlike classical cryptographic algorithms, chaos based algorithms require machines at both ends to use the same compiler, precision, number representation, etc. If not, after a certain number of iterations, the orbits followed by both systems (transmitter and receiver), although starting from the same initial point and with same parameter value, will diverge exponentially (this rate of divergence can be estimated by computing the Lyapunov exponent of the system).

The computational high cost, the machine accuracy problem, and the cryptographic weakness, turn down these cryptosystems as serious candidates to outstand number theory based classical algorithms. At most, they can be regarded as curious information concealment methods to frustrate the casual eavesdropper, but in no case the determined attacker. In too many occasions they are so easily broken and in such a short time that no secure application can be found for them.

REFERENCES

[1] E. Álvarez, A. Fernández, P. García, J. Jiménez, A. Marcano, "New approach to chaotic encryption", Phys. Lett. A, 263 (1999) 373-375.

[2] G. Álvarez, F. Montoya, G. Pastor, M. Romera, "Chaotic Cryptosystems", IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology, 332-338, Madrid (1999).

[3] G. Álvarez y Miguel A. F. Sanjuán, "Comunicaciones Seguras utilizando Señales Caóticas", Revista Española de Física, 13(5), 23-27 (1999).

[4] M. S. Baptista, "Cryptography with chaos", Physics Letters A, 240, 50-54 (1998).

[5] D. Erdmann and S. Murphy, "Hénon Stream Cipher", Electronic Letters, 28(9), 893-895 (1992).

[6] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps", Int. J. Bifurc. Chaos, 8(6), 1259-1284 (1998).

[7] T. Habutsu, Y. Nishio, I. Sasase and S. Mori, "A Secret Key Cryptosystem by Iterating a Chaotic Map", Advances in Cryptology, Eurocrypt '91, 127-140 (1991).

[8] L. Kocarev, K. S. Halle, K. Eckert, L. O. Chua, and U. Parlitz, "Experimental demonstration of secure communications via chaotic synchronization", Int. J. Bifurc. Chaos, 2(3), 709-713 (1992).

[9] G. Voyatzis and I. Pitas, "Chaotic Mixing of Digital Images and Applications to Watermarking", European Conference on Multimedia Applications, services and Techniques (ECMAST'96), Louvain-la-Neuve, Belgium (1996).

[10] G. Álvarez, F. Montoya, M. Romera y G. Pastor, "Evaluación de la seguridad de los criptosistemas caóticos", V Reunión Española sobre Criptología, 89-100, Torremolinos (1998).

[11] G. Álvarez, F. Montoya, M. Romera, G. Pastor, "Criptoanálisis de sistema criptográfico basado en la sincronización de osciladores caóticos", Mundo Electrónico, 307, 56-58 (2000).

[12] G. Álvarez, F. Montoya, M. Romera, G. Pastor, "Criptoanálisis de sistema caótico basado en la ergodicidad", VI Reunión Española de Criptología y Seguridad en la Información (2000).

[13] G. Álvarez, F. Montoya, M. Romera y G. Pastor, "Cryptanalysis of a chaotic encryption system", Physics Letters A, 276, 191-196 (2000).

[14] E. Biham, "Cryptanalysis of the chaotic map cryptosystem suggested at EUROCRYPT '91," Advances in Cryptology, Eurocrypt '91, 532-534 (1991).

[15] Th. Beth, D. E. Lazic, A. Mathias, "Cryptanalysis of Cryptosystems based on Remote Chaos Replication", Advances in Cryptology —CRYPTO '94, 318-331, USA (1994).

[16] G. Pérez and H. A. Cerdeira, "Extracting Messages Masked by Chaos", Phys. Rev. Lett., 74(11), 1970-1973 (1995).

[17] G. Álvarez, M. Romera, G. Pastor y F. Montoya, "Gray Codes in 1D Quadratic Maps", Electronics Letters, 34(13), 1304-1306 (1998).

[18] G. Álvarez, M. Romera, G. Pastor, y F. Montoya, "Determination of Mandelbrot Set's Hyperbolic Component Centres", Chaos, Solitons and Fractals, 9(12), 1997-2005 (1998).