Military Technical College
Kobry Elkobbah,
Cairo, Egypt
May 27-29,2008

4th International Conference on
Mathematics and Engineering
Physics (ICMEP-4)

EM-11
Cryptanalysis of a cryptosystem based on chaotic Baker map

Hassan El-Hamouly [a], Gamal Mabrouk [a], Haitham Shabana [a]

Abstract

Recently, the idea of using chaos to design digital ciphers and analog secure communication systems has provoked a great deal of research efforts since early 1990s. Meanwhile, security analysis of various proposed chaotic cryptosystems also attracts increasing attention, and some chaotic cryptosystems have been found insecure. This paper discusses the security weaknesses of a proposed cryptographic algorithm with chaos at the physical level. Some simple linear maps have a well known chaotic behavior in specific interval. But when studding their long-time dynamics with computers, the result is different from the true long-time chaotic dynamics, even in a high-precision floating-point arithmetic. This phenomenon will affect the security of a proposed cryptosystem for practical implementation with finite computing precision and for the use of the iteration number n as the secret key. In addition, we present some possible improvements to the encryption scheme to obtain higher security.

1 Introduction

When we realize some chaotic systems using digital computers, their true long-time dynamics cannot be exhibited at all, even in a high-precision floating-point arithmetic. Although the results cannot be directly generalized to most other chaotic systems, the risk of using digital computers to numerically study continuous dynamical systems is exposed. The essential reason of this case can be attributed to the use of the multiplication factor 2 and its powers. It is because all digital computers are based on binary arithmetic, in which a multiplication with 2i is equal to the left bit-shifting operation << i. A well-known piecewise linear chaotic maps, V-map, $V(x) = 2|x - 0.5|$, is studied to clarify this behaviour. The results on the V-map can be directly extended to other analogue chaotic maps, including the reflected Bernoulli map $f(x) = 1 - (2x \bmod 1)$, and the Baker map (considering Bernoulli shift map is the x-transformation of the Baker map) [2].

Due to the ergodicity and random-like behavior in addition to sensitive dependence on initial conditions and parameters, it is believed that chaos can be used for secure communication efficiently. A growing number of cryptosystems based on chaos have been proposed, many of them fundamentally flawed by a lack of robustness and security. Recently, a cryptosystem based on the chaotic Baker map have been presented [1], which is a scheme that encrypts wave signals. The security defects caused by the Baker map realized in finite precision and the fact that the secret key n can be directly deduced from the ciphertext are discussed.

2 Floating point representation in Digital Computers

For real numbers, there are two kinds of representation formats: fixed-point format, and floating-point format. Fixed point places a radix point somewhere in the middle of the digits, and the floating-point representation - the most common solution - basically represents reals in scientific notation. Floating-point solves a number of representation problems. Fixed-point has a fixed window of representation, which limits it from representing very large or very small numbers. Also, fixed-point is prone to a loss of precision when two large numbers are divided. Floating-point, on the other hand, employs a sort of "sliding window" of precision appropriate to the scale of the number.

IEEE Standard floating point is the most common representation today for real numbers on computers, including Intel-based PC's, Macintoshes, and most Unix platforms. IEEE floating point numbers have three basic components: the sign, the exponent, and the mantissa. The mantissa is composed of the *fraction* and an implicit leading digit. The exponent base (2) is implicit and need not be stored.

The following figure shows the layout for single (32-bit) and double (64-bit) precision floating-point values. The number of bits for each field are shown (bit ranges are in square brackets):

	Sign	Exponent	Fraction	Bias
Single Precision	1 [31]	8 [30-23]	23 [22-00]	127
Double Precision	1 [63]	11 [62-52]	52 [51-00]	1023

To realize a higher simulation precision, generally double-precision is used for the study of chaotic systems. Thus, this paper will focus on double-precision floating-point arithmetic, and briefly call it floating-point arithmetic. Note that the extension from double-precision floating-point arithmetic to single-precision arithmetic is very easy.

Following the IEEE/ANSI floating-point standard, almost all real numbers are stored in the following normalized format:

$$(-1)^{b_{63}} \times (\overbrace{1.b_{51}\ldots b_0}^{\text{mantissa}})_2 \times 2^{(\overbrace{b_{62}\ldots b_{52}}^{\text{exponent}})_2-1023} \qquad (1)$$

where $(\cdot)_2$ means a binary number and $(b_{51} \ldots b_0)_2$ is called the fraction of the mantissa.

The range of positive floating point numbers can be split into normalized numbers (which preserve the full precision of the mantissa), and *denormalized* numbers which use only a portion of the fractions's precision.

	Denormalized	Normalized	Approximate Decimal
Single Precision	$\pm 2^{-149}$ to $(1-2^{-23})\times 2^{-126}$	$\pm 2^{-126}$ to $(2-2^{-23})\times 2^{127}$	$\pm \sim 10^{-44.85}$ to $\sim 10^{38.53}$
Double Precision	$\pm 2^{-1074}$ to $(1-2^{-52})\times 2^{-1022}$	$\pm 2^{-1022}$ to $(2-2^{-52})\times 2^{1023}$	$\pm \sim 10^{-323.3}$ to $\sim 10^{308.3}$

• denormalized numbers: IEEE reserves exponent field values of all 0s and all 1s to denote special values in the floating-point scheme, ±0, ±∞, indeterminate value, NaN (Not a Number), among which ±0 can be considered as two special denormalized numbers. Note that $+0 \neq -$ and $+\infty \neq -\infty$. The five types of special values are stored in the following formats [4, 5]:

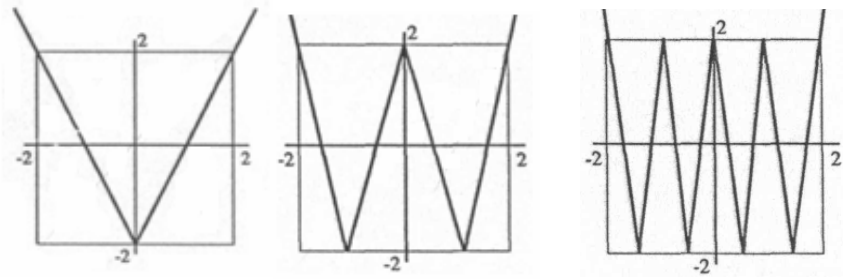
3 Studying the V-Map under Floating-Point Arithmetic

A well-known discrete-time chaotic map is studied to show the incapability of digital computers to compute and represent the chaotic behavior of a well-known discrete-time chaotic map: the V-map [2]. The V-map is defined as

$$V(x) = 2|x| - 2 = \begin{cases} 2x-2, & 0 \leq x \\ -2x-2, & x \leq 0 \end{cases} \qquad (2)$$

which are shown in Fig. 1a. Graphical analysis shows that if $|x| > 2$, then the orbit of x under V tends to infinity. To compute higher iterates of V, we first make use of the definition of the absolute value to write

$$V^2(x) = 2|2|x| - 2| - 2 = \begin{cases} 4|x| - 6, & |x| \geq 1 \\ -4|x| + 2, & |x| \leq 1 \end{cases} = \begin{cases} 4x - 6, & 1 < x \\ -4x + 2, & 0 < x \leq 1 \\ 4x + 2, & -1 < x \leq 0 \\ -4x - 6, & x \leq -1 \end{cases}$$



(a) (b) (c)

Fig. 1: The graph of (a) $V(x)$, (b) V^2, (c) V^3

To prove that V is chaotic on the interval [–2, 2], Fig. 1b, 1c shows the graphs of V^2 and V^3, note that graphs of V^2 consists of four linear pieces, each of slope ±4, and that the graph of V^3 consists of eight pieces, each with slop ±8. In general, the graph of V^nconsists of 2^n pieces, each of which is a straight line with slope $\pm 2^n$. Each of these linear portions of the graph is defined on an interval of length $1/2^{n-2}$.

This fact shows immediately that V is chaotic on [–2, 2]. To see this, we consider an open subinterval J in [–2, 2]. From the above observation, we may always find a subinterval of J of length $1/2^{n-2}$ on which the graph of V^nstretches from –2 to 2 In particular, V^n has a fixed point in J, so this proves that periodic points are dense in [-2,2]. Also, the image of J covers the entire interval [–2, 2], so V is transitive. Finally, to prove the sensitivity on initial conditions we need to find $\beta > 0$ such that for any x and any $\varepsilon > 0$ there is a y within ε of x and a k such that $|V^k(x) - V^k(y)| \geq \beta$. For any $x \in J$, there is a $y \in J$ such that $|V^n(x) - V^n(y)| \geq 2$. Thus we may choose $\beta = 2$ and we have sensitive dependence on initial conditions.

Firstly, assume the initial condition $x_0 = (0.b_2 b_2 \ldots b_j \ldots b_{L-1} b_L)_2 = 0$, where $b_L = 1$ (the least significant 1-bit), $1 - x_0 = (0.\bar{b}_1 \bar{b}_2 \ldots \bar{b}_j \ldots \bar{b}_{L-1} b_L)_2$ and $1 + x_0 = (1.b_1 \ldots b_j \ldots b_{L-1} b_L)_2$. Then, the iteration of the V- map will be

$$x_1 = \begin{cases} 2(x_0 - 1) = (b_1.b_2 \ldots b_j \ldots b_{L-1} b_L)_2, & 1 \leq x \leq 2 \\ 2(x_0 - 1) = -(\bar{b}_1.\bar{b}_2 \ldots \bar{b}_j \ldots \bar{b}_{L-1} b_L)_2, & 0 \leq x < 1 \\ -2(x_0 + 1) = -(\bar{b}_1.\bar{b}_2 \ldots \bar{b}_j \ldots \bar{b}_{L-1} b_L)_2, & -1 \leq x < 0 \\ -2(x_0 + 1) = (b_1.b_2 \ldots b_j \ldots b_{L-1} b_L)_2, & -1 < x \leq -2 \end{cases}$$

Apparently, after L – 1 iterations, $x_{L-1} \equiv (0.b_L)_2 = (0.1)_2$. Then, $x_L \equiv 1$, $x_{L+1} \equiv 0$, $x_{L+2} \equiv -2$, $x_{L+3} \equiv 2$. That is, the number of required iterations to converge to zero is $N_r = L + 3$. Note that $N_r = 0$ when $x_0 = 2$.

From the above analysis, it is clear that no any quantization error is introduced in the digital chaotic iterations, which is because the chaotic iterations can be exactly carried out with the digital operation □.

Generally denormalized numbers will not be used by most pseudo-random number generators, such as the embedded rand function in almost all programming languages, so let us consider the value of L in the condition that x_0 is a normalized number ≠ 0:

Let $x_0 = (1.b_{51} \cdots b_0)_2 \times 2^{-c} = (0.0 \cdots 01 b_{51} \cdots b_0)_2$. Assuming the least 1-bit of x_0 is $b_i = 1$, one can

c-1 52-i i

immediately get $x_0 = (0.0\cdots01\,b_{51}\cdots b_i 0\cdots0)_2$ and deduce L = (c − 1) + 1 + (52 − i) = c + 52 − i. Considering 0 ≤ c ≤ 1022 and 0 ≤ i ≤ 51, 1≤ L ≤ 1074.

We will prove that if x_0 distributes uniformly in the space of all valid floating-point numbers in [-2,2]. That is, the mathematical expectation of N_r is much smaller than 1074 for the V-map.

Assume the mantissa fraction $(b_{51} \ldots b_0)_2$ distributes uniformly over the discrete set $\{0, \ldots, 2^{52}-1\}$. Then, the probability that $(b_i = 1, b_{i-1} = \ldots = b_0 = 0)$ is $\frac{2^{51-i}}{2^{52}} = \frac{1}{2^{i+1}}$, and the probability that $(b_{51} = \ldots = b_0 = 0)$ is $\frac{1}{2^{52}}$. Then, the mathematical expectation of $i \in \{0, \ldots, 51\}$ for a normalized number with a fixed exponent c is

$$E(i) = \sum_{i=0}^{51} i \cdot \frac{1}{2^{i+1}} + 52 \cdot \frac{1}{2^{52}} = \frac{1}{2}\sum_{i=0}^{51} \frac{i}{2^i} + \frac{52}{2^{52}} = \frac{1}{2}\cdot\left(2 - \frac{53}{2^{51}}\right) + \frac{52}{2^{52}} = 1 - \frac{1}{2^{52}} \cong 1$$

Assume x_0 is distributed normally in the interval [-2, 2] with mean 0 and standard deviation 0.5, Then, let us compute the mathematical expectation of c ∈ {1, …, 1022}. The probability of the exponent is c is about $\text{Prob}[2^{-c} \le x < 2^{-c+1}] = \Phi(2^{-c+1}/0.5) - \Phi(2^{-c}/0.5)$. Thus, the mathematical expectation of c can be computed by a simple matlab program:

$$E(c) = \sum_{c=0}^{1022} c\left[\Phi\left(\frac{2^{-c+1}}{0.5}\right) - \Phi\left(\frac{2^{-c}}{0.5}\right)\right] = 1.2084 \cong 1$$

From the above deductions, we can immediately deduce

$$E(L) \approx E(c) + \left(52 - E(i)\right) \approx 1 + 52 - 1 = 52$$

So, $E(L) \approx 52$. That is, $E(N_r) \approx 55 << 1074$ for the V- map, see figure 2and figure 3.

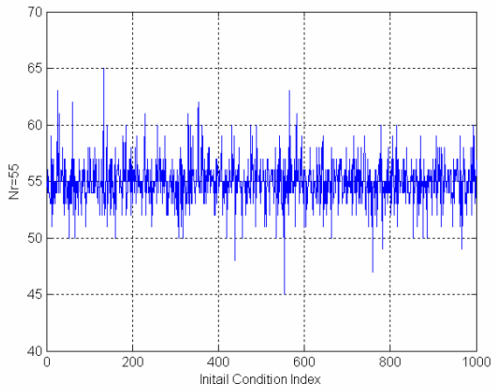


Fig. 2: The values of N_r for 1000 randomly-generated initial conditions

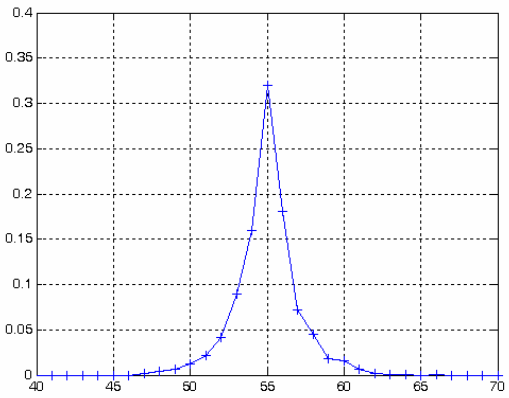


Fig. 3: The occurrence frequency of different values of N_r in the total 1000 values

4 Cryptography with the chaotic Baker map

A proposed chaos-based cryptography scheme designed for digital communication based on the chaotic Baker map was presented, which is a scheme where the encryption is realized at the physical level, that is, a scheme that encrypts the wave signal itself [1].

In the sampling process, a signal varying continuously in time, which is limited in the bandwidth W, is replaced by a set of measurements (samples) taken at instants separated by a suitable time interval provided by the sampling theorem [11,12]. According to the sampling theorem, it is possible to reconstruct the original signal from samples taken at times multiple of the sampling interval $T_S \le 1\backslash 2W$. Thus, at the end of the sampling process, the signal is converted to a sequence $s^0 = \{s_1^0, s_2^0, \ldots, s_i^0\}$ of real values. After being sampled the

signal is quantized. In this process, the amplitude of the signal is divided into N subintervals and every interval is assigned a real amplitude value q_k, $k = 1, \dots, N$, its middle point for example. A new sequence, the y sequence, is generated by replacing each s_i^0 by the q_k associated to the subinterval it belongs to: $y^0 = \{y_1^0, y_2^0, \dots, y_i^0\}$ where each y_i^0 takes its value from the set $\{q_1, \dots, q_N\}$.
Suppose, now, that the amplitude of the wave signal is restricted to the interval [0,1]. The first step of the process is to obtain the chaotic encrypting signal, a sequence $\{x_i^0\}_{i=1}^{i}, 0 < x_i^0 < 1$, is used to generate the ciphertext. This signal is obtained by either sampling a chaotic one or by a chaotic mapping. For the purposes of our analysis, the process to generate the chaotic signal is irrelevant since our results apply equally to any signal. Finally, the ordered pair (x_i^0, y_i^0) is constructed, localizing a point in the unit square. In order to encrypt y_i^0, the Baker map is applied n times to the point (x_i^0, y_i^0) to obtain:

$$(x_i^n, y_i^n) = \left(2x_i^{n-1} \bmod 1, 0.5\left(y_i^{n-1} + \lfloor 2x_i^{n-1} \rfloor\right)\right), \quad (3)$$

where $\lfloor 2x_i^0 \rfloor$ is the largest integer equal to or less than $2x_i^0$. The encrypted signal is given by y_i^n, where n is considered as the secret key of the cryptosystem. As a result, a plaintext signal with values $y_i^0 \in \{q_1, \dots, q_N\}$, is encrypted into a signal which can take $2^n N$ different values.

5 Cryptanalysis

Before starting to analyze the security of the proposed secure encryption scheme, some security guidelines are reviewed. Following the well-known Kerckhoffs' principle in cryptology [13], the security of a cryptosystem should rely on the secret key only, which means that an attacker knows all details about the cryptosystem except for the secret key.
Actually, in some special scenarios, it is possible for an attacker to get some useful information or even intentionally choose some information from the transmitter and/or the receiver. As a result, from the cryptographical point of view, to provide a high level of security, a cryptosystem should be secure enough against all the following four attacks (listed from the hardest to the easiest):
• the ciphertext-only attack - the attacker can only get ciphertexts and other publicly-transmitted information (such as the common driving signal in the scheme under discussion);
• the known-plaintext attack - in addition to some basic information, the attacker can get some plaintexts and the corresponding ciphertexts;
• the chosen-plaintext attack - in addition to some basic information, the attacker can choose some plaintexts and get the corresponding ciphertexts;
• the chosen-ciphertext attack - in addition to some basic information, the attacker can choose some ciphertexts and get the corresponding plaintexts.

The last two attacks, which seem to seldom occur in practice, are feasible in some real applications and have become much more common in today's networked world. In the following, it will be pointed out that the secure communication system under study is not secure enough against these attacks.

5.1 Convergence to zero of the digital Baker map

In this section, the security defects caused by the Baker map realized in finite precision are discussed. The proposed cryptosystem uses the Baker map as a mixing function. The Baker map is an idealized one in the sense that it can only be implemented with finite precision in digital computers and, as a consequence, in this case it will has a stable attractor at (0, 0). This is easy to see when the value of x is represented in binary form with L significant bits. Assuming $x_0 = 0.b_1b_2 \cdots b_j \cdots b_{L-1}b_L$, the Baker map runs as follows:

$$x_1 = 2x_0 \bmod 1 = x_0 << 1 = 0.b_2b_3 \cdots b_j \cdots b_{L-1}b_L0, \quad (4)$$

Apparently, the most significant bit b_1 is dropped during the current iteration. As a result, after m > L iterations, $x^m = 0$. Once $x^m = 0$, it is apparent that within a finite number of iterations y^j will exponentially converge to zero, i.e., the digital Baker map will eventually converge to the stable attractive point at (0, 0) as shown in figure 4. As

we noted in the preceding section, this result does not depend on the real number representation method or the precision.

So, it is expected that each plaintext sample y_i^0 cannot be correctly decrypted when n is greater than 53 (or even smaller but close to 53), since the counter-iterating process is unable to get x_i^0 from $x_i^n = 0$ due to the loss of precision during the forward iterations. It can be appreciated how the plaintext is correctly recovered only when n < 45. For n > 52, the system does not work at all. As a consequence, only n = 45 secret keys have to be tried to break a ciphertext encrypted with this cryptosystem. This takes a modern desktop computer less than a second for moderated lengths of the plaintext. This attack is called a brute-force attack, which breaks a cipher by trying every possible key. The feasibility of a brute force attack depends on the size of the cipher's key space and on the amount of computational power available to the attacker.

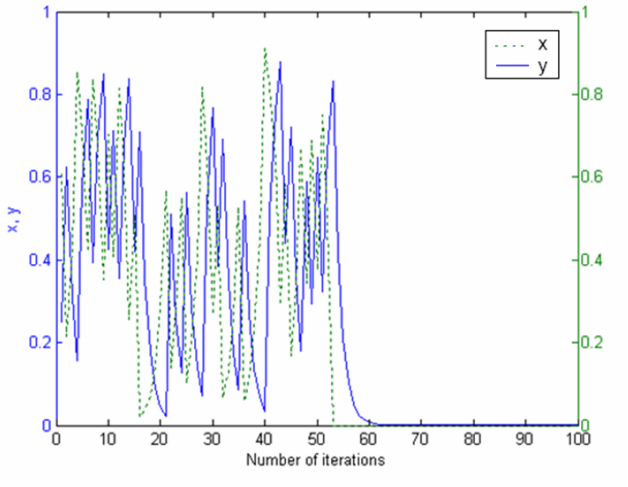


Fig. 4. Orbits of x and y of the Baker map

If the value of n could be arbitrarily enlarged, then the encryption process would slow down until it would be unusable in practice. Thus, from any point of view, this is an impractical encryption method because it is either totally insecure or infinitely slow, without any reasonable tradeoff possible. In [1] it is said that the encryption is applied to the wave signal instead of the symbolic sequence. Therefore, in Table 1 a review of some widely used multimedia communications systems with their bandwidth and sampling frequencies is given. These are the kind of signals that might be encrypted by the system proposed in [1]. Consider for example TV broadcasting, which transmits 12,000,000 samples per second. It is impossible to iterate the Baker map billions of times for 12,000,000 samples in one second with average computing power.

5.2 Determination of the secret number *n*

Here we point out to the fact that the secret key n can be directly deduced from the ciphertext. Suppose, for example, that $N = 2$, and we have $q_1 = 0.25$ and $q_2 = 0.75$. If $s_i^0 < 0.5$ then $y_i^0 = 0.25$ and if we use $n = 1$, we have $y_i^1 = 0.125$ if $x_i^0 < 0.5$ or $y_i^1 = 0.625$ if $x_i^0 \geq 0.5$. On the other hand, if $s_i^0 > 0.5$, then $y_i^0 = 0.75$ and we have $y_i^1 = 0.375$, if $x_i^0 < 0.5$ or $y_i^1 = 0.875$ if $x_i^0 \geq 0.5$. So, the encrypted signal takes on values from the set {0.125, 0.375, 0.625, 0.875}, where the first and third values can be decrypted as 0.25 in the non-encrypted signal while the second and the forth as 0.75. During the encryption process a binary tree is generated in the following way:

$$y_i^0 = \begin{cases} 0.25\ (0.01)_2 \rightarrow y_i^1 = \begin{cases} 0.125\ (0.001)_2 \rightarrow y_i^2 = \begin{cases} 0.0625\ (0.0001)_2 \rightarrow y_i^3 = \begin{cases} 0.03125\ (0.00001)_2 \\ 0.53125\ (0.10001)_2 \end{cases} \\ 0.5625\ (0.1001)_2 \rightarrow y_i^3 = \begin{cases} 0.28125\ (0.01001)_2 \\ 0.78125\ (0.11001)_2 \end{cases} \end{cases} \\ 0.625\ (0.101)_2 \rightarrow y_i^2 = \begin{cases} 0.3125\ (0.0101)_2 \rightarrow y_i^3 = \begin{cases} 0.15625\ (0.00101)_2 \\ 0.65625\ (0.10101)_2 \end{cases} \\ 0.8125\ (0.1101)_2 \rightarrow y_i^3 = \begin{cases} 0.40625\ (0.01101)_2 \\ 0.90625\ (0.11101)_2 \end{cases} \end{cases} \end{cases} \\ 0.75\ (0.11)_2 \rightarrow y_i^1 = \begin{cases} 0.375\ (0.011)_2 \rightarrow y_i^2 = \begin{cases} 0.1875\ (0.0011)_2 \rightarrow y_i^3 = \begin{cases} 0.09375\ (0.00011)_2 \\ 0.59375\ (0.10011)_2 \end{cases} \\ 0.6875\ (0.1011)_2 \rightarrow y_i^3 = \begin{cases} 0.34375\ (0.01011)_2 \\ 0.84375\ (0.11011)_2 \end{cases} \end{cases} \\ 0.875\ (0.111)_2 \rightarrow y_i^2 = \begin{cases} 0.4375\ (0.0111)_2 \rightarrow y_i^3 = \begin{cases} 0.21875\ (0.00111)_2 \\ 0.71875\ (0.10111)_2 \end{cases} \\ 0.9375\ (0.1111)_2 \rightarrow y_i^3 = \begin{cases} 0.46875\ (0.01111)_2 \\ 0.96875\ (0.11111)_2 \end{cases} \end{cases} \end{cases} \end{cases} \tag{5}$$

In a general case, where we apply n iterations of the mapping, $y_i^{\frac{1}{n}}$ can assume $2^n N$ different values. This fact is weakest point, the cryptosystem would be broken as well because the secret key n can be derived from only one known amplitude value of the ciphertext. In eq. (5), it is obvious that y_i^n is always one value in the set

$$\left\{\frac{2i+1}{2^{n+2}}\right\}_{i=0}^{i=2^{n+1}-1} = \left\{\frac{1}{2^{n+2}}, \ldots, \frac{2^{n+2}-1}{2^{n+2}}\right\} \tag{6}$$

The value y_i^n will be represented in the following form:

$$y_i^n = (1.b_1 \cdots b_l)_2 \times 2^{-c} = (0.0 \cdots 01 b_1 \cdots b_l)_2$$

where b_l is the least significant 1-bit. From Eq. (6), one can see that $l + c = n + 2$. Therefore, we can directly derive $n = (l + c) - 2$, by checking which bit is the least significant bit (i.e., the least significant 1-bit) in all bits of y_i^n.

Similarly, for other values of $N = 2^v$, one can easily deduce that $n = (l + c) - (v + 1)$; and for $N \neq 2^v$, the value of n can still be derived easily, but the calculation algorithm depends on how the binary tree shown in Eq. (4) is re-designed.

Although in [1] it is hinted that the value of n could be changed dynamically based on some information of the encrypted trajectory, this idea would not further increase the security of the cryptosystem as long as $2^n N$ different amplitudes are still possible for each different n value. This means that the ciphertext value y_i^n, whatever n_i, can only take values from the finite set defined in Eq. (6) for the given n_i. Hence, for each y_i^n the value of n_i, can be computed as described above and the security is again compromised.

in addition to the discussed security defects of the secret key n, using n as the secret key has another obvious paradox: from the point of view of the security, n should be as large as possible; while from the point of view of the encryption speed, n should be as small as possible.

Apparently, n is not a good option as the secret key.

6 The improved scheme

There are many ways to improve the security of the attacked cryptosystem. As example we may introduce three possible ones: changing the key, changing the chaotic map, and masking the ciphertext with a secret signal.

The secret key could be changed instead of *n* to be the control parameter of the 2-D chaotic map, or the generation parameter of the encryption signal *x*. If the key is chosen to be the control parameter of the 2-D chaotic map, the Baker map has to be modified to introduce some secret control parameters, or another 2-D chaotic map with one or more adjustable parameters has to be used.

There are two alternative definitions of the Baker's map which are in common use. One definition folds over or rotates one of the sliced halves before joining it (similar to the horseshoe map) and the other does not.

The folded baker's map acts on the unit square as

$$B_{folded}(x,y) = \begin{cases} 2x, \frac{y}{2} & for\ \ 0 \le x < 0.5 \\ 2-2x, 1-\frac{y}{2} & for\ \ 0.5 \le x < 1 \end{cases}.$$

When the upper section is not folded over, the map may be written as

$$B_{unrotated}(x,y) = (2x \bmod 1, 0.5(y + \lfloor 2x \rfloor)).$$

The folded baker's map is a two-dimensional analog of the tent map, while the non-rotated map is analogous to the Bernoulli map. Both maps are topologically conjugate. The Bernoulli map can be understood as the map that progressively lops digits off the dyadic expansion of x. Unlike the tent map, the baker's map is invertible.

To overcome the problem of limited number of iterations, the baker map may be modified to use the well-known logistic equation instead of the Bernoulli map

$$F(x,\mu) = \mu x(1-x),$$

where μ is the control parameter. The logistic equation maps the unit interval into itself for $\mu \in [0,4]$. It is known that when $\mu > 3.57$ chaos sets in. So, equation (3) will be modified as follows:

$$(x_i^n, y_i^n) = \left(\mu x_i^{n-1}(1-x_i^{n-1}), 0.5\,(y_i^{n-1} + \lfloor 2x_i^{n-1} \rfloor)\right) \qquad (7)$$

Now, the negative convergence to zero is removed as shown in figure 5, but the secret key n still can be derived from only one known amplitude value of the ciphertext. This security defect can be enhanced if we didn't choose the values of $q_k, k = 1, \dots, N$ to be the middle point of its associate subinterval in the quantization process. The amplitude of the wave signal is restricted to the interval [0, 1] and divide it into N equal ε-intervals $\{[ie, (i+1)e], 0 \le i \le N-1\}$, where $e = 1/N$. Then $q_k, k = 1, \dots, N$ can be calculated as follows:

$$q_k = (k-1)e + me, \qquad 0 \le m \le 1.$$

The parameter μ, the number of iterations n and the parameter m $q_k$are used to as the secret keys in our cryptosystem.

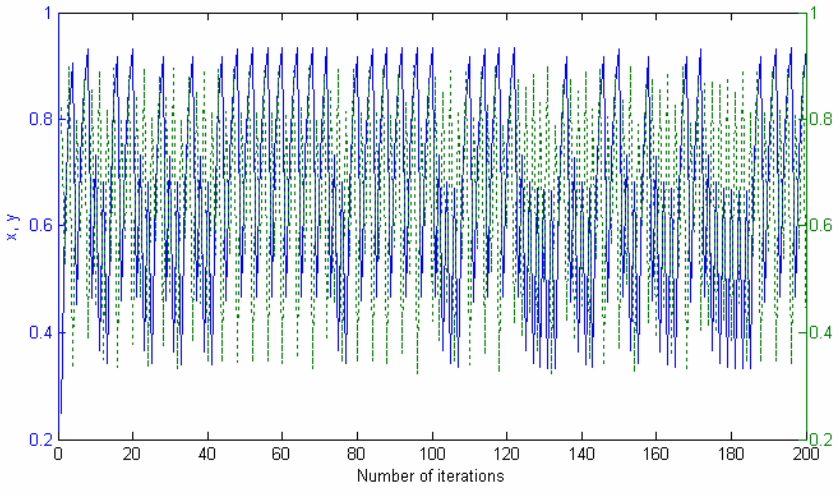


Fig. 5. Orbits of x and y of the modified map (n=200)

Another way to improve the security of the cryptosystem is to mask the ciphertext with a secret signal, this method can overcome the fact that the secret key n can be directly deduced from one amplitude of the ciphertext. The secret masking sequence can be the chaotic encryption signal $\{x_i^0\}$, and the parameters of controlling the generation process of $\{x_i^0\}$ can be added as part of the secret key. In this case, the ciphertext is changed from $\{y_i^n\}$ into $\{y_i^n + x_i^0\}$. This technique is commonly used to achieve stronger ciphers [6].

6 Conclusions

This paper has carefully studied the security of a secure communication scheme published in [1] which is based on chaotic baker map, showing it has security problems. The cryptosystem uses the Baker map as a mixing function. The Baker map is an idealized one , when it is implemented with finite precision in digital computers , it will has a stable attractor at (0, 0) within Nr iterations, and that the value of Nr is uniquely determined by the details of digital floating-point arithmetic. So, we can conclude that an idealized map cannot be used in a practical implementation of a chaos-based cipher. Furthermore, it has been found that the core of this secure communication scheme – the secret key n – is not secure against attacks. Some possible enhancements to the cryptosystem to improve its security are introduced.

References

[1] R. F. Machado, M. S. Baptista, and C. Grebogi. Cryptography with chaos at the physical level. Chaos, Solitons and Fractals, 21(5):1265–1269, 2004.
[2] Dean J. Driebe. Fully Chaotic Maps and Broken Time Symmetry, volume 4 of Nonlinear Phenomena and Complex Systems. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.
[3] Robert L. Devaney. A First Course in Chaotic Dynamical Systems: Theory and Experiment. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1992.
[4] IEEE Computer Society. IEEE standard for binary floating-point arithmetic. ANSI/IEEE Std. 754-1985, August 1985.
[5] Steve Hollasch. IEEE standard 754 floating point numbers. online document at http://stevehollasch.com/cgindex/coding/ieeefloat.html, February 2005.
[6] A. Baranovsky and D. Daems. Design of one-dimensional chaotic maps with prescribed statistical properties. Int. J. Bifurcation and Chaos, 5(6):1585–1598, 1995.
[7] Proakis JG, Salehi M. Communication systems engineering. New Jersey: Prentice Hall; 2002.
[8] D. R. Stinson. Cryptography: Theory and Practice. CRC Press, 1995.
[9] S. Li. When chaos meets computers. arXiv:nlin.CD/0405038, May 2004.
[10] J. Fridrich. Symmetric ciphers based on two-dimensional chaotic maps. Int. J.Bifurcation and Chaos, 8(6):1259–1284, 1998.
[11] Pierce JR. An introduction to information theory, symbols, signals and noise. New York: Dover; 1980.
[12] Proakis JG, Salehi M. Communication systems engineering. New Jersey: Prentice Hall; 2002.
[13] B. Schneier, Applied Cryptography – Protocols, algorithms, and souce code in C, 2nd ed. (John Wiley & Sons, Inc