On the security of a class of discrete-time chaotic cryptosystems

Ercan Solak

National Research Institute of Electronics and Cryptology, Kocaeli, Turkey

Received 29 September 2003; accepted 6 November 2003
Communicated by A.P. Fordy

Abstract

In this Letter we analyze the security of some recently proposed chaotic cryptosystems and give methods to break the cipher to reveal the encrypted information. The method exploits the dependencies between the parameters and the output sequence of a dynamical system to reveal the secretly shared system parameters. For each encryption scheme a known plaintext attack and a ciphertext only attack are given.

© 2003 Elsevier B.V. All rights reserved.

PACS: 05.45.Vx

Keywords: Communication using chaos; Cryptanalysis

1. Introduction

Starting with the observation that chaotic systems can be synchronized through shared signals, the prospect of exploiting chaos for the purpose of secure communication has intrigued many researchers across diverse disciplines [1,2]. Seemingly random behavior of signals produced by chaotic systems brings about a natural similarity to one-time pads and pseudorandom number generators that are some of the basic building blocks in modern cryptography [3]. Another similarity is the one between the exponential divergence of chaotic trajectories starting from nearby initial conditions to the diffusion/confusion properties of digital nonlinear maps as used in block ciphers [4].

Such similarities have further fueled research in chaos-based secure communication schemes and many systems exploiting these peculiarities were proposed in the literature. Common themes include chaotic masking, parameter modulation, chaotic frequency modulation [2,5–7].

Despite the abundance of proposals for chaotic cryptosystems, there seems to be a need for their rigorous analysis against the well-established criteria of security and reliability. As more effort is channelled at cryptanalysis of chaotic cryptosystems and as many existing chaotic encryption schemes are shown to be weak in providing security, new designs can better avoid those pitfalls and stand against possible attacks.
This work aims to analyze a recent chaos based secure communication scheme proposed in [8] and to show that it can easily be broken with a little computational cost. The attacks proposed against the scheme exploit the basic mathematical relations that exist between the output of a dynamical system and its parameters.

The rest of the Letter is arranged as follows; in Section 2 the chaotic system employed in [8] is given, in Section 3 we analyze the case of modulation by multiplication and in Section 4 we examine the security of the case where both multiplication and feedback are used in modulation. The Letter finishes with some concluding remarks.

2. Henon’s map for communication

The Henon map is given by the following discrete-time recursion:

\begin{align}
  x_1(k+1) &= 1 - ax_1^2(k) + x_2(k), \\
  x_2(k+1) &= bx_1(k),
\end{align}

where the parameters $a$, $b$ are chosen to guarantee chaotic operation. These parameters also act as secret keys shared between the receiver and the transmitter. For the purpose of secure communication it is assumed that the chaotic system is sensitive to parameter changes so that anyone who does not possess the accurate knowledge of the secret parameters $a$ and $b$ cannot possibly eavesdrop on the communication.

Two different modulation schemes are proposed in [8]. Both of the schemes use the message sequence to modulate some of the signals of the system (1) and send the resulting output as the ciphertext. At the receiver end, a nonlinear observer is constructed by using the secret parameters. The properties of a global observer guarantees that the states of the observer synchronizes with those of the transmitter system, enabling the legitimate receiver to recover the plaintext message sequence.

In the following discussion we show that both of these encryption schemes can easily be broken to yield the secret parameters and to eavesdrop on the subsequent communication. For each case we propose two attacks; a known plaintext attack where a number of pairs of message and the synchronization signal measurements are available and a ciphertext only attack where the attacker does not have access to the original message used to modulate the system states.

3. Modulation by multiplication

In the first scheme, the first state $x_1$ is chosen as the output of the system (1) and is multiplied by the message $m$. Namely, the transmitter system is given by

\begin{align}
  x_1(k+1) &= 1 - ax_1^2(k) + x_2(k), \\
  x_2(k+1) &= bx_1(k), \\
  s(k) &= x_1(k)m(k),
\end{align}

where $s(k)$ is transmitted over an insecure channel. The message $m$ and the transmitted signal $s$ are also referred to as the plaintext and the ciphertext, respectively.

Note that it is also possible to combine (2a), (2b) into a second order recursion as

\begin{equation}
  x_1(k+1) = 1 - ax_1^2(k) + bx_1(k - 1).
\end{equation}

At the receiver end, an observer based system is used to synchronize with the state $x$ of (2). Having an accurate knowledge of the parameters $a$ and $b$, the legitimate receiver is able to construct the observer so as to recover the original message $m$. 
On the other hand, the attacker’s job is to try to determine the secret parameters \(a, b\) by using the available quantities, which are the sequence \(s(k)\) for a ciphertext-only attack and the pair of sequences \((m(k), s(k))\) for a known plaintext attack. In [8], only the case where the message takes values in the set \([-1, 1]\) is considered. Although such a restriction makes the cryptanalysis of (2) easier, we are going to show that successful attacks can be launched for more general classes of messages.

### 3.1. Known plaintext attack

Suppose that the system (2) starts from the initial state \(x = [x_1(0), x_2(0)]^T\) and the initial segment of the message to be transmitted is \(m = [m(0), m(1), m(2), m(3)]\). In the known plaintext attack, the attacker is assumed to know the message \(m\) as well as the ciphertext \(s\). In most cases this is a reasonable assumption because in almost all communication protocols, message blocks are prepended with some form of header which can be guessed at least partially, [9].

Therefore, after having obtained 4 consecutive message-ciphertext measurement pairs, the attacker can construct the following equations

\[
s(k) = x_1(k)m(k), \quad k = 0, 1, 2, 3.
\]

By solving (4) for \(k = 0, 1, 2, 3\) the attacker can determine the quantities \(x_1(0), x_1(1), x_1(2), x_1(3)\) and substitute those in (3) to obtain the set of linear relations

\[
x_1(2) = 1 - ax_1^3(1) + bx_1(0), \quad x_1(3) = 1 - ax_1^3(2) + bx_1(1),
\]

which can be solved to yield the secret parameters \(a\) and \(b\) as

\[
a = \frac{x_1(1)x_1(2) - x_1(1) - x_1(0)x_1(3) + x(0)}{x_1^2(2)x_1(0) - x_1^3(1)},
\]

\[
b = \frac{x_1^3(2) - x_1^3(2) - x_1^3(1)x_1(3) + x_1^3(1)}{x_1^2(2)x_1(0) - x_1^3(1)}.
\]

Thus, in a known plaintext attack, a message segment of length 4 is enough to find the parameters \(a\) and \(b\). Once these parameters are known, a globally converging observer, which is similar to the one that the legitimate receiver has, can be constructed to decipher the subsequent communication.

Note that we implicitly assumed that it is possible to select four consecutive nonzero message values because if \(m(i) = 0\) for some \(k\), \(0 \leq k \leq 3\), \(x_1(k)\) cannot be uniquely determined. Apart from that, the known plaintext attack explained above does not make any assumptions on the nature of the message such as being generated by a finite alphabet. Indeed, the message sequence can take values from any real set. However, knowing that the message sequence is formed by letters of a finite alphabet enables one to launch a ciphertext only attack, as explained in the next section. Here, we illustrate the known plaintext attack with an example.

**Example 1.** Suppose that, at the transmitter end, we have a copy of the system (2) with the parameters \(a = 1.4\) and \(b = 0.3\). A random binary message of length 40 is encrypted by multiplication by the system output \(x_1\). First four values of the plaintext-ciphertext sequence are used to determine the parameters as \(\hat{a}\) and \(\hat{b}\). Then the following observer is constructed by the attacker to eavesdrop on the subsequent communication:

\[
\hat{x}_1(k + 1) = 1 - \hat{a}s^2(k) + \hat{x}_2(k), \quad \hat{x}_2(k + 1) = \hat{b}\hat{x}_1(k), \quad m(k) = \frac{s(k)}{\hat{x}_1(k)}, \quad k \geq 5.
\]

The decrypted message and the original message are plotted in Fig. 1. Note that after the first few iterations the global observer produces the true state of the system at the receiver, thus resulting in an accurate recovery of the message.
3.2. Ciphertext-only attack

When some partial knowledge about the nature of the plaintext is available to the attacker, it might be possible to launch a ciphertext-only attack. As an illustration of the simplifications that arise, we first analyze the case of a binary alphabet, where \( m(k) \in \{-1, 1\} \). This also corresponds to the one proposed in [8].

The transmitter system is again given by (2). For a message segment of length 4, there are 16 possibilities for the plaintext. For each of these possible plaintext segments, the attacker solves (4) for \( x_1(0), x_1(1), x_1(2), x_1(3) \) and substitutes them in (5) and (6) to obtain one of 16 possible solutions for the secret parameters \( a \) and \( b \). Some of these solutions fall out of the range for chaotic operation and can readily be discarded. Others are tried against a decryption of another piece of intercepted ciphertext sequence and the one that decrypts to a meaningful plaintext is chosen as the true set of parameters.

Although the ciphertext-only attack works well against a binary alphabet, the computational complexity increases exponentially when the alphabet size increases. Indeed, if \( m(k) \) takes values from a finite set of size \( N \), i.e., \( m(k) \in \{r_1, \ldots, r_N\} \), \( r_i \neq 0 \), \( i = 1, \ldots, N \), then \( 4^N \) different solutions need to be computed and tried against a known ciphertext. Obviously, even for moderate alphabet sizes, this makes the attack prohibitively expensive. For example, if we assume the plaintext corresponds to the visible characters in the standard ASCII, so \( N = 96 \), the attack becomes \( 2^{188} \) times harder than it is for the case of a binary alphabet.

4. Modulation by multiplication and feedback

In the second scheme proposed in [8] the difference between the modulated signal (ciphertext) \( s(k) \) and the output \( y(k) \) is fed back into the system dynamics through a gain vector \( L \) which is secretly shared between the transmitter and the intended receiver. Another modification from the multiplication only modulation is that the nonlinear part of the system dynamics takes \( s(k) \) as the argument rather than the output \( y(k) \). When the Henon map (3) is modified accordingly, one obtains the following description of the transmitter system:

\[
\begin{align*}
x_1(k+1) &= 1 - as^2(k) + x_2(k) + l_1(s(k) - y(k)), \\
x_2(k+1) &= bx_1(k) + l_2(s(k) - y(k)), \\
s(k) &= x_1(k)m(k), \\
y(k) &= x_1(k).
\end{align*}
\]
Now there are four secret parameters $a$, $b$, $l_1$, $l_2$ for the attacker to determine before she can eavesdrop on the subsequent communication.

For the purpose of cryptanalysis, it will be useful to rewrite (7) as a recursion in terms of the measurable quantities. By combining (7a), (7b) and (7d), we obtain

$$x_1(k + 1) = -as^2(k) + bx_1(k - 1) + l_2(s(k - 1) - x_1(k - 1)) + l_1(s(k) - x_1(k)).$$

Multiplying both sides by $m(k)m(k - 1)$ and using (7c) yield the desired recursion:

$$m(k)m(k - 1)s(k + 1) = m(k + 1)m(k)m(k - 1) - m(k + 1)m(k)m(k - 1)as^2(k) + bm(k + 1)m(k)s(k - 1)$$

$$+ l_2(m(k + 1)m(k)m(k - 1)s(k - 1) - m(k + 1)m(k)s(k - 1))$$

$$+ l_1(m(k + 1)m(k)m(k - 1)s(k) - m(k + 1)m(k - 1)s(k)).$$

By writing this in vector form we obtain

$$c_1(k) = c_2^T(k)\theta,$$

(8)

where $c_1(k) = m(k)m(k - 1)s(k + 1) - m(k + 1)m(k)m(k - 1)$, the parameter vector $\theta = [a\ b\ l_1\ l_2]^T$ and

$$c_2(k) = \begin{bmatrix}
-m(k + 1)m(k)m(k - 1)s^2(k) \\
-m(k + 1)m(k)m(k - 1)s(k - 1) \\
m(k + 1)m(k)m(k - 1)s(k) - m(k + 1)m(k)m(k - 1)s(k)
\end{bmatrix}.$$

4.1. Known plaintext attack

Assuming that the attacker can measure 6 consecutive (ciphertext, plaintext) pairs, she can combine 4 equations of the form (8) into a matrix equality as

$$C_1 = C_2\theta,$$

(9)

where $C_1 = [c_1(1)\ c_1(2)\ c_1(3)\ c_1(4)]^T$ and $C_2 = [c_2(1)\ c_2(2)\ c_2(3)\ c_2(4)]$. Hence, the unknown parameters are found by the solution $\theta = C_2^{-1}C_1$. Note that both $C_1$ and $C_2$ are in terms of measured quantities, thus, knowing the plaintext-ciphertext pairs $(m(k), s(k))$ for $k = 0, \ldots, 5$ enables the attacker to uniquely construct the matrices $C_1$ and $C_2$. Provided that the resulting matrix $C_2$ is invertible, the secret parameters $a$, $b$, $l_1$, $l_2$ can be uniquely determined.

4.2. Ciphertext only attack

When the ciphertext $s(k)$ is the only data available to the attacker, any combination of relations of the form (8) yields an under-determined set of equations in terms of the unknowns $m$ and $\theta$. In the general case where the message $m(k)$ takes values from a real-valued set, one has a continuum of solutions for $\theta$, which cannot be exhaustively searched for the correct set of parameters. However, when the message $m(k)$ is restricted to take values from a finite set, it might be possible to obtain every solution of (9) for every different message sequence $(m(0), m(1), m(2), m(3), m(4), m(5))$ and exhaustively try each against another segment of ciphertext to see which one decrypts to a meaningful plaintext. This approach is basically the same as the one given in the previous section.

In the binary message case where $m(k) \in \{-1, 1\}$ there are $2^6 = 64$ possible message sequences of length 6 and corresponding to those there are 64 different pairs $(C_1, C_2)$. For example, a particular choice of the message
sequence $m_1 = \{-1, 1, -1, 1\}$ yields the pair

$$C_{1}^{(m_1)} = \begin{bmatrix} -s(2) + 1 \\ s(3) + 1 \\ -s(4) - 1 \\ s(5) - 1 \end{bmatrix}, \quad C_{2}^{(m_1)} = \begin{bmatrix} s^2(1) & s^2(2) & -s^2(3) & -s^2(4) \\ s(0) & -s(1) & s(2) & -s(3) \\ 0 & 0 & 2s(3) & 0 \\ -2s(0) & 0 & 0 & 2s(3) \end{bmatrix}. $$

which are then used to obtain a particular one of the possible 64 parameter vectors as

$$\theta^{(m_1)} = \left[C_{2}^{(m_1)}\right]^{-1} C_{1}^{(m_1)}.$$ 

Each such solution is then tried against another segment of ciphertext to check if it decrypts to a meaningful plaintext.

In general, an attacker would have to try 64 possible parameter vectors. However, certain message segments make $C_2$ singular and hence need to be skipped. Indeed, tedious but straightforward calculation reveals that there are 10 message segments that make $C_2$ singular. If the actual message sent is one of them, then the secret parameters will not be recovered, and another attempt must be made by choosing a different segment of the ciphertext.

Still, some of the remaining 54 possible parameter vectors, when used in the observer, yield unstable observer systems, which clearly indicates that the tried parameter vector does not match the actual parameter vector and should be discarded.

**Example 2.** Suppose that, at the transmitter end, we use both multiplication and feedback to modulate the signals of the chaotic system (2) with the secret parameters $a = 1.4$ and $b = 0.3$. The message to be transmitted is the clown image as specified in [8] and shown in Fig. 2. The gain vectors are chosen as $l_1 = -0.1$ and $l_2 = 0.1$.

In a ciphertext only attack, 11 candidates for the secret parameter vector are determined and for each candidate, the decrypted image is plotted in Fig. 3. It is immediately apparent that the parameter vector $\theta^{(48)}$ is the actual one. The skipped images in Fig. 3 correspond to the cases of either a singular $C_2$ matrix or an unstable observer at the receiver.
5. Conclusion

In this Letter we analyzed the cryptographic strength of a secure communication scheme that uses discrete-time chaos synchronization. We have shown that the scheme is rather weak against simple known plaintext as well as ciphertext only attacks. In particular, in the known plaintext case, the computational burden of the attack is as little as solving a set of linear equations in two unknowns. We have also given methods to implement a ciphertext only attack when the message takes values from a binary alphabet.

References