Available online at www.sciencedirect.com

SCIENCE DIRECT®

Chaos, Solitons and Fractals 30 (2006) 1143–1152

CHAOS
SOLITONS & FRACTALS

www.elsevier.com/locate/chaos

A new chaotic cryptosystem

Jun Wei [a,b], Xiaofeng Liao [a,*], Kwok-wo Wong [c], Tao Xiang [a]

[a] *Department of Computer Science and Engineering, Chongqing University, Chongqing 400044, PR China*
[b] *Zhunyi Medical College, Zhunyi 563000, Guizhou, PR China*
[c] *Department of Computer Engineering and Information Technology, City University of Hong Kong, Hong Kong*

Accepted 1 September 2005

Abstract

Based on the study of some previously proposed chaotic encryption algorithms, we found that it is dangerous to mix chaotic state or iteration number of the chaotic system with ciphertext. In this paper, a new chaotic cryptosystem is proposed. Instead of simply mixing the chaotic signal of the proposed chaotic cryptosystem with the ciphertext, a noise-like variable is utilized to govern the encryption and decryption processes. This adds statistical sense to the new cryptosystem. Numerical simulations show that the new cryptosystem is practical whenever efficiency, ciphertext length or security is concerned.

1. Introduction

The noise-like and deterministic oscillation behavior of chaotic signals has been employed to hide information or to design the S-box used in cryptographic systems [1,2]. Since most chaotic communications based on chaotic synchronization are found not secure enough, the chaotic cryptosystem without the need for synchronization has attracted more and more research interest. Here we present a chaotic cryptosystem with a noise-like variable, which is inspired by the two classes of chaotic cryptosystems proposed in Refs. [3,4] and Refs. [5–13], respectively.

Since Baptista proposed the original version of the so-called Baptista-type chaotic cryptosystem in 1998 [5], attacks to it and its variants have been reported [15–17]. Remedial modifications have also been suggested [6–13]. By assuming that the plain message is composed of N different characters so that a chaotic attractor is accordingly divided into N sites, the Baptista-type algorithm encrypts plaintext into the iteration number performed on a one-dimensional logistic map that governs the cryptosystem in order to reach a site associated with that plain character. The key of the Baptista-type cryptosystem is composed of the above association, the initial condition and the control parameter of the chaotic system.

However, the iteration number sequence contains important and direct information about the chaotic cryptosystem. This can be presented with the fact: when $N \to \infty$, the site associated with a plain character will degenerate into a

[*] Corresponding author.
E-mail address: xfliao@cqu.edu.cn (X. Liao).

0960-0779/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chaos.2005.09.005

chaotic state. Then, from sufficient pairs of plaintext and ciphertext, one can easily find out how many steps the chaotic cryptosystem has to iterate for achieving a target chaotic state. This is equivalent to knowing the key of the cryptosystem. Based on the above fact, the iteration number becomes the security loophole under the attacks proposed in Refs. [14–17].

Alvarez et al. presented another chaotic cryptosystem [3] based on a d-dimensional chaotic equation whose control parameter is the key of the chaotic cryptosystem. In this cryptosystem, by assuming that the plain message is a binary file consisting of a chain of 0s and 1s, each plaintext block is encrypted into a 3-tuple cipher-block (U_i, b_i, X_i) where X_i is a chaotic state from which a plaintext block with size b_i is restored according to such a rule: $x_n \leqslant U_i \to 0$ and $x_n > U_i \to 1$. Apparently, the chaotic state X_i involves useful information to crack the chaotic cryptosystem. Thus, taking it as a security loophole, Alvarez et al. presented a kind of attacks in Ref. [18] to cryptanalyze this chaotic cryptosystem only several months after it is proposed. A remedy is proposed in Ref. [4] to resist the above attack by simply replacing the original ciphertext block (U_i, b_i, X_i) with (U_i, b_i, n_i), where n_i is the iteration number of the chaotic equation departing from the chaotic state after the last plain-block is encrypted. When U_i is a constant, the ciphertext block is reduced to (b_i, n_i) to decrease the *ciphertext size/plaintext size* ratio. Considering that the iteration number sequence actually involves important and direct useful information about the chaotic cryptosystem, the above remedy is not secure enough.

Although chaotic signal appears as noise-like when it is observed as a whole, it actually follows a deterministic rule, which would be discovered if one obtains sufficient iteration details. To let chaotic states, iteration numbers or their simple transform serve as ciphertext or a component of the ciphertext is equivalent to revealing the iteration details of the chaotic system to certain extent, i.e. risks revealing the deterministic rule of the chaotic system. We believe that the above reflection is the inherent cause why the two classes of cryptosystems mentioned above are easily broken. Besides security drawback, we have also observed that both these two classes of cryptosystems not only have very poor efficiency, but their *ciphertext-size/plaintext-size* ratio is much greater than 1, except the one described in Ref. [9].

Inspired by the above studies, we present a new chaotic cryptosystem that takes advantage of some useful properties of chaos favoring cryptography. These properties include sensibility to initial condition, noise-like behavior, ergodicity, and so on. At the same time, we try to avoid as much as possible negative influence of other properties of chaos such as the deterministic rule we describe above.

Without loss of the generality, we assume that the plain message consists of a number of plaintext blocks of size S and generally $S = 2^n$, where n is an integer, in the new cryptosystem. The advantageous property of chaos to cryptography is employed by using the simple one-dimensional logistic map:

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

where $x_n \in [0, 1]$ and the parameter $b\psi$ is chosen so that Eq. (1) behaves chaotically in a continuous interval. Furthermore, b, together with the initial condition x_0 constitutes the key of the cryptosystem.

2. Designing the switch-table of our chaotic cryptosystem

As different values of the control parameter b will result in different natural density distribution of the logistic map (1), it causes negative effect on the efficiency and security of the chaotic cryptosystem as we have learned in Refs. [3–18]. Therefore, we no longer divide the chaotic attractor of Eq. (1) to associate a plain-block with a portion of the attractor. Rather we employ two switch tables (For simplicity, we call it ST in the remaining part of this paper.) built at the two ends of the communication channel. The structure of the two STs are shown in Fig. 1.

At the transmitter of the communication channel, each ST unit includes two components—entry component shows the position index of the unit while value of the unit is in data component. Yet, every ST unit at the receiving end includes three components—entry component shows position index, data component stores value of the unit and index component indicates position index of another unit whose value equals to position index of this unit. For simplicity, $u_t(i)$.data means the data component of the ith unit of the ST at the transmitter. Obviously, $u_t(i)$.entry = i at any case. Thus we can define a rule about the STs. Given $u(i)$.entry ∈ X, then $u(i)$.data ∈ X essentially. A one-to-one map $X \to X$ exists between them while generally there is not any presentable mathematical function relating them. In Fig. 1, the set is $X = \{0, 1, 2, \ldots, 255\}$.

At the transmission end of the communication channel, $u_t(i)$.entry provides entry for the plaintext whose value equals to i and $u_t(i)$.data gives ciphertext of the plaintext i. On the contrary, at the receiving end, $u_r(i)$.entry provides entry for the ciphertext i while $u_r(i)$.index gives plaintext of the ciphertext i. The unsymmetrical structure of the two STs makes it unnecessary to spend time to search the correct output at the receiver of the communication channel.