On The Security of a Spatiotemporal Chaotic Cryptosystem

Rhouma Rhouma\textsuperscript{1}, Belghith Safya\textsuperscript{2}
Syscom laboratory, Ecole Nationale d'Ingénieurs de Tunis, 37, Le belvédère 1002 TUNIS, Tunisia
Email : rhoouma@yahoo.fr\textsuperscript{1}, safya.belghith@enit.rnu.tn\textsuperscript{2}

Abstract—Recently, G.Tang et al. proposed a spatiotemporal cryptosystem based on a one way coupled map lattices (OCML). They claim that it has high security, fast encryption speed and short synchronization transient. This work proves that this scheme has some security weaknesses. Indeed, we propose an attack which the system can’t withstand.

This paper presents an attack on a cryptosystem designed by using a spatiotemporal chaotic system. It’s important to show if a cryptosystem is secure or not. The attack operates like the chosen plaintext attack, it uses two drawbacks in the design of the cryptosystem: first, the generation of the ciphertext is independent of its length and the second is the weakness of encryption operation which uses a modular one; By one pair of (plaintext/ciphertext) we can guess other $2^n - 1$ pairs. The attack is rapid and shows that the cryptosystem is insecure and not suitable for practical use.

I. INTRODUCTION

Modern telecommunication networks, and especially the Internet and mobile-phone networks, have tremendously extended the limits and possibilities of communications and information transmissions. Associated with this rapid development, there is a growing demand of cryptographic techniques, which has spurred a great deal of intensive research activities in the study of cryptography [1, 2].
The appeal of using chaos is mainly due to its random behaviour, sensitivity to initial conditions and parameter settings, which fulfil the classic Shannon requirements of confusion and diffusion [3].

There exist two main approaches of designing chaos-based cryptosystems: analog and digital.

Most analog chaos-based cryptosystems are secure communication schemes designed for noisy channels, based on the technique of chaos synchronization [4]. Chaos synchronization is a technique developed since 1990s. Roughly speaking, it means that two chaotic systems can synchronize with each other under the driving of one or more scalar signals, which are generally sent from one system to another. In chaos-synchronization-based cryptosystems, the information can be transmitted by one or more chaotic signals in a number of ways, including (but not limited to) the following ones: chaotic masking [5-8], chaotic modulation [9-11] and chaotic shift keying [12-13].

Digital chaos-based cryptosystems (also called digital chaotic ciphers), on the other hand, are designed for digital computers, where one or more chaotic maps are implemented in finite computing precision to encrypt the plain-message in a number of ways. The two more recent chaotic cryptosystems are the Baptista-type [14-20] and the Alvarez-type cryptosystems [21, 22]. Such schemes have, both, shown security flaws [23, 26].

On the other hand, spatiotemporal chaotic systems [27-29] have been investigated in secure communication as an alternative to low-dimensional ones [14-22]. Indeed, such hyperchaotic systems are characterized by a great number of positive Lyapunov exponents and are assumed more complex in dynamics. In particular, Guoning Tang et al. have proposed in [28] a chaos communication method based on one way coupled map lattices (OCML). They claim that it has high security, fast encryption speed and short synchronization transient.
This work proves that the scheme proposed in [28] has some security weaknesses. Indeed, we propose an attack for which the system can’t withstand.

Section II of this paper, briefly presents the classical types of attacks. In section III, we present the encryption schemes proposed in [28]. Then, section IV, shows the drawbacks of the cryptosystem and how can they be exploited to maintain an attack. Section V describes the attack algorithm. Section VI draws some simulations results. At last, some conclusions are drawn in section VII.

II. CLASSICAL TYPES OF ATTACKS

When cryptanalyzing a ciphering algorithm, the general assumption made is that the cryptanalyst knows exactly the design and working of the cryptosystem under study, i.e., he knows everything about the cryptosystem except the secret key. This is an evident requirement in today’s secure communications networks, usually referred to as Kerchoffs’ principle [1, p. 24]. According to [1, p. 25], it is possible to differentiate between different levels of attacks on cryptosystems. They are enumerated as follows, ordered from the hardest type of attack to the easiest:

(1) Ciphertext only: the opponent possesses a string of ciphertext.

(2) Known plaintext: the opponent possesses a string of plaintext, \( p \), and the corresponding ciphertext, \( c \).

(3) Chosen plaintext: the opponent has obtained temporary access to the encryption machinery. Hence he can choose a plaintext string, \( p \), and construct the corresponding ciphertext string, \( c \).

(4) Chosen ciphertext: the opponent has obtained temporary access to the decryption machinery. Hence he can choose a ciphertext string, \( c \), and construct the corresponding plaintext string, \( p \).
In each of these four attacks, the objective is to determine the key that was used. It suffices that one of the attacks is feasible to consider an algorithm insecure. The last two attacks, which might seem unreasonable at first sight, are very common when the cryptographic algorithm, whose key is fixed by the manufacturer and unknown to the attacker, is embedded in a device which the attacker can freely manipulate. Daily life examples of such devices are smartcards, electronic purse cards, GSM phone SIM (Subscriber Identity Module) cards, POST (Point Of Sale Terminals) machines, or web application session token encryption.

III. SPATIOTEMPORAL CHAOTIC CRYPTOSYSTEM

Particular attention has been paid recently to spatiotemporal chaotic systems for secure communication [27-29]. One-way coupled map lattices (OCML) is the most common family of spatiotemporal chaotic systems [27]. They are defined as follows:

\[
\begin{align*}
\begin{cases}
x_{n+1}(j) &= (1-\varepsilon) f_j[x_n(j)] + \varepsilon f_{j-1}[x_n(j-1)] \\
x_n(0) &= x_a
\end{cases}
\end{align*}
\]

(1)

Where \(j\) is the space index: \(j = 1...N\); \(n\) is the time index: \(n = 1...L\); \(\varepsilon\) is the coupling coefficient.

\(f(.)\) is a one dimensional chaotic map, as for example the logistic map: \(f(x) = 4x(1-x)\).

Tang et. al. [28] have proposed a cryptosystem scheme based on OCML system and have shown that it is advantageous in terms of security and synchronizing time transient.

The encryption transformation of the proposed scheme is given by:

\[
x_{n+1}(j) = (1-\varepsilon) f_j[x_n(j)] + \varepsilon f_{j-1}[x_n(j-1)],
\]

\[n = 1...L\]

(2a)
\[ f_j(x) = (3.75 + a_j / 4)x(1-x), \]
\[ j = 1 \ldots m. \]

\[ x_{n+1}(m+1) = (1-\varepsilon) f[x_n(m+1)] + \varepsilon f[x_n(m)] \]

\[ Q_n = \text{int}[x_n(m+1) \times 2^\nu] \mod 2^\nu, \quad (2b) \]

\[ Q_n = \text{Sbox}(Q_n), \quad (2c) \]

\[ x_{n+1}(m+1) = Q_n / 2^\nu, \quad (2d) \]

\[ x_{n+1}(j) = (1-\varepsilon) f[x_n(j)] + \varepsilon f[x_n(j-1)], \]

\[ f_j(x) = 4x(1-x), \quad j = m+1 \ldots N, \quad (2f) \]

\[ j = m+1 \ldots N, \quad n = 1 \ldots L \]

\[ x_n(0) = \begin{cases} 
C_n / 2^\nu, & \text{for } \frac{1}{8} \leq C_n / 2^\nu \leq \frac{7}{8}, \\
C_n / 2^\nu + \frac{1}{8}, & \text{for } C_n / 2^\nu < \frac{1}{8}, \\
C_n / 2^\nu - \frac{1}{8}, & \text{for } C_n / 2^\nu > \frac{7}{8}, 
\end{cases} \quad (2g) \]

\[ C_n = (K_n + I_n) \mod 2^\nu \quad (2h) \]

\[ K_n = [\text{int}(x_n(N) \times 2^\nu)] \mod 2^\nu \quad (2i) \]

Where the S-box is defined as follows:

\[ A_1 = [Q_n \gg 24] \& 255, \]

\[ A_2 = [Q_n \gg 16] \& 255, \]

\[ A_3 = [Q_n \gg 8] \& 255, \]

\[ A_4 = [Q_n \& 255], \]

\[ A_0 = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \]

\[ Q_n = [A_0 \ll 24] + [A_1 \ll 16] + [A_3 \ll 8] + A_2 \quad (2j) \]
Where \( I_n \) is the plaintext, \( C_n \) is the ciphertext and \( a = (a_1, a_2, \ldots, a_m) \) are adjustable control parameters serving as the secret key. \( \mu = 52, \nu = 32 \), \( m \) and \( N \) are fixed parameters of the cryptosystem. \( L \) is the length of the plaintext.

The decryption transformation is then:

\[
y_{n+1}(j) = (1 - \varepsilon)f_j[y_n(j)] + \varepsilon f_{j+1}[y_n(j-1)], \quad n = 1 \ldots L
\]

\[
f_j(x) = (3.75 + a_j/4)x(1-x),
\]

\( j = 1 \ldots m. \)

\[
y_{n+1}(m+1) = (1 - \varepsilon)f[y_n(m+1)] + \varepsilon f[y_n(m)],
\]

\[
Q_n = \text{int}\left[y_n(m+1) \times 2^\mu \right] \mod 2^\nu,
\]

\[
Q_n = Sbox(Q_n),
\]

\[
x_{n+1}(m+1) = Q_n / 2^\nu.
\]

\[
y_{n+1}(j) = (1 - \varepsilon)f[y_n(j)] + \varepsilon f[y_n(j-1)],
\]

\[
f_j(x) = 4x(1-x), \quad j = m+1 \ldots N,
\]

\( j = m+1 \ldots N, \quad n = 1 \ldots L \)

\[
y_n(0) = x_n(0),
\]

\[
K_n = \text{int}(y_n(N) \times 2^\mu) \mod 2^\nu
\]

\[
I_n = (C_n - K_n) \mod 2^\nu
\]

By setting \( b_j = a_j, \quad j = 1, \ldots, m \), the receiver can achieve synchronization with the transmitter and correctly extract the message as follows:

\[
y_n(N) = x_n(N) \Rightarrow K_n = K_n' \Rightarrow I_n' = I_n
\]
In some chaotic cryptosystems where more than one parameter is used as part of the key, it is possible to fix one of them and then try to estimate the others using a bit-error-rate (BER) attack, also known as error function attack (EFA) [30]. Tang et al. [28] have evaluated the security level of this system by means of error function attack (EFA) which is thought to be the most effective method for public-structure and known-plaintext cryptosystems. The authors suppose that the intruder knows a segment of the past plaintext, and tries to find the secret key by minimizing the following error function:

\[ e(b) = \frac{1}{T} \sum_{n=1}^{T} |I_n' - I_n| \]  

(5)

The more sensitive the chaos synchronization to parameter mismatch, the more secure the cryptosystem against EFA.

IV. DRAWBACKS OF THE CRYPTOSYSTEM AND HOW TO USE THEM

The weakness of the proposed cryptosystem is due mainly to the fact that:

1. The generation of the ciphertext doesn’t depend on the plaintext length L.
2. The encryption function (2h) is not robust enough.

These drawbacks can be exploited to maintain an attack to decrypt a given ciphertext without knowing the key.

We assume that the source of plaintext emits 256 different symbols, \( s_{256} = \{s_1, s_2, s_3, \ldots, s_{256} \} \). Given a ciphertext C of length L formed by the symbols \( \{c_1, c_2, c_3, \ldots, c_{256} \} \), suppose that we know the corresponding plaintext P. Let’s take the ciphertext \( C_T = (C, c_j) \) of length \( L+1 \) formed by the ciphertext C concatenated with a ciphertext symbol \( c_j \); the keystream \( K_n \) for C and \( C_T \) is the same for \( n=1 \ldots L \).
- To discover the plaintext unit corresponding to the ciphertext $C_T$, we maintain in a scenario of chosen plaintext-attack, the encryption of the plaintexts $M_k = (P, s_i)$ for $k = 1...256$. We obtain 256 ciphertexts $C^k_T = (C, c_k)$ for $k = 1...256$. The desired plaintext unit noted $m \in \{s_1, s_2, s_3, ..., s_{256}\}$ is the symbol which has a ciphertext $c_k$ equal to $c_j$. The plaintext corresponding to the ciphertext $C_T = (C, c_j)$ is $P_T = (P, m)$.

- We can make faster the algorithm of attack by exploiting another weakness of the cryptosystem that is the equation (2h): $C_n = (K_n + I_n) \mod 2^v$

From a given pair of (plaintext / ciphertext) unit, we can deduce other 255 pairs in the case of using the same keystream $K_n$.

We suppose that we will encrypt these symbols one by one: $s_1, s_2, s_3, s_4, ..., s_{256}$

These symbols are integers, if we consider that the alphabet of the plaintext is the first 256 positives integers: $0...255$ corresponding to the 256 symbols: $s_1... s_{256}$, then, for $i = 1...255$:

$$C_i = (K_n + s_i) \mod 2^v$$
$$= (K_n + s_j + (i - j)) \mod 2^v$$
$$= [(K_n + s_j) \mod 2^v + (i - j)] \mod 2^v$$
$$= [C_j + (i - j)] \mod 2^v \quad (6)$$

If we encrypt, for example, the first symbol $s_1$ by computing $C_1$:

$$C_1 = (K_n + s_1) \mod 2^v$$

we find that $C_1 = 150$.

With the same $K_n$, we can deduce that the ciphertext of $s_2$ is:

$$C_2 = (K_n + s_2) \mod 2^v = (C_1 + 1) \mod 2^v = 151$$

And the ciphertext of $s_j$ is $C_j = (C_1 + j) \mod 2^v$.

This means that we can guess another 255 pairs of (plaintext / ciphertext) symbols by knowing just one pair unit.
V. DETAILED DESCRIPTION OF THE ATTACK

We consider this ciphertext: $C = (c_1, c_2, \ldots, c_L) = (247, 204, 73, \ldots, 69)$.

To construct the plaintext $P$, without having the key, we have to firstly find the symbol that corresponds to the “test ciphertext” $C_T = 247$. We save this symbol, secondly, we have to find the second plaintext unit that corresponds to the second unit in the “test ciphertext” $C_T = (247, 204)$. And we continue the process in the same way until we get the whole plaintext (see Figure 1).

Under these assumptions, in a chosen-plaintext attack scenario, we request the ciphertext of this following plaintext: $M = s_1$. We find $C^1_T = 244$. So we can deduce the other 255 ciphertext symbols $\{C^2_T, C^3_T, \ldots, C^{256}_T\} = \{245, 246, \ldots, 299\}$ corresponding to the plaintexts units $\{s_2, s_3, \ldots, s_{256}\}$ by evaluating Equation (6). The desired plaintext unit corresponding to $C_T = 247$ is $P = s_4$.

The next step is to request the ciphertext of this following plaintext: $M = (s_4, s_1)$. We find $C^2_T = (247, 30)$. The same weakness in the modular operation is exploited to deduce the other 255 ciphertext symbols: $\{C^2_T, C^3_T, \ldots, C^{256}_T\} = \{(247, 31), (247, 32), \ldots, (247, 285)\}$ corresponding to the plaintext $\{(s_4, s_2), (s_4, s_3), \ldots, (s_4, s_{256})\}$. We find that the plaintext which has the ciphertext $C_T = (247, 204)$ is $P = (s_4, s_{175})$.

We continue by requesting the ciphertext of $M= (s_4, s_{175}, s_1)$. We find $C^3_T = (247, 204, 65)$. The ciphertext corresponding to the plaintexts $\{(s_4, s_{175}, s_2), (s_4, s_{175}, s_3), \ldots, (s_4, s_{175}, s_{256})\}$ are $\{(247, 204, 66), (247, 204, 67), \ldots, (247, 204, 320)\}$. So we can deduce that the plaintext corresponding to the ciphertext $C_T = (247, 204, 73)$ is $P = (s_4, s_{175}, s_9)$. 
This operation continues in this way until decrypting the whole ciphertext 
\[ C = (c_1, c_2, \ldots, c_L) = (247, 204, 73, \ldots, 69) \]. We will obtain then the plaintext \[ P = (s_4, s_{175}, s_9, \ldots, s_{200}) \].

To make things even simpler, the fourth order symbol source \( S_4 \) is assumed. This means that the source of the plaintext emits four different symbols, \( S_4 = \{s_1, s_2, s_3, s_4\} \). These symbols are integers and for every \( k: s_{k+1} = s_k + 1 \). We suppose that the ciphertext which will be decrypted with the attack algorithm is \( C = (c_1, c_2, c_3, c_4, c_5) = (37, 68, 1, 12, 9) \). So \( L \): the length of the ciphertext is equal to 5 here. Steps leading at the deciphering of \( C \) with the attack algorithm (Figure 1) is listed in Table I, the final plaintext corresponding to \( C \) is \( P = (s_4, s_2, s_4, s_3, s_1) \).

VI. Simulations results

To show the power of the attack, we decipher two different files using the attack algorithm, results show that the attack works perfectly. In Table II, we list the speed in seconds of the attack and the necessary number of times when using the encryptions machine in the attack process (The attack is similar to a “chosen plaintext-attack”: we have a list of chosen plaintext and we can generate their corresponded ciphertext through the encryption machine. So, we need to know the number of time using the encryption machine in the attack process). The simulator for the proposed attack is implemented using Matlab 7.0. Performance was measured on a 1.6 GHz Pentium IV with 752 Mbytes of RAM running Windows XP. The files used for this process are:

File 1: Text (.txt) file of size 3 KB which contains 3071 characters.

File 2: 78×78 image (.bmp) file named Lena of size 8 KB.
VII. CONCLUSIONS

A rapid attack is presented in this paper to cryptanalyze a chaos-based cryptosystem using an OCML function. In addition of the weakness of the cryptosystem, there is another important factor which makes the cryptosystem not suitable for the real applications, that is the size of the generated ciphertext. By taking the parameter $\nu=32$, the ciphertext unit is saved in 32 bits. This mean 8 bits in each plaintext unit correspond to 32 bits in ciphertext. So the size of the final ciphertext will be four times larger than the original plaintext file. There are other versions of the present cryptosystem proposed in [27, 29], the same drawbacks still unchanged, so the present attack can also break these chaos-based cryptosystems by the same way.

REFERENCES


