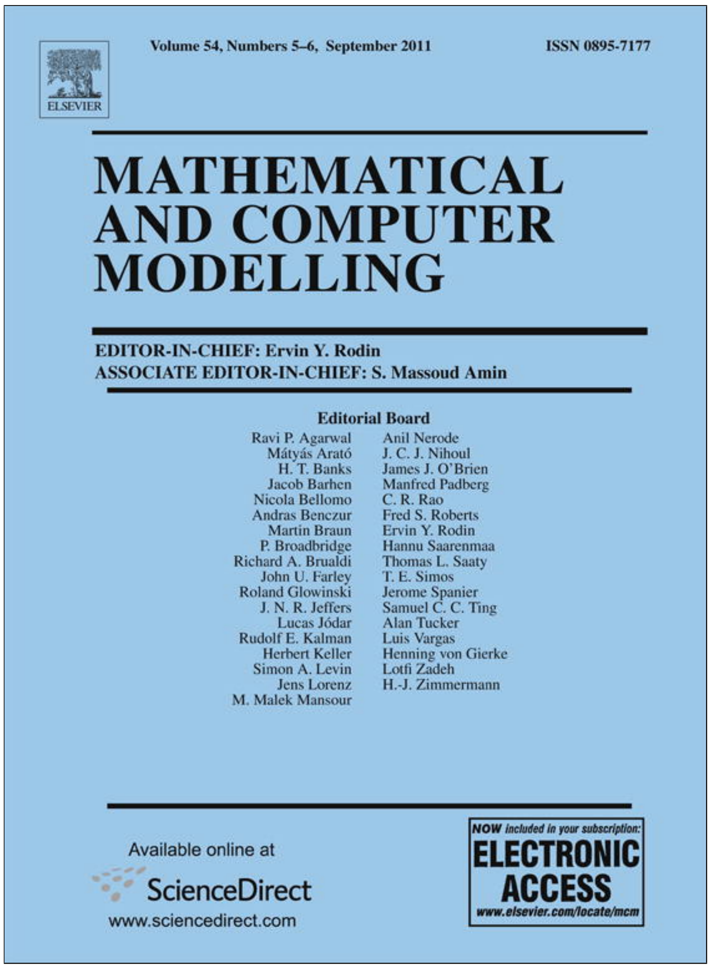

Volume 54, Numbers 5–6, September 2011 ISSN 0895-7177

ELSEVIER

MATHEMATICAL AND COMPUTER MODELLING

Mathematical and Computer Modelling 54 (2011) 1413–1419

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

# Breaking an orbit-based symmetric cryptosystem

Ercan Solak[a,*], Rhouma Rhouma[b], Safya Belghith[b]

[a] *Computer Science and Engineering, Isik University, Turkey*

[b] *Syscom Laboratory, Ecole Nationale d'Ingénieurs de Tunis, Tunisia*

ARTICLE INFO

*Article history:*
Received 30 March 2010
Received in revised form 8 April 2011
Accepted 11 April 2011

*Keywords:*
Orbit
Chaos
Fixed points
Cryptanalysis

ABSTRACT

We report a break for a recently proposed class of cryptosystems. The cryptosystem uses constant points of a periodic secret orbit to encrypt the plaintext. In order to break the system, it suffices to sort the constant points and find the initial fixed point. We also report breaks for modified versions of the cryptosystem. In addition, we discuss some efficiency issues of the cryptosystem.

## 1. Introduction

Chaos-based cryptography has attracted increasing research effort during the last decade. Many proposals have been made with different architectures, including stream ciphers, block ciphers, search-based ciphers, image ciphers, and public-key ciphers. At a higher level of classification, we can separate chaos cryptography into analog and digital architectures. A more elaborate classification can be found in [1].

Chaotic stream ciphers use chaotic systems to generate pseudo-random sequences which are then used to algebraically mask the plaintext sequence. Chaotic block ciphers inject a plaintext block into a chaotic system and use a transformation of the state variables as the ciphertext.

Search-based chaotic cryptosystems were first proposed by Baptista in [2]. In this scheme, a plaintext character is encrypted into the number of iterations required to reach the set corresponding to the particular character. Several variations of Baptista-type cryptosystems have been proposed [3–6]. However, many of these proposals were found to be weak against simple types of attack [7–10].

As chaos-based cryptography has grown through many proposals [11–13], there has also been a dual development of the cryptanalyses of these proposals [7–10]. Some basic frameworks and requirements in the design of chaos-based cryptosystems are investigated in [14,15], and some analyses and new design of digital chaotic ciphers are investigated in [1].

In this work, we cryptanalyze a cryptosystem recently proposed in [16]. The cryptosystem under study uses a nonlinear dynamical system. The cryptosystem exploits the idea of nonlinear mappings and their fixed points to encrypt information. In particular, the plaintext consists of symbols (characters) from a known alphabet (e.g. ASCII), while the ciphertext consists of constant points of the orbit. The secret key of the algorithm is the ordering of the constant points of a periodic orbit. In its most basic form, the chaotic system is iterated as many times as the plaintext character and the state of the chaotic system is sent as the ciphertext.

* Corresponding author.
*E-mail address:* ercan@isikun.edu.tr (E. Solak).

0895-7177/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2011.04.012

Although the core of the cryptosystem under study is a nonlinear dynamical system and it encrypts the plaintext through the iteration of this nonlinear dynamical system, it falls outside the family of digital chaos-based cryptosystems. Indeed, the cryptosystem does not directly exploit the virtues of chaos such as the sensitivity to initial conditions and system parameters.

We show that simple known-plaintext and chosen-plaintext attacks can be launched against the algorithm. For additional discussion of these and other types of attack, see [17, page 41].

The rest of this paper is organized as follows. Section 2 gives a brief description of the cryptosystem under study, with its four versions. Section 3 presents the cryptanalysis for each version. Section 4 comments on some efficiency issues of the cryptosystem. Section 5 gives two simple examples illustrating the attacks. Section 6 discusses the effectiveness of the proposed attacks, and Section 7 concludes the paper.

## 2. Description

In this section, we give a brief description of the cryptosystem proposed in [16].

A periodic orbit of $p$ constant points is generated using a $d$-dimensional chaotic system with a set of secret parameters. $X_0$ is the initial point of the orbit. The orbit length $p$ assumed to be prime. Let $\Phi$ denote the evolution function of the chaotic system; i.e., $\Phi^i(X_0)$ denotes the $i$th point on the orbit starting from $X_0$. Since the orbit is periodic, we have $\Phi^p(X_0) = X_0$.

The key of the encryption algorithm is the ordering of the points $X_0, X_1, \dots, X_{p-1}$ on the orbit.

The plaintext is the sequence of integers $m_1 m_2, \dots, m_n$. We can assume that each $m_i$ is an element of the alphabet $\mathbf{Z}_{256}$ which corresponds to the ASCII codes of all the 256 characters. The sequence $C_1 C_2, \dots, C_n$ denotes the ciphertext sequence. Each $C_i \in \mathbf{R}^d$ represents a constant point on the orbit. Four versions of the algorithm have been proposed.

### 2.1. Version 1: Basic version of the cryptosystem

In the basic version, the ciphertext is calculated by

$$C_i = \Phi^{k_i}(X_0), \quad 1 \le i \le n, \tag{1}$$

where $k_i = \left(\sum_{j=1}^{i} m_j\right) \bmod p$.

### 2.2. Version 2

In this version, there are two more secret parameters, $\beta$ and $\gamma$. We also have a known hash function $H$. The ciphertext is calculated as

$$\begin{aligned} C_1 &= \Phi^{(\beta H(m)+m_1) \bmod p}(X_0), \\ C_i &= \Phi^{(\gamma m_{i-1}+m_i) \bmod p}(C_{i-1}). \end{aligned} \tag{2}$$

### 2.3. Version 3

This time, we use a random number generator that is seeded with $\beta H(m)$. Let $r_1, r_2, \dots, r_n$ denote the sequence of random numbers generated thus. Each $r_i$ is an integer smaller than $p$. There are two modes: stream cipher and non-stream cipher. The encryption is different for each case.

*Stream cipher*:

$$C_i = \Phi^{(\gamma r_i+m_i) \bmod p}(X_0).$$

*Non-stream cipher*:

$$\begin{aligned} C_1 &= \Phi^{(\gamma r_1+m_1) \bmod p}(X_0), \\ C_i &= \Phi^{(\gamma r_i m_{i-1}+m_i) \bmod p}(X_0). \end{aligned}$$

### 2.4. Version 4

This version can be applied to any of the previous versions. Instead of sending $C_i$ as the ciphertext, we send $\overline{C}_i = C_i \oplus X_0$.

## 3. Cryptanalysis

For each version of the cryptosystem, we propose simple attacks.

### 3.1. Version 1

#### 3.1.1. Known-plaintext attack

Assume that the attacker knows a sufficiently long plaintext sequence $m_1 m_2, \dots, m_n$ and its corresponding ciphertext $C_1 C_2, \dots, C_n$.

Assume that the attacker observes a repetition in the ciphertext, i.e., $C_{n_1} = C_{n_2}$ for some integers $n_1 < n_2$. Then, by (1), the attacker knows that

$$\sum_{j=1}^{n_1} m_j \equiv \sum_{j=1}^{n_2} m_j \pmod{p},$$

or

$$\sum_{j=n_1+1}^{n_2} m_j \equiv 0 \pmod{p}. \tag{3}$$

By observing a few such repetitions in the ciphertext, the attacker constructs partial sums of plaintext as in (3), which are all multiples of the secret $p$. By finding the common prime divisor of these partial sums, the attacker reveals $p$.

Another way to reveal $p$ is to count the number of distinct symbols in the ciphertext. Since there are $p$ distinct points on the secret orbit, for a sufficiently long and rich ciphertext, we expect to encounter all of these symbols. Hence, a simple counting reveals $p$.

Once the attacker reveals the secret length $p$ of the orbit, he/she then attempts to reveal the ordered points $X_0, \Phi(X_0), \Phi^2(X_0), \dots, \Phi^{p-1}(X_0)$ of the orbit.

In order to reveal $X_0$, the attacker searches for an integer $n_3 < n$ such that $\sum_{j=1}^{n_3} m_j \equiv 0 \pmod{p}$. Using (1), the attacker obtains

$$X_0 = C_{n_3}.$$

For the rest of the points on the orbit, the attacker uses

$$\Phi^{\left(\sum_{j=1}^{i} m_j\right) \bmod p}(X_0) = C_i, \quad 1 \le i \le n. \tag{4}$$

If the plaintext is sufficiently long and rich that, for every $k \in \{1, 2, \dots, p-1\}$, there exists an integer $i_k$ such that

$$k \equiv \sum_{j=1}^{i_k} m_j \pmod{p},$$

then the attack reveals the whole orbit.

### 3.1.2. *Chosen-plaintext attack*

The attacker chooses the simple plaintext of $n$ characters as

$$m_i = 1, \quad 1 \le i \le n$$

and obtains the corresponding ciphertext $C_1 C_2, \dots, C_n$. By the periodicity of the orbit, the attacker will observe $C_0 = C_k$ for some $k$. The smallest such $k$ is the secret orbit length $p$. Furthermore, using (1) with this particular choice of plaintext, the attacker knows that

$$\Phi^i(X_0) = C_i, \quad 1 \le i \le p$$

and

$$X_0 = C_{p+1}.$$

## 3.2. *Version 2*

### 3.2.1. *Non-invertibility of the encryption*

An encryption algorithm must be invertible, otherwise it is impossible to decrypt the ciphertext and obtain the corresponding plaintext. However, depending on the particular choice of the hash function $H$, the second version of the algorithm might turn out to be non-invertible. Before illustrating the cases when the encryption might be non-invertible, we derive an equivalent expression for Version 2.

Using (2) repeatedly, we can express the encryption in Version 2 as

$$C_i = \Phi^{k_i}(X_0), \quad 1 \le i \le n, \tag{5}$$

where

$$k_i \equiv \beta H(m) + (1+\gamma) \sum_{j=1}^{i-1} m_j + m_i \pmod{p}. \tag{6}$$

Assume that we have two distinct plaintexts $m \ne \overline{m}$ and that each contains a single symbol, i.e., $m = m_1$ and $\overline{m} = \overline{m}_1$. Using (5) and (6), we see that the single-symbol ciphertexts $C_1$ and $\overline{C}_1$, corresponding to $m_1$ and $\overline{m}_1$, are equal when we have

$$\beta H(m_1) + m_1 \equiv \beta H(\overline{m}_1) + \overline{m}_1 \pmod{p}. \tag{7}$$

Hence, if (7) has a solution where $m_1 \ne \overline{m}_1$, then the encryption is not invertible because two distinct plaintexts are encrypted to the same ciphertext.

Since all the calculations in (7) are in $\bmod p$, we can treat the hash function $H$ as mapping from the set of all messages to the set $\{0, 1, 2, \ldots, p-1\}$. For the sake of illustration, let us choose the hash function as

$$H(x) = \alpha x \bmod p,$$

where $\alpha$ is such that $\alpha\beta \equiv p-1 \pmod p$. With this choice of $H$, (7) is satisfied for all $m_1$ and $\overline{m}_1$. Therefore, the encryption is not invertible.

When an arbitrary hash function is used, it is not difficult to find plaintext pairs for which the encryption is not invertible. In [16], MD5 is suggested as a possible hash function. Here, we provide a simple example in which two distinct plaintext strings are mapped to the same ciphertext.

Let us choose $\beta = 85$, $\gamma = 17$ and $p = 256$. Let $m$ be the string `"vi"` and $\overline{m}$ be the string `"uz"`. The MD5 values of these two strings are given in hexadecimal notation as

```
MD5("vi") = 35b36b28916d38b34abddf832e286126

MD5("uz") = 8b3274b755aa033902f57fb557e25923
```

Substituting these in (6) with the ASCII values of `v,i,u,z` as 118, 105, 117, 122, we obtain $k_1 = \overline{k}_1 = 20$ and $k_2 = \overline{k}_2 = 83$. Therefore, the plaintext strings `"vi"` and `"uz"` are encrypted to the same ciphertext. Note that, in this example, we deliberately chose $p = 256$, so that in $\bmod p$ we use only the last two digits of hashes, and the calculations can be done easily. Obviously, $p$ is not prime. However, this is irrelevant as far as the invertibility is concerned.

Thus, it is crucial that the hash function is chosen properly to make sure that the encryption is invertible. For the rest of the paper, we assume that such a hash is used.

Note that the hash $H(m)$ needs to be known by the receiver so that he/she can decrypt the ciphertext. Hence, the hash value must be transmitted along with the ciphertext. If MD5 is used as the hash function, 16 bytes need to be transmitted. For a long stream of plaintext message, this brings about a negligible overhead.

### 3.2.2. *Known-plaintext attack*

Assume that the attacker knows a sufficiently long plaintext $m_1 m_2, \ldots, m_n$ and its corresponding ciphertext $C_1 C_2, \ldots, C_n$. Assume that the attacker already knows the period $p$ of the orbit. Even if the attacker does not already know $p$, he/she can simply count the number of distinct symbols in the ciphertext to find it.

Assume that the attacker observes a repetition in the ciphertext, i.e., $C_{n_1} = C_{n_2}$ for some integers $n_1 < n_2$. Then, by (5) and (6), the attacker knows that

$$\beta H(m) + (1+\gamma)\sum_{j=1}^{n_1-1} m_j + m_{n_1} \equiv \beta H(m) + (1+\gamma)\sum_{j=1}^{n_2-1} m_j + m_{n_2} \pmod p,$$

or

$$\gamma \equiv \left(m_{n_1} - m_{n_2}\right)\left(\sum_{j=n_1}^{n_2-1} m_j\right)^{*} - 1 \pmod p, \tag{8}$$

where $u^*$ denotes the modular inverse of $u$, i.e., $uu^* \equiv 1 \pmod p$.

Now that the attacker knows the secret $\gamma$, he/she goes on to reveal $\beta$. This time, assume that the attacker knows two distinct plaintexts $m$ and $\overline{m}$ and their corresponding ciphertexts $C$ and $\overline{C}$. Also assume that the attacker observes a coincidence of two ciphertexts at indices $n_3$ and $n_4$, i.e., $C_{n_3} = \overline{C}_{n_4}$. Using (5) and (6), the attacker knows that

$$\beta H(m) + (1+\gamma)\sum_{j=1}^{n_3-1} m_j + m_{n_3} \equiv \beta H(\overline{m}) + (1+\gamma)\sum_{j=1}^{n_4-1} \overline{m}_j + \overline{m}_{n_4} \pmod p. \tag{9}$$

Since the hash function $H$ is a known public function, the attacker reveals $\beta$ as

$$\beta \equiv (H(m) - H(\overline{m}))^{*}\left((1+\gamma)\left(\sum_{j=1}^{n_4-1} \overline{m}_j - \sum_{j=1}^{n_3-1} m_j\right) + \overline{m}_{n_4} - m_{n_3}\right) \pmod p.$$

Once the attacker reveals $\gamma$ and $\beta$, the attack proceeds to reveal the order of the points on the orbit.

In order to reveal $X_0$, the attacker searches for an integer $n_5 < n$ such that $\beta H(m) + (1+\gamma)\sum_{j=1}^{n_5-1} m_j + m_{n_5} \equiv 0 \pmod p$. Using (5), the attacker obtains

$$X_0 = C_{n_5}.$$

For the rest of the points on the orbit, the attacker uses

$$\Phi^{\left(\beta H(m)+(1+\gamma)\sum_{j=1}^{i-1} m_j + m_i\right) \bmod p}(X_0) = C_i, \quad 1 \le i \le n \tag{10}$$

and populates the ordered orbit with revealed points.

### 3.3. Version 3

The security of this version of the algorithm relies on the security of the random number generator (RNG) that is used to produce the sequence of integers $r_i$. If there is a detectable correlation among the generated numbers $r_i$, then this will reflect as a weakness of the encryption algorithm. However, no details about the structure of this RNG was provided in [16]. Therefore it is impossible either to analyze or to establish trust on the security of this version.

In order to better see why the security relies on the RNG, assume that we simply XOR the random numbers with the plaintext to generate the ciphertext, i.e.,

$$C_i = m_i \oplus r_i, \quad 1 \leq i \leq n.$$

This form of encryption is as close as one could get to a perfect cipher provided that the RNG is good and that the random number sequence $\{r_i\}$ is never reused; see [17]. However, if the attacker is able to establish the existence of a correlation among different portions of the sequence $\{r_i\}$, a whole range of attacks becomes possible. In order to be able to analyze the RNG and make sure that it is strong against such attacks, its algebraic structure must be specified.

### 3.4. Version 4

The only modification in this version is a one-to-one replacement of ciphertext symbols $X_i$ by $\overline{X}_i = X_i \oplus X_0,\ 1 \leq i \leq p$. Such a modification does not contribute to the security of the encryption algorithm. Indeed, the attacker is only interested in the ordering of the ciphertext symbols, not in their particular values. Therefore, the attack for this version tries to order $\overline{X}_i$ instead of $X_i$. Attacks that work for other versions work for this version with only trivial modifications.

Interestingly, such a replacement of ciphertext symbols leaks some information about the underlying ordering on the orbit. Indeed, the new symbol for $X_0$ is $\overline{X}_0 = X_i \oplus X_0 = 0$. Hence, the attacker knows the initial point of the orbit to be 0.

## 4. Efficiency issues

### 4.1. Representation of ciphertext

Other than the security problems identified in the previous sections, the cryptosystem in [16] suffers from an efficiency problem. The secret orbit lies in a $d$-dimensional state-space. Hence, the algorithm encrypts every byte of a plaintext with $d$ floating point numbers. If double precision is used for floating point numbers, $8 \times n \times d$ bytes of ciphertext is generated for $n$ bytes of plaintext. Obviously, this leads to an inefficient use of the communication bandwidth.

### 4.2. Choice of the periodic orbit

The cryptosystem proposed in [16] relies on the secrecy of the ordering of a large periodic orbit. Obviously, the behavior of the orbit depends on the particular nonlinear system and the values of its control parameters. For the designer of such a cryptosystem, the first task is the choice of a nonlinear dynamical system that has a rich set of periodic orbits. Second, the designer has to make sure that the range of control parameters allows for an efficient determination of required periodic orbits.

### 4.3. Finite-precision arithmetics

The underlying nonlinear map $\Phi$ in (1) maps between continuous values. However, its digital implementation necessarily uses finite-precision arithmetic. For communication parties to use the exact same orbit, the precision must be specified in a complete description of the cryptosystem. Otherwise, different implementations of the same underlying system might end up with different orbits.

## 5. Simulations

In this section, we provide simple examples to illustrate the success of our attacks on the proposed cryptosystem. In order to simplify the discussion, we use letters for ciphertext symbols rather than $d$-dimensional floating point numbers.

### 5.1. Known-plaintext attack on Version 1

Assume that the following plaintext and ciphertext sequences are known by the attacker.

$$m = 4, 3, 2, 1, 4, 6, 1, 1, 2, 0, 4, 2, 6, 2, 4, 1, 2, 1, 5, 6, 2, 4, 2, 3, 5, 4, 2, 2, 3, 2$$

$$C = \mathtt{d, g, e, k, c, e, k, j, i, i, a, f, c, b, e, k, h, i, g, i, d, f, k, i, g, j, i, d, g, e}.$$

The first symbol d in the ciphertext is repeated at the 21st place. Hence the secret period $p$ is a factor of $\sum_{i=2}^{21} m_i = 55$. The second symbol g repeats at 19th place. Hence, $p$ is a factor of $\sum_{i=3}^{19} m_i = 44$. At this point, the attacker reveals the secret $p$ as $\gcd(55, 44) = 11$.

Now, the attacker searches for $n_3$ such that $\sum_{i=1}^{n_3} m_i \equiv 0 \pmod{11}$. Inspecting the plaintext sequence, we see that $n_3 = 8$. Therefore, $X_0 = C(8) = \mathtt{j}$. Using (4) for $i \leq 17$ reveals the secret order of the orbit as `jhicdbagfek`.

### 5.2. *Known-plaintext attack on Version 2*

For this example, we assume that MD5 is used as the hash function. Each element of the plaintext is taken as the ASCII code in the calculation of the 128 bit MD5 value. For MD5 calculations, we used the MATLAB package `CalcMD5` [18], and for integer arithmetic with variable precision, we used the MATLAB package `vpi` [19].

Assume that the attacker already knows that $p = 11$. Also, assume that the attacker knows the following plaintext and ciphertext pairs.

$$m = 6, 5, 5, 0, 0, 0, 5, 6, 4, 0, 5, 0, 0, 4, 2, 4, 5, 4, 5, 1, 5, 6, 6, 0, 2, 2, 4, 4, 5, 2$$
$$C = \mathtt{c, i, d, i, i, i, a, k, j, b, g, h, h, k, k, f, c, a, d, e, i, c, e, c, i, e, j, f, c, k}$$
$$\overline{m} = 1, 0, 5, 1, 2, 3, 1, 4, 3, 1, 5, 0, 2, 1, 3, 0, 2, 0, 0, 5, 2, 4, 6, 3, 4, 5, 3, 4, 0, 6$$
$$\overline{C} = \mathtt{i, a, h, k, h, e, f, j, a, h, c, e, k, k, i, j, f, a, a, h, j, g, i, i, d, i, f, k, e, g}.$$

Knowing that MD5 is used, the attacker calculates the hash values as

$$\mathrm{MD5}(m) \equiv 2 \pmod{11},$$
$$\mathrm{MD5}(\overline{m}) \equiv 7 \pmod{11}.$$

Next, the attacker sees that $C(1) = C(17)$. Using (8) with $n_1 = 1$ and $n_2 = 17$, he/she reveals the secret $\gamma = 5$.

Comparing $C$ and $\overline{C}$, the attacker sees that $C(1) = \overline{C}(11)$. Using (9) with $n_3 = 1$ and $n_4 = 11$, the attacker reveals $\beta = 8$.

Knowing the values of $\gamma$ and $\beta$, the attacker uses (10) to reveal the secret orbit as `chiebkjafgd`.

## 6. Effectiveness and complexity of the attacks

The attacks that we have proposed rely on finding repetitions in the ciphertext. However, if the orbit length $p$ is very large and the sequence of available plaintext is short, then, obviously, the attacks cannot succeed.

However, no specifications are given in the original proposal of [16] regarding the range of $p$. The typical range will necessarily depend on the particular choices of nonlinear mapping $\Phi$, its parameters, and initial conditions. Still, the periodicity of the key orbits must be verified before the system can be used in encryption. Indeed, the stability checking algorithm given in Section 3 of the original proposal [16] includes a loop that runs $p - 1$ times. Hence, it might be safe to conclude that, in practice, $p$ is not so big as to prevent a ciphertext repetition for practically long plaintext.

Here, we use a result that gives the expected length of the known-plaintext sequence for the attack to succeed.

Consider an infinite random sequence $m_1 m_2 \ldots$, where each symbol $m_i$ is randomly drawn from the set $\{0, 1, 2, \ldots, k\}$. Let us define the partial sum $s_n$ as

$$s_n = \sum_{j=1}^{n} m_j.$$

The expected waiting time until we have $s_n \equiv 0 \pmod{p}$ is $p$ [20]. This gives us an estimate on the order of the length of the plaintext sequence that an attacker needs to carry out the attack given in Section 3.1.1.

In order to find the first repeating symbol in a sequence of length $p$, the attacker can use any of the well-known methods such as sorting the sequence or using hashes. In this case, the attack complexity is less than $p \log p$, [21, chapters 8 and 11]. Such a low complexity makes our attacks quite feasible for practical ranges of $p$.

## 7. Conclusion

We have reported known-plaintext and chosen-plaintext attacks for different versions of a recently proposed cryptosystem. We have shown that the cryptosystem is extremely weak against those attacks. The break of the cryptosystem consists of finding the secret sorting sequences of the periodic orbit and the initial fixed point. The computational burden of the attack is so low that they can be even performed using pen, paper, and a simple calculator. We have provided simple examples to illustrate the steps of the attacks.

We also discussed the efficiency and implementation issues of the original cryptosystem.

We conclude that neither the original version of the cryptosystem nor its three modified versions are suitable for secure communications.

## References

[1] S. Li, Analyses and new designs of digital chaotic ciphers, Ph.D. Thesis, Xian Jiaotong University, China, June 2003.
[2] M.S. Baptista, Cryptography with chaos, Physics Letters. A 240 (1–2) (1998) 50–54.
[3] S. Li, G. Chen, K.-W. Wong, X. Mou, Y. Cai, Baptista-type chaotic cryptosystems: problems and countermeasures, Physics Letters. A 332 (5–6) (2004) 368–375.
[4] K.W. Wong, A fast chaotic cryptographic scheme with dynamic look-up table, Physics Letters A 298 (4) (2002) 238–242.
[5] K.W. Wong, S.-W. Ho, C.-K. Yung, A chaotic cryptography scheme for generating short ciphertext, Physics Letters A 310 (1) (2003) 67–73.

[6] M. Ariffin, M. Noorani, Modified Baptista type chaotic cryptosystem via matrix secret key, Physics Letters A 372 (2008) 5427–5430.
[7] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of an ergodic chaotic cipher, Physics Letters A 311 (2–3) (2003) 172–179.
[8] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of dynamic look-up table based chaotic cryptosystems, Physics Letters A 326 (2004) 211–218.
[9] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Keystream cryptanalysis of a chaotic cryptographic method, Computer Physics Communications 156 (2004) 205–207.
[10] R. Rhouma, E. Solak, D. Arroyo, S. Li, G. Alvarez, S. Belghith, Comment on modified Baptista type chaotic cryptosystem via matrix secret key, Physics Letters A 373 (37) (2009) 3398–3400.
[11] V. Patidar, N.K. Pareek, K.K. Sud, A new substitution–diffusion based image cipher using chaotic standard and logistic maps, Communications in Nonlinear Science and Numerical Simulation 14 (2009) 3056–3075.
[12] V. Patidar, N.K. Pareek, G. Purohit, K.K. Sud, Modified substitution–diffusion image cipher using chaotic standard and logistic maps, Communications in Nonlinear Science and Numerical Simulation 15 (10) (2010) 2755–2765.
[13] V. Rozouvan, Modulo image encryption with fractal keys, Optics and Lasers in Engineering 47 (1) (2009) 1–6.
[14] D. Arroyo, Framework for the analysis and design of encryption strategies based on discrete-time chaotic dynamical systems, Ph.D. Thesis, ETSIA of the Polytechnic University of Madrid, Spain, July 2009.
[15] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, International Journal of Bifurcation and Chaos 16 (8) (2006) 2129–2151.
[16] M. Vrahatis, G. Tsirogiannis, E. Laskari, Novel orbit based symmetric cryptosystems, Mathematical and Computer Modelling 51 (2010) 239–246.
[17] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.
[18] J. Simon, CalcMD5. http://www.mathworks.com/matlabcentral/fileexchange/25921.
[19] J. D'Errico, vpi. http://www.mathworks.com/matlabcentral/fileexchange/22725.
[20] M. Brown, On two problems involving partial sums, Probability in the Engineering and Informational Sciences 3 (04) (1989) 511–516.
[21] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd ed., MIT Press, 2001.