Mariam T. Sulaiman
Computer Science Department
University of Technology,
Baghdad, Iraq
mariamtaha201420@yahoo.com

Nidaa F. Hassan
Computer Science Department
University of Technology,
Baghdad, Iraq.
nidaaalalousi_5@yahoo.com

Received on: 13/7/2017
Accepted on: 23/11/2017

Propose an Arabic CAPTCHA System based on Chaotic Maps

Abstract- *CAPTCHA is the facility that prevents web bots from accessing the web services by generating tests to check whether the user is human or computer program. In this paper, a new pseudo-random bits generator based on chaotic system is offered to generate Arabic letters and numbers for CAPTCHA system. The proposed generator uses two Jacobian elliptic Chebyshev rational maps that are combined in the algorithm to produce a block of 32bits in each iteration. A specified number of bits are selected from the resulted blocks to be converted to a set of Arabic letters and numbers. National Institute of Standards and Technology (NIST) statistical test suite are used to assess the generator randomness, all tests has been passed except Longest Run of Ones in a Block Test, Binary Matrix Rank Test and Random Excursions Test.*

Keywords- *CAPTCHA, Chaotic Map, Jacobian elliptic Chebyshev rational map, NIST statistical test suite, pseudo-random bits generator.*

How to cite this article: M.T. Sulaiman and N.F. Hassan, "Propose an Arabic CAPTCHA System based on Chaotic Maps," *Engineering and Technology Journal*, Vol. 36, Part B, No. 1, pp. 48-52, 2018.

1. Introduction

Many people use internet to take advantage of diverse accessible services, e.g., email services, online shopping and blogs. In these services, the user should sign up itself by filling an online sample, but intelligent programs pretend as a human, register for these free services spontaneously. A challenge is the facility to prevent these online services from these automated-bots and it is utilized to distinguish human from bots [1].

"Completely Automated Public Turing Test to Tell Computers and Humans Apart "is a short for the word "CAPTCHA". CAPTCHA is a bot that can be used to distinguish humans from computers by creating tests. These tests must be:

1. passed by most humans.
2. not be passed by computer programs [2].

In this paper, a new pseudo-random bits generator based on chaotic system is suggested to generate Arabic letters and numbers for Arabic CAPTCHA system. This generator combines Two Jacobian elliptic Chebyshev rational maps and a block of 32bit is produced in each iteration. A specified number of bits are selected from the resulted blocks to be converted to a set of Arabic letters and numbers, which are shown as a Captcha, test to be solved. Rigorous statistical analyses are carefully conducted to evaluate the quality, the robustness and the randomness of the generator.

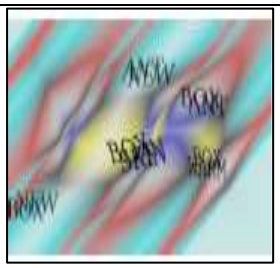
In this paper, sec 2. Types of CAPTCHA is described, chaotic based pseudo-random bit generator is presented in sec. 3. The proposed schema is shown in sec. 4 and sec. 5 the conclusion of the proposal is presented.

2. CAPTCHA Types

CAPTCHAs are classified into different types such as image based, text based and audio based. The following sections present a brief description to each type:

I. Text based CAPTCHAs

This type of CAPTCHA depends on letters and numeric values, to prevent bot attacks; specific distortions and noise are added, however, human eyes have the ability to recognize them [3]. Figure 1 shows examples of text CAPTCHA [1].



a. Gimpy CAPTCHA

b. EZ-Gimpy CAPTCHA

Figure 1: An example of text based CAPTCHAs [1]

II. Image based CAPTCHAs

Image recognition task is needed to be performed by users where to make CAPTCHA more difficult to solve by bots. Users are required to

https://doi.org/10.30684/etj.36.1B.8
2412-0758/University of Technology-Iraq, Baghdad, Iraq

recognize a simple concept among many images presented [3]. Figure 2 shows examples of image based CAPTCHA [4].

Figure 2: image-based CAPTCHA [4]

III. Audio based CAPTCHAs

In this type, a word or sequence of numbers is picked randomly by the program, this word is transformed to a sound clip and a sound clip is distorted, then it is presented to the users and the users are required to enter its contents. Figure 3 shows a sample of audio CAPTCHA [4].

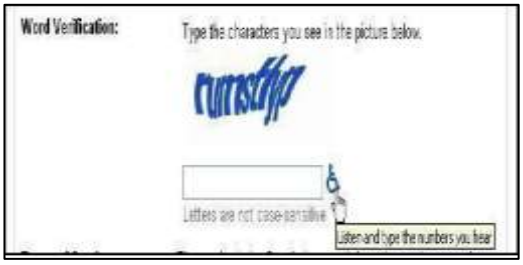


Figure 3: Audio CAPTCHA

IV. Video based CAPTCHAs

In a video CAPTCHA, a test passing requires user to describe a video with three words. If one of the presented words matches the automatically generated ground truth tag set then the test is passed. Figure 4 shows an example of video CAPTCHA [5].

3. Chaotic Based Pseudo-Random Bit Generator

A strong role is played by Chaos theory in the quality enhancement of the PRNGs. In this field, the benefit of using chaos is in its messy behavior and its unpredictability. Chaotic iterations (CIs) are proved as an appropriate tool for fast computing repeatable algorithms, satisfy the feature of the topological chaotic [6]. The characteristics that recognize chaotic systems are high sensitivity to initial conditions, ergodicity and random behaviors [7].

Figure 4: Video based CAPTCHA [5]

I. Chaotic Maps

The performance of specific nonlinear dynamic system is described by chaos theory and under particular conditions display dynamics that are sensitive to initial conditions, so the sensitivity to initial conditions and mixing feature are the two basic features of chaotic systems. The use of chaotic map is to generate the chaotic sequence. Different chaotic maps are used to generate the chaos streams, among the different chaotic maps, four types of maps are listed below:

1. Logistic Map

The logistic map is a simple and well-calculated sample of a 1D map that shows complex behavior from the interval [0,1] to[0,1] , parameterized by the parameter μ:

$$x(n+1) = \mu * \mathrm{x(n)} * \big(1 - \mathrm{x(n)}\big) \quad (1)$$

The equation (1) describes the state evolution where $0 \leq \mu \leq 4$

Where:

μ : a positive real constant.

x: the initial condition[8].

3.1.1. Tent Map

The tent map can be defined by the following equation:

$$T_r(x) = \begin{cases} 2rx, & if\ 0 \leq x \leq \frac{1}{2} \\ 2r(1-x), & if\ \frac{1}{2} < x \leq 1 \end{cases} \quad (2)$$

2. 2D Arnold's Cat Map

The purpose of this type of map is to permute the location of bits and is given by:

$$x_1 = (x_0 + y_0 a) mod\ n \quad (3)$$

$$y_1 = (x_0 b + y_0(ab+1)) mod\ n \quad (4)$$

Where:

n: the dimension of the arnolds cat map .

x_0 , y_0 : the original locations of the bit.

x_1 , y_1 : the locations of bit after the transformations of 2D arnolds cat map .

a , b : the control parameters [10].

3. Jacobian Elliptic Chebyshev Rational Map

Chaotic characteristics like pseudo-random, sensitivity to a small alteration in the initial

conditions, ergodicity are satisfied by Jacobian elliptic chaotic map.

The following equation (5) defines Jacobian elliptic map recursively:

$$R_{n+1}(v,w) = \frac{2vR_n(v,w)}{1-w^2(1-v^2)\left(1-R_n^2(v,w)\right)} - R_{n-1}(v,w) \quad (5)$$

Where:

v: the variable belongs to the periodl [-1,1].

n: integer number let n ≥ 1 .

w: modulus wϵ[0,1].

$R_0(v,w) = 1$, $R_1(v,w)$=v[11].

4. The Proposed Schema

In this paper, a new pseudo-random bits generator based on two Jacobian elliptic Chebyshev rational maps is proposed; these maps are used to generate Arabic letters and numbers randomly for Captcha test. The generator inputs are one initial condition that differs for each map and the same control parameter for both maps. These inputs are applied to each map by using the iterative Equation 5. Two times and the result from each equation is a real number, which is converted to a binary number. The last 32 bits from each sequence are XORed together and at the end, a block of 32 bits is produced. A specified number of bits are selected from the resulted blocks to be converted in the form of Arabic letters and numbers. The following algorithm illustrates the steps of generating Arabic letters and numbers for CAPTCHA system using pseudo-random bits generator based on chaotic systems with two Jacobian elliptic chaotic maps:

Algorithm (1): generate random Arabic letters and numbers using pseudo-random bits generator based on two jacobian elliptic chaotic maps

Input: v1: initial condition for the first jacobian elliptic map where v1ϵ[-1,1].
v2: initial condition for the second jacobian elliptic map where v2ϵ[-1,1].
w: control parameter where wϵ[0,1].
count: the counter for the number of iteration.
Num_of_Iterations: the number of the iterations // used to generate block of 32bits in each one.

Output: Pseudorandom combination of Arabic letters and numbers.

This pseudo-random Arabic letters and numbers generator is based on Jacobian elliptic Chebyshev Rational Map has been analyzed by NIST statistical test suite, which involves 15 tests, and the Table 1 shows the outcomes.

As illustrated in the above table, the pseudo-random bits generator based on two Jacobian elliptic Chebyshev rational maps, this generator is passed all NIST tests except Longest Run of Ones in a Block Test, Binary Matrix Rank Test and Random Excursions Test due to inputs of the chaotic maps. In this paper, a Jacobian elliptic Chebyshev rational map is chosen because its output has high randomness compared to other types of chaotic maps.

The advantages of using pseudo-random bits generators based on chaotic maps (CPRBG) over the traditional pseudo-random bits generator (PRBG) are: high sensitivity to the initial condition, since any small change to the initial condition leads to complete different results.

- high randomness compared to other normal PRBGs.
- Ergodicity according to ergodic theory, which studies the dynamical systems.
- Unpredictability.
- Disordered behavior.

Step 1: Iterate the two Jacobian elliptic chaotic maps 800 times to remove the transient effect of the chaotic maps by using the equation (5) for each v1 and v2.

Step 2: Convert the values taken from the two Jacobian elliptic maps after the iteration of each jacobian elliptic map 800 times to the binary values.

Step 3: Perform an XOR operation between the last 32bits of the two binary sequence taken from step 2 and the result is a block of 32 bits.

Step 4: Set count=1.
While(count ≤ Num_Of_Iterations) do
{
- Apply the two Jacobian elliptic maps by using Equation 5. with the values resulted from the step 2.
- Convert the values taken from the two Jacobian elliptic maps to the binary values.
- perform an XOR operation between the last 32bits of the two binary sequence and the result is a block of 32 bits.
- count=count+1.

}

Step 5: Take a specific required number of bits and convert them to the Arabic letters and numbers.

Step 6: end.

Table 1: analyzing randomness by NIST statistical test package for pseudo-random Arabic letters and numbers generator based on two Jacobian elliptic Chebyshev rational maps.

Test Name	*P-value*	*Result*
The Frequency Test	0.62687081853618987	*Pass*
The Frequency within Block Test	0.76943442505154147	*Pass*
The Run Test	0.999999999	*Pass*
The Longest Run of Ones in a Block Test	0.0029733625605687926	*Fail*
The Binary Matrix Rank Test	0.0055439147831519275	*Fail*
The Discrete Fourier Transform Test	0.77167063761183041	*Pass*
The Non-overlapping Template Matching Test	0.999999999999991	*Pass*
The Overlapping Template Matching Test	0.9999714733895565	*Pass*
The Maurer's "Universal Statistical" Test	0.93573207768305477	*Pass*
The Linear Complexity Test	0.99852934605644	*Pass*
The Serial Test	0.3880392484878506	*Pass*
	0.31725194993957595	*Pass*
The Approximate Entropy Test	0.42742613717432776	*Pass*
The Cumulative Sums Test	0.72221177167402917	*Pass*
The Random Excursions Test	0.94736135897740947	*Pass*
	0.65354946503814393	*Pass*
	0.00000851404472251304	*Fail*
	0.90421124960566768	*Pass*
	0.89590874200435588	*Pass*
	0.0000000334839094	*Fail*
	0.77492362051054287	*Pass*
	0.27245262856567726	*Pass*
The Random Excursions Variant Test	0.94950354826301442	*Pass*
	0.92837186798902149	*Pass*
	0.846863603619318	*Pass*
	0.8956043955180788	*Pass*
	0.7940027991134575	*Pass*
	0.9475407628030168	*Pass*
	0.69709171705917938	*Pass*
	0.80161363203679714	*Pass*
	0.7277236777459174	*Pass*
	0.38408816978754312	*Pass*
	0.26892507216248152	*Pass*
	0.55930486312766026	*Pass*
	0.71744673911865375	*Pass*
	0.7940027991134575	*Pass*
	0.85424698257868492	*Pass*
	0.88483607852776436	*Pass*
	0.78740661527770728	*Pass*
	0.45999759729552947	*Pass*

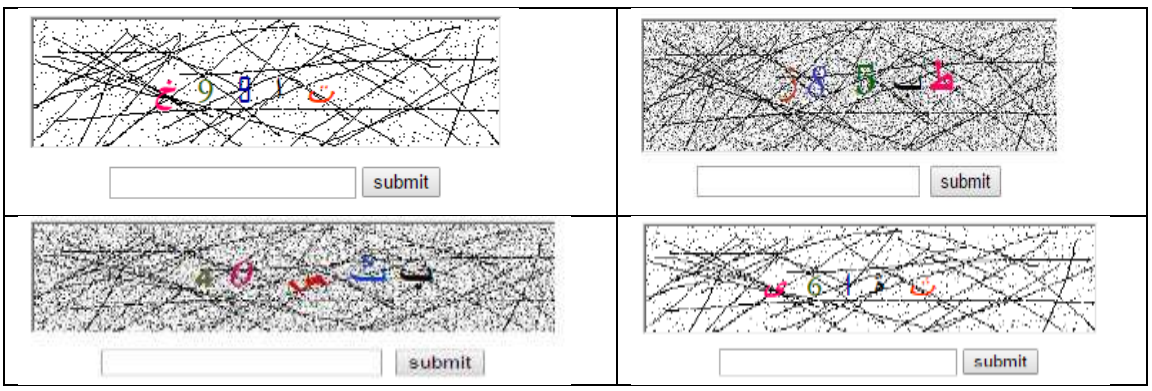


Figure 5: Arabic letters and numbers based CAPTCHA

Generally, the way used in producing any symbol (Arabic letters, English letters, numbers and others) from any pseudo-random bits generator is to choose a finite number of bits by determining the number of symbols to be displayed for CAPTCHA test and the maximum number of bits to represent each symbol.
Determining the maximum number of bits is done by observing the largest ASCII-code number among the ASCII-code numbers of the existing symbols that are the candidates to appear in the CAPTCHA test.
Figure 5 shows some samples of Arabic letters and numbers based CAPTCH**5.**

Conclusion

CAPTCHA is a way to prohibit malicious programs from accessing available services on web sites by creating tests to validate whether the user is human or bot. In this paper, a new pseudo random bit generator based on chaotic system has been proposed to generate random Arabic letters and numbers , within this new generator, two jacobian elliptic Chebyshev rational maps have been used with different initial condition for each map, the same control parameter is used for both maps for each iteration, this generator is produced a block of 32 bits and has passed all NIST tests except Longest Run of Ones in a Block Test, Binary Matrix Rank Test and Random Excursions Test due to the inputs of the chaotic maps. Pseudo-random bits generators based on chaotic maps (CPRBG) has features like sensitivity to initial conditions, unpredictability and ergodicity. Number of bits is selected from the resulted blocks to be converted to Arabic letters and numbers that are showed as a captcha test.

References

[1] B.Khan, K.Alghathbar, M.K.Khan, A.Alkelabi and A.Alajaji "Cyber Security Using Arabic CAPTCHA Scheme," The International Arab Journal of Information Technology, Vol. 10, No. 1, 76-84, January 2013.

[2] L.V. Ahn, M. Blum, N.J. Hopper, and J. Langford "CAPTCHA: Using Hard AI Problems for Security," International Conference on the Theory and Applications of Cryptographic Techniques, Vol. 2656, pp. 294-311, 2003.

[3] K.R. Soumya, R.M. Abraham and K.V. Swathi "A Survey on Different CAPTCHA Techniques," International Journal of Advances in Computer Science and Technology, Vol. 3, No.2, 117-122, 2014.

[4] S.S. Pawar1, P.P. Kalyankar "Understanding Captcha: with its types, for security," International Journal of Engineering and Computer Science, Vol. 5, Issue 8, 17359-17363, 2016.

[5] K.A. Kluever and R. Zanibbi "Video CAPTCHAs: Usability vs. Security," Rochester Institute of Technology, Rochester, NY USA, pp.1-4, September 26, 2008.

[6] C. Guyeux, Q. Wang and J. Bahi "A Pseudo Random Numbers Generator Based on Chaotic Iterations. Application to Watermarking," International Conference on Web Information Systems and Mining, Vol. 6318, pp. 202-211, 2010.

[7] M. Francois, T. Grosges, D. Barchiesi and R. Erra "A New Pseudo-Random Number Generator Based on Two Chaotic Maps," Vol. 24, No. 2, 181–197, 2013.

[8] G.A. Sathishkumar, K. Bhoopathy bagan and N. Sriraam "Image Encryption Based on Diffusion and Multiple Chaotic Maps," International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.2, 181-194, March 2011.

[9] W.F.H. Al-shameri and M.A. Mahiub, "Some Dynamical Properties of the Family of Tent Maps," Int. Journal of Math. Analysis, Vol. 7, No. 29, 1433 – 1449, 2013.

[10] P.N. Khade and M. Narnaware "3D Chaotic Functions for Image Encryption," IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, 323-328, May 2012.

[11] Q. Ke, Z. Mingtian, L. Naiqi, H. Yujie and G. Jiandong "A Novel Group Key Management Based on Jacobian Elliptic Chebyshev Rational Map," IFIP International Conference on Network and Parallel Computing, pp. 287-295, 2007.