Open access • Proceedings Article • DOI:10.1109/ICITST.2016.7856658

Robust chaos-based stream-cipher for secure public communication channels

Ons Jallouli, Safwan El Assad, Maryline Chetto

Institutions: University of Nantes

Published on: 01 Dec 2016 - International Conference for Internet Technology and Secured Transactions

Topics: Stream cipher, Stream cipher attack, Running key cipher, Affine cipher and Transposition cipher

Robust Chaos-based Stream-Cipher for Secure Public Communication Channels

Ons Jallouli, Safwan El Assad, Maryline Chetto

Robust Chaos-based Stream-Cipher for Secure Public Communication Channels

Ons Jallouli
IETR Université de Nantes
Nantes, France
Email: ons.jallouli@univ-nantes.fr

Safwan El Assad
IETR Université de Nantes
Nantes, France
Email: safwan.elassad@univ-nantes.fr

Maryline Chetto
IRCCyN, Université de Nantes
Nantes, France
Email: maryline.chetto@univ-nantes.fr

***Abstract*—In this paper, we propose a novel stream cipher based on a chaotic system. In order to get the cipher text, the plain text is simply XORed with the key-stream generated by a strong pseudo chaotic number generator (PCNG). Then, all the security of the system is based on the used PCNG. The structure of the proposed PCNG includes two chaotic maps which are weakly coupled by a predefined matrix and integrates a swap function. The PCNG passes all statistical NIST's randomness tests. Also, analysis and experimental results show that the proposed stream cipher has a large key space, a high key sensitivity, and can resist against chosen-plaintext attack and chosen-ciphertext attack. Indeed, for each new execution of the system using the same secret key, the generated key-steram is different due to the IV-setup function. The computing performance of the proposed system is comparable to the main algorithms of eStream such as Rabbit and HC-128.**

I. Introduction

Nowadays, due to the widespread transmission over various communication networks, data security is gaining more importance and has been a subject of intense research. Obviously, designing efficient image encryption schemes has become a focal research topic. During the last years, and due to the interesting features of chaos in cryptography, a large number of chaos-based image encryption schemes has been proposed. Some of them have proved to be vulnerable to certain types of attacks [1][2][3]. Recently, some published works on chaos-based block ciphers are very efficient in terms of security and computing time [4][5][6][7]. Also, some interesting works on lightweight stream ciphers are recently published [8][9] [10] [11]. Stream ciphers are potential candidates for data encryption, along with block ciphers. Stream ciphers are a class of symmetric cryptography. They are inherently suitable for time-critical applications, with little computational resources, where the speed of encryption and decryption is a concern. The main characteristic of a stream cipher is its speed on different platforms, and also chip area, power consumption for hardware implementations. A stream cipher is a symmetric cryptosystem. It means that, in order to encrypt / decrypt the message, emitter and receiver must share the same key. A common way to build a stream cipher is to use a pseudo-random number generator (or keystream generator) and xoring this keystream with the plain-text / cipher text. Several stream ciphers have been proposed in the literature. e.g., A5/1 [12], RC4 [13], SEAL [14], SNOW [15]. Most of the proposed stream ciphers have proved to be very weak and insecure [16].

The eStream project was held in February 2004 [17] by the European Network of Excellence for Cryptology (ECRYPT), in order to encourage cooperation between researchers in security. Its main goal is to give rise to a standarisation of secure and efficient stream ciphers. 34 stream cipher candidates were submitted as a response to the call by eStream and only few proposals are chosen at the end of the project: HC-128, Rabbit, Salsa20/12 and SOSEMANUK as software-oriented ciphers, and Grain v1, MICKEY 2.0 and Trivium in the hardware category. It had been noted that the finalist stream ciphers were proven to be secure against attacks. However, new cryptanalysis works mention some security flaws for some of these ciphers [8].

In this paper, a robust chaos-based stream cipher is proposed. It based on a strong Pseudo Chaotic Number Generator (PCNG) to generate the key stream. To provide high robustness, the PCNG integrates two chaotic maps : Piece-Wise Linear Chaotic Map (PWLCM) and Skewtent map. Its structure includes a coupling and swap functions. The proposed structure passes all NIST's randomness tests justifying the good statistical properties of the key stream. In addition, security analysis and experimental results show that our stream cipher has a high security level, resistant to various typical attacks. Also, the proposed chaos-based stream cipher has good computing performance.

The paper is organized as follows: In section II, we describe the structure of the proposed PCNG and we give the NIST 's result. The performance of the proposed chaos-based stream cipher in terms of security analysis and time consuming are given in section III. Finally, in section IV, we conclude the paper.

II. Proposed pseudo chaotic number generator

In this section, first, we describe in details the structure of the proposed PCNG used in our stream cipher. The PCNG is the central element of any stream cipher. Second, we present the experimental results of the NIST statistical test that were carried out on the key-strem generated by the proposed PCNG in order to quantify its statistical cryptographic properties.

A. Architecture of the pseudo chaotic generator

The proposed structure is presented in Fig.(1). It consists of three main function blocks: Key-setup, Internal State and Output function. It takes a secret key "K" and 64 bits initial vector "IV" as input, and as output, it generates a pseudo-chaotic samples, each quantified on N=32 bits.

The structure uses two chaotic maps: PWLCM and Skew Tent maps, and includes a coupling and swap chaotic techniques.

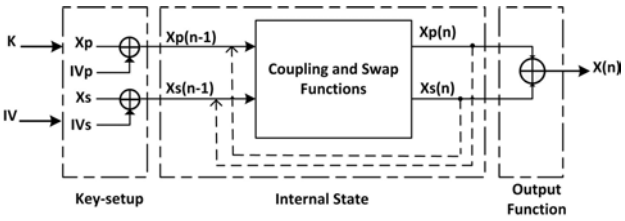

Fig. 1: Structure of the proposed PCNG.

Key-setup: Its role is to calculate the initial values $Xp(0)$ and $Xs(0)$ of the chaotic maps Pwlcm and Skewtent respectively from Xp and Xs of the secret key K and from the initial vector IV. The initial values $Xp(0)$ and $Xs(0)$ are calculated as follows:

$$\begin{cases} Xp(0) = Xp \oplus IVp \\ Xs(0) = Xs \oplus IVs \end{cases} \quad (1)$$

Where $IVp = lsb(IV)$ and $IVs = L_{cir}[lsb(IV), 3]$.

$\oplus$ denotes the XOR operator, $lsb(IV)$ is the 32 least significant bits of IV and $L_{cir}[S, q]$ performs the q-bits left circular shift on the binary sequence S.

Notice that, the secret key K of the system is formed by:

- the initial conditions Xp and Xs of the chaotic maps Pwlcm and Skewtent, ranging from 1 to 2^N-1,
- the control parameters Pp and Ps of Pwlcm and Skew-tent maps, in the range $[1, 2^{N-1} - 1]$ and $[1, 2^N - 1]$ respectively,
- the parameters of the coupling technique, ε_{ij}, ranging from 1 to 2^k with k ≤ 5 and $i, j \in \{1, 2\}$.

These initial conditions, parameters and initial vector are chosen randomly from Lunix generator: `"/dev/urandom"`.

Internal state: The samples $Xp(n)$ and $Xs(n)$ are produced by using a coupling and swap chaotic techniques. The equation of the system is governed by:

$$\begin{bmatrix} Xp(n) \\ Xs(n) \end{bmatrix} = \begin{bmatrix} (2^N - \varepsilon_{11}) & \varepsilon_{12} \\ \varepsilon_{21} & (2^N - \varepsilon_{22}) \end{bmatrix} \times \begin{bmatrix} Fp[Xs(n-1)] \\ Fs[Xp(n-1)] \end{bmatrix}. \quad (2)$$

Where $Fp[Xp(n-1)]$ and $Fs[Xs(n-1)]$ are the discrete functions of the chaotic maps Pwlcm and Skew Tent maps respectively [18].

Output function: The output samples $X(n)$ are calculated throughout the produced samples $Xp(n)$ and $Xs(n)$ as follows:

$$X(n) = Xp(n) \oplus Xs(n). \quad (3)$$

B. NIST statistical test

We highlight the robustness of the proposed chaotic generator through the NIST (National Institute of Standards and Technology) statistical test that is carried out in order to quantify the good randomness of the produced sequences (key-streams). The NIST designed a battery of 15 statistical tests based on mathematical theory to test randomness of binary sequences produced by pseudo-random number generators [19], [20]. We generated 100 different binary sequences each one with a different secret key and containing 31250 samples (size of each sequence is n=10^6 bits), with a level of significance $\alpha = 0.01$). We present the obtained results of NIST for a generated sequence X in Table I. As we can see, the sequence X has successfully passed all NIST tests, since the computed $P - value$ of each test is > 0.01. Therefore we conclude that the output sequence of our PRNG is enough random, and it resists against statistical attacks.

TABLE I: P-values and Proportion results of NIST.

Test	P-value	Proportion
Frequency test	0.249	98
Block-frequency test	0.063	99
Cumulative-sums test	0.862	98
Runs test	0.456	100
Longest-run test	0.720	100
Rank test	0.924	100
FFT test	0.596	98
Nonperiodic-templates	0.540	98.784
Overlapping-templates	0.817	98
Universal	0.720	98
Approximty entropie	0.972	98
Random-excursions	0.325	98.674
Random-excursions-variant	0.273	98.401
Serial test	0.720	99.5
Linear-complexity	0.475	100

III. Security analysis and simulation results of the proposed stream cipher

In this section, we study the performance in terms of security and computing time of our proposed chaos based stream-cipher. We use as plaintext the color image of "Lena" image ($256 \times 256 \times 3$) shown in Fig.(2a). Fig.(2c) shows the encrypted image of Lena. It can be seen from this result that visually there are no relationship between the original image and its encrypted one.

A. Key space, chosen plaintext and chosen ciphertext attacks

A good stream-cipher algorithm should resist all kinds of known attacks such as exhaustive attack, chosen plaintext and chosen ciphertext attacks. Basically, it should be sensitive to the secret key, and the key space should be large enough to resist brute-force attacks. For the proposed crypto-system, the size of the secret key is given by:

$$|K| = (|Xp| + |Xs|) + (|Pp| + |Ps|) + 4 \times |\varepsilon_{ij}| = 147 \; bits \tag{4}$$

where $|Xp| = |Xs| = |Ps| = 32$ bits ; $|Pp| = 31$ bits and $|\varepsilon_{ij}|$ is equal to 5 bits. Therefore the secret key size is large enough to prevent the exhaustive searching. Thus, the brute-force attack on the key is computationally unfeasible. Also, the proposed system can resist against chosen plaintext and chosen ciphertext attacks. Indeed, due to the IV-Setup function, encrypting the same plaintext several times using a same secret key, gives ciphertexts totally different from each other.

B. Key sensitivity

An efficient crypto-system must be very sensitive to the secret key; a small change in the secret key causes a greatly significant change in the output. We evaluate the key sensitivity by using the Hamming Distance $D_{Hamming}$ test.

We calculate the average Hamming Distance $D_{Hamming}$ (using 100 secret keys), between two sequences X and Y, ciphered with only one change in the least significant bit of the parameter Pp. The Hamming distance $D_{Hamming}(X, Y)$ is given by the following equation :

$$D_{Hamming}(X, Y) = \frac{1}{Nb} \times \sum_{K=1}^{Nb} (X[K] \oplus Y[K]) \tag{5}$$

With Nb is the number of bits in a sequence. The obtained result of Hamming distance is equal to 0.499995 (close to the optimal value of 50%). This means that the secret key is very sensitive.

Two others quantitative measures are usually used, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI), in order to test the key sensitivity attacks. NPCR is designed to evaluate the changed pixel numbers in cipher image, while only one bit changes in the key. The optimal value of NPCR is almost 99.61 %. UACI refers to the average values of difference in intensity between two cipher images. The optimal value of UACI is almost 33.46 %. The parameters NPCR and UACI are defined as bellow :

$$NPCR = \frac{1}{L \times C \times P} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} D[i, j, p] \times 100\% \tag{6}$$

$$D[i, j, p] = \begin{cases} 0, & \text{if } C_1[i, j, p] = C_2[i, j, p] \\ 1, & \text{if } C_1[i, j, p] \neq C_2[i, j, p] \end{cases} \tag{7}$$

$$UACI = \frac{1}{L \times C \times P \times 255} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} |C_1 - C_2| \times 100\% \tag{8}$$

We have found that the NPCR is 99.608 % and the UACI is 33.47 %, showing thereby that the encryption scheme is very sensitive with respect to little change in the secret Key.

C. Histogram and Chi-square test

The histogram of the encrypted image should be uniformly distributed. We give in Fig.(2) **(a)** Lena image, **(b)** histogram of Lena Image, **(c)** Encrypted Lena image, and **(d)** histogram of encrypted Lena Image. It can be observed that the histogram of the encrypted image is very close to a uniform distribution.

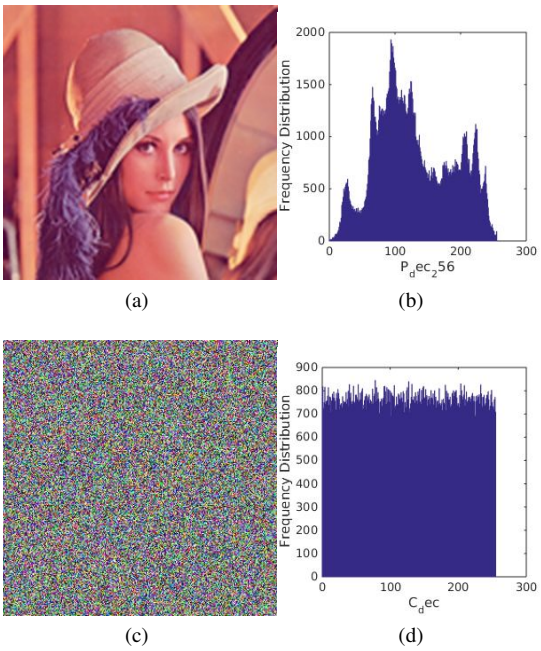

Fig. 2: Lena image, encrypted Lena and their histograms.**(a)** Lena image, **(b)** histogram of Lena Image, **(c)** Encrypted Lena image, **(d)** histogram of encrypted Lena Image.

We ensure this uniformity by applying the chi-square test given by:

$$\chi^2_{\text{exp}} = \sum_{i=0}^{K-1} \frac{(O_i - E_i)^2}{E_i}. \tag{9}$$

Where K is the number of levels (here 256), O_i are the observed occurrence frequencies of each color level (0-255) in the histogram of the ciphered image, and E_i is the expected occurrence frequency of the uniform distribution, given here by $E_i = (L \times C \times P)/256$ [6]. We give in Table II, the results of Chi-square test of three ciphered images (Baboon, Peppers, and Lena) having the size of (256×256×3), with a significant level of 0.05. We observe that for the three images, $\chi^2_{\text{exp}} < \chi^2_{th}(255, 0.05)$. Therefore, the distribution of the histograms is uniform.

TABLE II: Theoretical and experimental values for the Chi-Square test.

	Baboon	Peppers	Lena
χ^2_{th}	293	293	293
χ^2_{exp}	253.5	259.1	251.2

D. Computing performance

We have analyzed the speed of the proposed algorithm on an Intel Core i5 @ 2.60 GHz with 15.6 GB Running on Ubuntu 14.04 Trusty Linux, using a C compiler. In this implementation, we do not parallelize processes and operations. We applied the proposed algorithm to the Lena image of size (256×256×3). We evaluated the computing performance as follows: for 100 different keys, we executed our algorithm and then, we calculated the average encryption time in (Micro second), the average encryption throughput (ET) in (MByte) and the number of cycles per byte (NCpB).

$$ET = \frac{Image\ size(MByte)}{Encryption\ time(s)} \tag{10}$$

$$NCpB = \frac{CPUspeed(Hz)}{ET(Byte/s)} \tag{11}$$

Table III presents the speed performance of the proposed algorithm.

TABLE III: Computing performance of the proposed algorithm.

	Proposed algorithm
Average encryption time (μs)	1059.5
ET (Mbits/s)	1415.76
NCpB	14

In Table IV we compare the speed performance in term of needed cycles to encrypt one byte (NCpB) of our proposed algorithm with two algorithms of the eStream project, namely Rabbit and HC-128 stream ciphers [17].

TABLE IV: Performance speed comparison.

	Proposed algorithm	Rabbit	HC-128
NCpB	14	10	14.6

We note that the proposed stream cipher has good computing performance while having a very high level of security.

IV. Conclusion

In this paper, we proposed a novel stream cipher based on a strong pseudo chaotic number generator (PCNG), which contains two chaotic maps weakly coupled by a predefined matrix and integrates a swap function. Through the security analysis and the obtained experimental results, we find that our algorithm is very secure and its computing performance is comparable with those of project eStream. All these features prove that our chaos-based stream cipher is very suitable for encrypting data before their transmission over public transmission channels.

Acknowledgment

The authors would like to thank the European program: Erasmus Mundus scholarship E-GOV-TN.

References

[1] K.-W. Wong, B. S.-H. Kwok, and W.-S. Law, "A fast image encryption scheme based on chaotic standard map," *Physics Letters A*, vol. 372, no. 15, pp. 2645–2652, 2008.

[2] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, no. 9, pp. 926–934, 2006.

[3] Y. Tang, Z. Wang, and J.-a. Fang, "Image encryption using chaotic coupled map lattices with time-varying delays," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2456–2468, 2010.

[4] W. Zhang, K.-w. Wong, H. Yu, and Z.-l. Zhu, "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 8, pp. 2066–2080, 2013.

[5] X. Zhang, Z. Zhao, and J. Wang, "Chaotic image encryption based on circular substitution box and key stream buffer," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 902–913, 2014.

[6] S. El Assad and M. Farajallah, "A new chaos-based image encryption system," *Signal Processing: Image Communication*, 2016.

[7] M. Farajallah, S. El Assad, and O. Deforges, "Fast and secure chaos-based cryptosystem for images," *International Journal of Bifurcation and Chaos*, pp. ID–paper, 2016.

[8] C. Manifavas, G. Hatzivasilis, K. Fysarakis, and Y. Papaefstathiou, "A survey of lightweight stream ciphers for embedded systems," *Security and Communication Networks*, 2015.

[9] G. Vidal, M. S. Baptista, and H. Mancini, "A fast and light stream cipher for smartphones," *The European Physical Journal Special Topics*, vol. 223, no. 8, pp. 1601–1610, 2014.

[10] R. Lozi and E. Cherrier, "Noise-resisting ciphering based on a chaotic multi-stream pseudo-random number generator," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*. IEEE, 2011, pp. 91–96.

[11] O. Garasym and I. Taralova, "High-speed encryption method based on switched chaotic model with changeable parameters," in *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*. IEEE, 2013, pp. 37–42.

[12] E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of gsm encrypted communication," in *Advances in Cryptology-CRYPTO 2003*. Springer, 2003, pp. 600–616.

[13] A. Klein, "Attacks on the rc4 stream cipher," *Designs, Codes and Cryptography*, vol. 48, no. 3, pp. 269–286, 2008.

[14] P. Rogaway and D. Coppersmith, "A software-optimized encryption algorithm," *Journal of Cryptology*, vol. 11, no. 4, pp. 273–287, 1998.

[15] P. Ekdahl and T. Johansson, "Snow-a new stream cipher," in *Proceedings of First Open NESSIE Workshop, KU-Leuven*, 2000, pp. 167–168.

[16] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Cryptanalytic attacks on pseudorandom number generators," in *Fast Software Encryption*. Springer, 1998, pp. 168–188.

[17] M. Robshaw, "The estream project," in *New Stream Cipher Designs*. Springer, 2008, pp. 1–6.

[18] O. Jallouli, S. El Assad, M. Chetto, R. Lozi, and D. Caragata, "A novel chaotic generator based on weakly-coupled discrete skewtent maps," in *International Conference on Internet Technology and Secured Transactions*, 2015, pp. 38–43.

[19] B. Elaine and K. John, "Recommendation for random number generation using deterministic random bit generators," NIST SP 800-90 Rev A, Tech. Rep., 2012.

[20] A. L. Rukhin, J. Soto, J. R. Nechvatal, M. Smid, E. B. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST SP 800-22 Rev 1, Tech. Rep., 2008.