# Comparative Study of two Pseudo Chaotic Number Generators for Securing the IoT

Ons Jallouli, Mohammad Abu Taha, Safwan El Assad, Maryline Chetto, Audrey Queudet, Olivier Déforges

▶ **To cite this version:**

Ons Jallouli, Mohammad Abu Taha, Safwan El Assad, Maryline Chetto, Audrey Queudet, et al.. Comparative Study of two Pseudo Chaotic Number Generators for Securing the IoT. International Conference on Advances in Computing, Communications and Informatics (ICACCI-2016), Sep 2016, Jaipur, India. pp.1345-1349, 10.1109/ICACCI.2016.7732234 . hal-01355173

**HAL Id: hal-01355173**
**https://hal.science/hal-01355173**

Submitted on 11 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

# Comparative Study of two Pseudo Chaotic Number Generators for Securing the IoT

Ons Jallouli*, Mohammed Abutaha*, Safwan El Assad*, Maryline Chetto†, Audrey Queudet†, Olivier Deforges‡
*Institut d'Electronique et de Télécommunications de Rennes, Université de Nantes, France,
Email: ons.jallouli@univ-nantes.fr, mohammad.abu-taha@univ-nantes.fr, safwan.elassad@univ-nantes.fr
†Institut de Recherche en Communications et cybernétique de Nantes, Université de Nantes, France,
Email: maryline.chetto@univ-nantes.fr, audrey.queudet@univ-nantes.fr
‡Institut National des Sciences Appliquées de Rennes, Rennes, France,
Email: olivier.deforges@insa-rennes.fr

***Abstract*—The extremely rapid development of the Internet of Things brings growing attention to the information security issue. Realization of cryptographically strong pseudo random number generators (PRNGs), is crucial in securing sensitive data. They play an important role in cryptography and in network security applications. In this paper, we realize a comparative study of two pseudo chaotic number generators (PCNGs). The First pseudo chaotic number generator (PCNG1) is based on two nonlinear recursive filters of order one using a Skew Tent map (STmap) and a Piece-Wise Linear Chaotic map (PWLCmap) as non linear functions. The second pseudo chaotic number generator (PCNG2) consists of four coupled chaotic maps, namely: PWLCmaps, STmap, Logistic map by means a binary diffusion matrix [D]. A comparative analysis of the performance in terms of computation time (Generation time, Bit rate and Number of needed cycles to generate one byte) and security of the two PCNGs is carried out.**

## I. Introduction

The conversion from closed enterprise IT networks to public networks (Internet) is increasing continuously and justly raising questions about security. With the huge and rapid revolution of the Internet, we are increasingly interconnected on intelligent devices, and a lot of various digital data, such as text, image, video, or audio, travel from one destination to another via the network channel. Some of these data might be sensitive and confidential, therefore they need to be secured. The need to develop methods to secure these transactions has become a challenge for many researchers. Randomness in data processing is a key factor for security. Then, the generation of pseudo-random numbers is a very important topic which continue to encourage the researchers since many decades. Indeed, a pseudo-random number generator (PRNG) is defined as an algorithm enabling to generate a sequence of numbers with high properties of randomness. The design of a PRNG depends on applications to which it is dedicated. For data protection, Pseudo Chaotic Number Generators (PCNGs) are the central element of any strongly secure chaos-based block and stream ciphers [1] [2] [3] [4] [5]. For that, many PCNGs have been proposed in the literature for stream ciphers [6] [7] [8].

In this paper, we realized and presented a comparative study of two PCNGs for stream ciphers. The first PCNG is based on non-linear recursive filter. The second PCNG uses a coupling technique based on a binary diffusion matrix. Both PCNGs integrates a chaotic switching technique.

The paper is organized as follows: In section II, we describe the structures of the two proposed PCNGs. The performance of both PCNGs in terms of time consuming and security analysis is given in section III. Finally, in section IV, we conclude the paper.

## II. Proposed pseudo chaotic number generators: PCNGs

In this section, we describe in details the structure of the two proposed PCNGs. The first PCNG uses two chaotic maps with a delayed feedback loop and a chaotic multiplexing technique. The second PCNG is based on four different chaotic maps coupled by a binary diffusion matrix and uses a chaotic switching technique.

All the initial conditions, parameters and initial vector for the two PCNGs are chosen randomly from Lunix generator `"/dev/urandom"`.

### A. Architecture of the pseudo chaotic number generators

The architecture of the PCNG1 is presented in Figure 1. It consists of two nonlinear recursive filters of order one. The first recursive filter contains a discrete Skew Tent map (STmap) and the second recursive filter contains a discrete Piecewise Linear Chaotic map (PWLCmap). The outputs of these recursive filters are given respectively by:

$$X\_s = STmap\{F1[n-1], P1\} \oplus Q1 \quad (1)$$

with

$$F1[n-1] = mod[U\_s + X\_s(0) + (K1\_s \times X1\_s), 2^N] \quad (2)$$

And the equation of the recursive cell containing the PWLC map is defined by:

$$X\_p = PWLCmap\{F2[n-1], P2\} \oplus Q2 \quad (3)$$

output function produces the output sequence $X(n)$, by using a chaotic switching technique.

The equation of the system is given by:

$$\begin{bmatrix} Xp1(n) \\ Xs(n) \\ Xp2(n) \\ Xl(n) \end{bmatrix} = \mathbf{D} \odot \begin{bmatrix} Fp[Xp1(n-1)] \\ Fs[Xs(n-1)] \\ Fp[Xp2(n-1)] \\ Fl[Xl(n-1)] \end{bmatrix}. \quad (7)$$

And $\odot$ is the operator defined as we can see in the following equation :

$$\begin{bmatrix} Xp1(n) \\ Xs(n) \\ Xp2(n) \\ Xl(n) \end{bmatrix} = \begin{bmatrix} Fp[Xp1(n-1)] \oplus Fs[Xs(n-1)] \oplus Fp[Xp2(n-1)] \\ Fp[Xp1(n-1)] \oplus Fs[Xs(n-1)] \oplus Fl[Xl(n-1)] \\ Fp[Xp1(n-1)] \oplus Fp[Xp2(n-1)] \oplus Fl[Xl(n-1)] \\ Fs[Xs(n-1)] \oplus Fp[Xp2(n-1)] \oplus Fl[Xl(n-1)] \end{bmatrix}. \quad (9)$$

The obtained samples of the sequence $X(n)$ are controlled by the chaotic switching technique, using the obtained sample $Xl(n)$ and two threshold $Th1$ and $Th2$, as defined as follows:

$$X(n) = \begin{cases} Xp1(n), & \text{if } 0 < Xl(n) < Th1 \\ Xs(n), & \text{if } Th1 \leq Xl(n) < Th2 \\ Xp2(n), & \text{otherwise} \end{cases} \quad (10)$$

Where $Th1 = 0.8 \times 2^N$ and $Th2 = 0.9 \times 2^N$.

## III. Experimental Results

In this section, we give the obtained performance of the two PCNGs in terms of computing time and robustness against known and statistical attacks.

### A. Computing Performance of PCNGs

The experiment is made using a two 32-bit multicore Intel Core(TM) i5 processors running at 2.60 GHz with 16 G of main memory. This hardware platform was used on top of an Ubuntu 14.04 Trusty Linux distribution, and the programming is done in code C. We provide below, for different sizes of data bytes, the average generation time in micro second GT($\mu$s), the average bit rate en Mega bit par second BR(Mbit/s), and the average of the needed number of cycles to generate one byte, NCpB($Cycles/B$). The average is calculated by using 100 different secret keys. The results obtained in tables I, II, III, show that the PCNG2 has better computing performance than PCNG1. Besides, these computing performance are better than some known pseudo-random number generators of the literature: Jallouli et al. [7], François et al. [9], QUANTIS [10] and Blum Blum Shub [11].

where $D$ is the binary diffusion matrix:

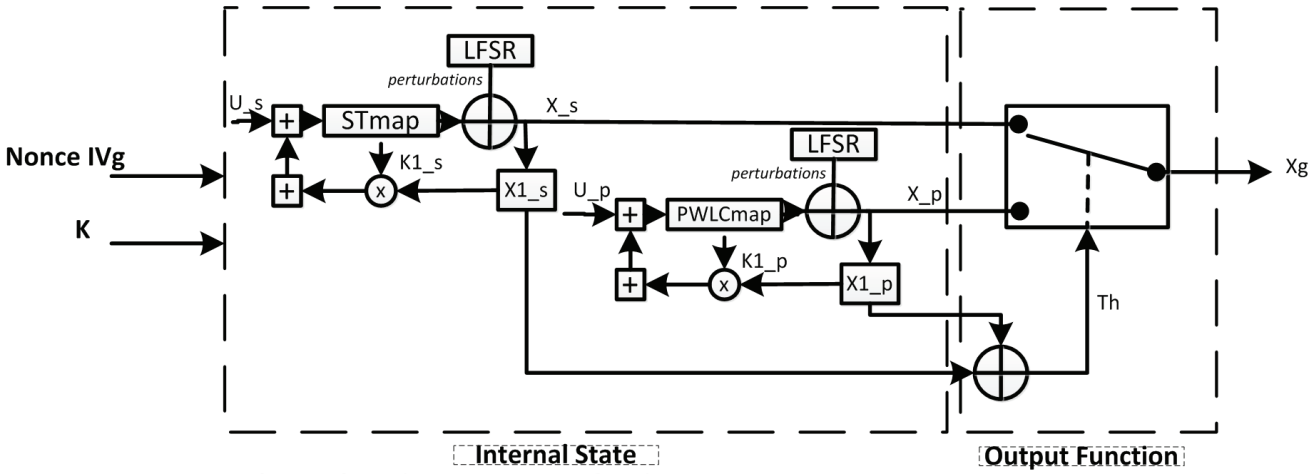
$$D = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \quad (8)$$

TABLE I: Generation Time of PCNG1 and PCNG2

| **Data Bytes** | **PCNG1 GT($\mu$s)** | **PCNG2 GT($\mu$s)** |
|---|---|---|
| 64 | 6 | 2 |
| 128 | 8 | 4 |
| 256 | 11 | 8 |
| 512 | 19 | 13 |
| 1024 | 32 | 23 |
| 2048 | 57 | 46 |
| 4096 | 109 | 52 |
| 16384 | 332 | 196 |
| 32768 | 520 | 338 |
| 65536 | 712 | 654 |
| 125000 | 1282 | 1179 |
| 196608 | 1830 | 1600 |
| 393216 | 2902 | 2801 |
| 786432 | 5502 | 4237 |
| 3145728 | 21723 | 16727 |
| 12582912 | 85009 | 66666 |

TABLE II: Bit Rate of PCNG1 and PCNG2

| **Data Bytes** | **PCNG1 BR($Mbit/s$)** | **PCNG2 BR($Mbit/s$)** |
|---|---|---|
| 64 | 85.33 | 171.81 |
| 128 | 128 | 212.44 |
| 256 | 186.18 | 255.36 |
| 512 | 215.58 | 306.12 |
| 1024 | 256 | 348.44 |
| 2048 | 287.44 | 349.11 |
| 4096 | 300.62 | 621.07 |
| 16384 | 394.8 | 666.35 |
| 32768 | 504.12 | 774.13 |
| 65536 | 736.36 | 801.61 |
| 125000 | 780.03 | 804.68 |
| 196608 | 859.49 | 982.54 |
| 393216 | 1083.99 | 1122.67 |
| 786432 | 1143.49 | 1484.75 |
| 3145728 | 1158.49 | 1504.48 |
| 12582912 | 1184.15 | 1509.95 |




IVg: {Us,Up } :Intilization Vector of 64-bit
K: Secret Key
Xg: sample of 32_bit

Fig. 1: Structure of the first proposed PCNG1

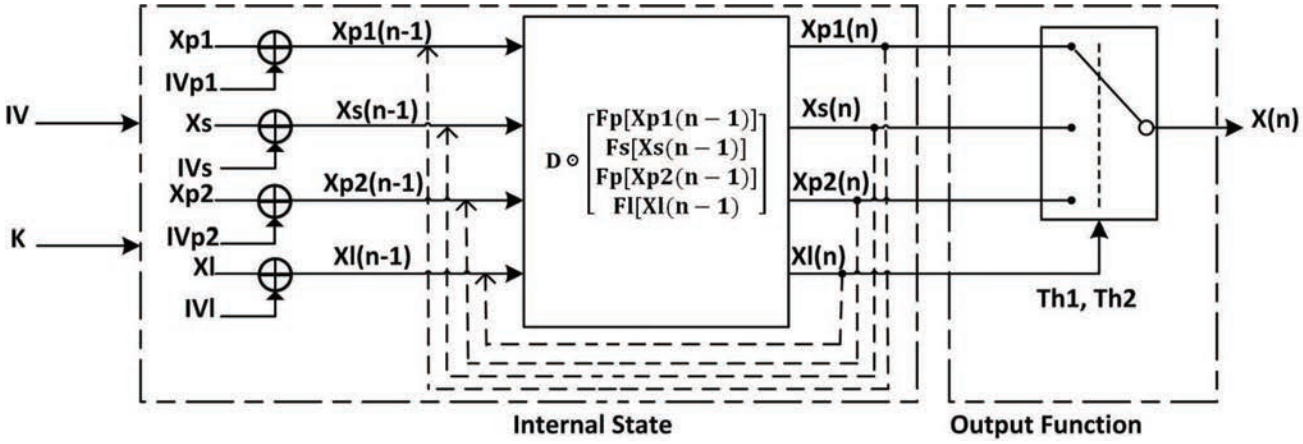



Fig. 2: Structure of the second proposed PCNG2.

TABLE III: Number of cycles per byte of PCNG1 and PCNG2

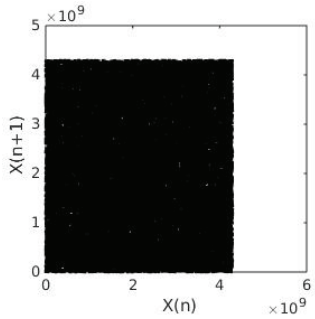
| Data Bytes | PCNG1 NCpB($Cycles/B$) | PCNG2 NCpB($Cycles/B$) |
|---|---|---|
| 64 | 232.5 | 121.06 |
| 128 | 155 | 97.91 |
| 256 | 106.5 | 81.45 |
| 512 | 92 | 67.95 |
| 1024 | 77.5 | 59.69 |
| 2048 | 69 | 59.58 |
| 4096 | 66 | 33.49 |
| 16384 | 50.2 | 31.21 |
| 32768 | 39.3 | 26.87 |
| 65536 | 26.9 | 25.95 |
| 125000 | 25.4 | 25.85 |
| 196608 | 23.1 | 21.17 |
| 393216 | 18.3 | 18.53 |
| 786432 | 17.3 | 14.01 |
| 3145728 | 17.1 | 13.83 |
| 12582912 | 16.8 | 13.78 |

### *B. Security analysis and statistical attacks*

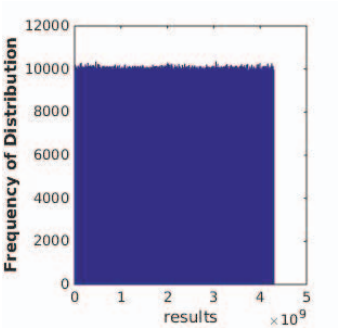
We report below first the security analysis in terms of key size, keystream attack and key sensitivity attack. Then, we give the obtained results of several statistical tests that were carried out in order to quantify the good statistical properties of the proposed PCNGs.

*1) Key size, Keystream attack and Key Sensitivity of the PCNGs :* The key size $|K1|$ and $|K2|$ of the PCNG1 and PCNG2 respectively consists of all initial conditions and parameters of the proposed system and they are large enough to resist the brute force attack. Indeed, $|K1| = 4 \times 32 + 23 + 21 + 32 + 31 + 32 + 32 = 299$ bits, and $|K2| = 4 \times 32 + 32 + 2 \times 31 = 222$ bits.

Also, as for each new execution the produced keystream is totally different from each others, due to the IVg value, so, the system can resist against keystream attack. Besides, the key

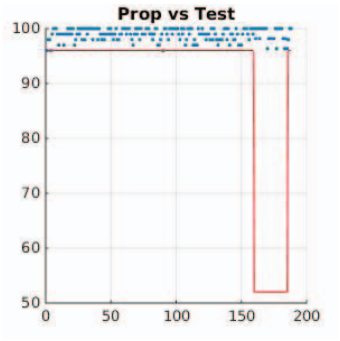



(a) Mapping

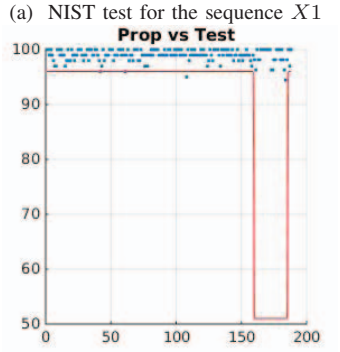



(b) Histogram

Fig. 3: Statistical tests results




(a) NIST test for the sequence $X1$




(b) NIST test for the sequence $X2$

Fig. 4: Statistical tests results

sensitivity is an essential property. This means, a small change in the secret key must cause a very big change in the output keystream. In order to verify this characteristic, we calculate the Hamming Distance (HD) of two sequences generated with only one bit change (lsb bit) in the parameter $X\_p$. We calculate the average Hamming Distance HD between two sequences $S_1$ and $S_2$, over 100 random secret keys. The $HD(S_1, S_2)$ is defined by the following equation:

$$HD(S_1, S_2) = \frac{1}{Nb} \times \sum_{K=1}^{Nb} (S_1(K) \oplus S_2(K)) \qquad (11)$$

Where Nb is the number of bits in a sequence.

The obtained average value of Hamming distance is equal to 0.499999 and 0.499887 for PCNG1 and PCNG2 repetitively. These values are close to the optimal value of 50%, which indicate the high sensitivity on the secret key.

*2) Mapping and Histogram:* The Mapping $(X(n+1) = f(X(n)))$ reflects the dynamic behavior of the system. As we can see in Figure 3a, the resulting mapping of a given produced sequence by the PCNG1 seems to be random in comparison with a mapping (a signature) of a given known map. Similar visual result is obtained for the PCNG2. A good PCNG must produce sequences that have uniform distribution in the whole phase space. Visually, the obtained histogram in Figure 3b for a given generated sequence is uniform. To confirm this result we applied the Chi-Square and we obtained $991.962210$ and $1030.832$ as an experimental value for PCNG1 and PCNG2, which is smaller than the theoretical value $1073.642651$, then the histogram is uniform for the two PCNGs. Also, the uniformity of the sequence generated by the PCNG1 is better than one produced by PCNG2. Indeed, more the experimental value of Chi-Square is smaller than the theoretical one, better is the uniformity of the generated sequence.

*3) Auto and Cross-correlation:* Another good property of a PCNG is that, the generated sequences must be uncorrelated. Thus, the cross-correlation of two sequences $x$ and $y$ (generated with slightly different keys) must be close to zero. The correlation coefficient $\rho_{xy}$ of the two sequences $x$ and $y$ is given by:

$$\rho_{xy} = \frac{\sum_{i=1}^{Ns}(x_i - \bar{x})(y_i - \bar{y})}{[\sum_{i=1}^{Ns}(x_i - \bar{x})^2]^{1/2} \times [\sum_{i=1}^{Ns}(y_i - \bar{y})^2]^{1/2}}. \qquad (12)$$

Where $\bar{x} = \frac{1}{Ns}\sum_{i=1}^{Ns} x_i$ and $\bar{y} = \frac{1}{Ns}\sum_{i=1}^{Ns} y_i$ are the mean values of $x$ and $y$ respectively. The obtained correlation coefficients are $0.0021$ and $0.0030$ for PCNG1 and PCNG2; they are close to zero.

*4) NIST Test:* NIST test is one of the most standards for investigating the randomness of binary data [12]. This test is a statistical package that consists of 188 tests and sub-tests that, were proposed by NIST in order to assess the randomness of an arbitrarily long binary sequences. Figure 4 give the results for sequences $X1$ and $X2$ generated by PCNG1 and PCNG2 respectively.

We observe that, sequence $X2$ does not pass only one sub-test. In contrast, sequence $X1$ has successfully passed all the NIST tests. Therefore, the proposed chaotic generator PCNGs are robust against statistical attacks and the security performance of the first PCNG1 is better than the second one.

## IV. Conclusion

In this paper, we realized a comparative study of the performance in terms of computation time and security of two proposed PCNGs. The analysis study and the obtained results of the two PCNGs show that the two proposed PCNGs have strong cryptographic properties. Security performance of the first proposed PCNG is better than the second one but it is slightly slower. Also, these PCNGs are faster than some proposed generators of the literature and will be used for a secure chaotic stream cipher in development.

## Acknowledgment

The authors would like to thank the European program: Erasmus Mundus scholarship E-GOV-TN, and the European Celtic-Plus project 4KREPROSYS - 4K ultra HD TV wireless REmote PROduction SYStems, 2015.

## References

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

[2] 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.

[3] 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.

[4] 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.

[5] N. Masuda and K. Aihara, "Cryptosystems with discretized chaotic maps," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 49, no. 1, pp. 28–40, 2002.

[6] S. El Assad, "Chaos based information hiding and security," in *Internet Technology And Secured Transactions, 2012 International Conference for*. IEEE, 2012, pp. 67–72.

[7] 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.

[8] S. Lian, J. Sun, J. Wang, and Z. Wang, "A chaotic stream cipher and the usage in video protection," *Chaos, Solitons & Fractals*, vol. 34, no. 3, pp. 851–859, 2007.

[9] M. François, T. Grosges, D. Barchiesi, and R. Erra, "A new image encryption scheme based on a chaotic function," *Signal Processing: Image Communication*, vol. 27, no. 3, pp. 249–259, 2012.

[10] A. K. Hartmann, *Practical guide to computer simulations*. World Scientific, 2009.

[11] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo random number generator," *SIAM J. Comput.*, vol. 15, no. 2, pp. 364–383, May 1986.

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