AIP ADVANCES **8**, 035004 (2018)

A new simple technique for improving the random properties of chaos-based cryptosystems

M. Garcia-Bosque,[a] A. Pérez-Resa,[a] C. Sánchez-Azqueta,[a] and S. Celma[a]
Group of Electronic Design, University of Zaragoza, Pedro Cerbuna, 12, 50009 Zaragoza, Spain

(Received 23 November 2017; accepted 22 February 2018; published online 5 March 2018)

A new technique for improving the security of chaos-based stream ciphers has been proposed and tested experimentally. This technique manages to improve the randomness properties of the generated keystream by preventing the system to fall into short period cycles due to digitation. In order to test this technique, a stream cipher based on a Skew Tent Map algorithm has been implemented on a Virtex 7 FPGA. The randomness of the keystream generated by this system has been compared to the randomness of the keystream generated by the same system with the proposed randomness-enhancement technique. By subjecting both keystreams to the National Institute of Standards and Technology (NIST) tests, we have proved that our method can considerably improve the randomness of the generated keystreams. In order to incorporate our randomness-enhancement technique, only 41 extra slices have been needed, proving that, apart from effective, this method is also efficient in terms of area and hardware resources. https://doi.org/10.1063/1.5017015

I. INTRODUCTION

The design and implementation of new encryption systems that are secure and capable of ciphering binary sequences with a high throughput is currently an important field of research.[1]

Typically, digital ciphers can be divided in asymmetric ciphers and symmetric ciphers. In asymmetric ciphers, a public key is used to cipher the plaintext and a private (secret) key is used to decrypt the ciphertext while, in symmetric ciphers, there is only a secret key shared by both the transmitter and the receiver and it is used to encrypt and decrypt the messages. At the same time, symmetric ciphers can be divided in block ciphers, where the plaintext is divided by blocks and each block is encrypted individually (e. g. AES) and stream ciphers, which encrypt each bit individually. The main advantage of the stream ciphers is that they do not have to save blocks of data so they can save memory resources. Furthermore, they do not need to add extra bits in case that some blocks are not full (padding). Therefore, stream ciphers are, in general, more suitable for high speed communications.[2]

In these cryptosystems, a key (seed) and a certain algorithm are used to generate binary sequence (keystream) that is then used to encrypt the message (Fig. 1). Some of the algorithms that have arisen much interest in the last years are based on chaotic maps. Chaotic systems present some intrinsic properties such as ergodicity, high sensitivity to the initial conditions and random-like behavior that are strongly related with the properties of confusion and diffusion that were identified by Shannon as the key properties of a secure cipher.[3] Therefore, many chaos-based stream ciphers have been recently proposed as an alternative to conventional encryption and, some of them, have proven to be suitable for applications that require high speed encryption such as real-time video transmission.[4]

[a]mgbosque@unizar.es, aprz@unizar.es, csanaz@unizar.es, scelma@unizar.es

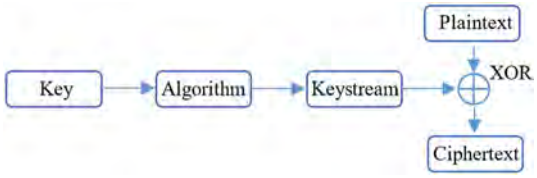

FIG. 1. Overall scheme of a stream cipher.

On the other hand, most of the chaos-based cryptosystems proposed so far can present some issues that make them unideal for secure applications. Some of those issues are the absence of a defined key space[5] and the poor randomness of the generated sequences as a consequence of the digitization of the algorithm.[6]

In this work, we propose a new technique to improve the security of chaos-based stream ciphers at a small cost of implementation resources. As an example, this technique has been applied to a particular algorithm based on a Skew Tent Map[7] and has been implemented on a Virtex 7 FPGA.

The paper is organized as follows: Section II presents an overview of the chaos-based stream ciphers and explains some of the problems that arise when they are implemented in hardware; Section III proposes new techniques to improve the security of these chaotic ciphers; Section IV presents the implementation and results of some proposed cryptosystems that use the previous techniques; finally, preliminary conclusions are drawn in Section V.

II. IMPLEMENTATION OF CHAOTIC STREAM-CIPHERS

A. General scheme of a chaos-based stream cipher

A stream cipher is a cryptosystem that starting from some initial parameters (seed or key) and using a certain algorithm generates a binary sequence (keystream). This keystream is XORed with the plaintext in order to generate the ciphertext. In order to decrypt the message, the receiver must use the same key and the same algorithm in order to generate an identical keystream. By XORing this keystream with the ciphertext, the original message is recovered.

A simple way of implementing a chaos-based stream cipher consists on using a chaotic map of the form:

$$x_{i+1} = f(x_i, \gamma) \tag{1}$$

where γ is a chaotic parameter that is constant for the whole sequence and x_i are the elements of the generated sequence.

Therefore, starting from an initial value of x_0 and a chaotic parameter γ, a sequence of randomly distributed elements $\{x_i\}$ is obtained. In this case, the key would be given by the values of x_0 and γ. In order to use this system as a stream cipher, the bits that compose each x_i can be used to build the keystream. However, typically there is a strong correlation between the bits of a given x_i. Furthermore, when all the bits of each x_i are used to encrypt the message, the receiver could obtain a lot of information about the values of $\{x_i\}$ and could use (1) to find the key of the system. Therefore, only a few bits of each x_i (preferably the least significant bits) are usually used to conform the keystream.

The security of these cryptosystems relies, among others, on the randomness of the generated sequences. Therefore, it is crucial to assure that the stream ciphers are capable of generating sequences with good random properties.

B. Effects of the digitization in the chaotic behaviour of the system

The main property of the chaotic system is its high sensitivity to the initial conditions. Therefore, two systems that use the same algorithm and the same initial parameters (x_0, γ) but different bit precision (number of bits for representing each x_i) will generate different orbits that will separate from each other very fast. Consequently, in order to encrypt and decrypt a message properly, both the transmitter and the receiver should use the same bit precision.

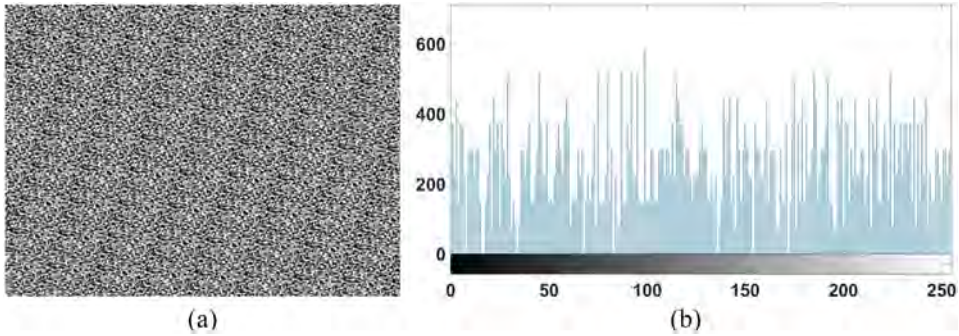

FIG. 2. (a) Binary sequence generated by a 32-bit Logistic Map from randomly chosen initial parameters γ, x_0. Although the maximum possible period is $2^{32} \approx 4 \times 10^9$, the period obtained in this case is 882. (b) Histogram of the sequence in (a).

The second problem that arises when one of these systems is digitized is the fact that the orbits become periodic. If x_i is represented using n bits, there will only be 2^n possible values for each x_i. Therefore, since each x_i determines the next value of the sequence, x_{i+1}, once a value of x_i is repeated, the rest of the sequence will also be repeated. Therefore, the maximum period of each sequence will be $P_M = 2^n$. In practice, however, the mean period of the sequences, $\bar{P}$, is usually much shorter than the maximum possible period:[8]

$$\bar{P} \approx \frac{1}{4}\sqrt{2\pi 2^n} \tag{2}$$

It must be noticed that, by increasing the precision in 2 bits, although the maximum period is increased by a factor of 4, the mean period is only increased by a factor of two. Typical precisions used such as 32 or 64 bits causes that the generated sequences have short average periods which results in most of the sequences failing the randomness tests regardless of the algorithm used. As an example, Fig. 2(a) shows in a matrix of white and black pixels the bits obtained from a particular sequence generated by a Logistic Map using a 32-bit precision. Its corresponding histogram is shown in Fig. 2(b). As it can be seen, the sequence is far from random.

More complex maps (i.e. multidimensional maps), with many parameters involved would have longer maximum and mean period lengths. However, although this approach has been used in the literature,[9,10] these cryptosystem usually present a poor performance in terms of encryption speed per area. Therefore, in this work we have focused on increasing the period length of "simple maps".

The first approach to minimize the digitization consists on using a precision of more bits. For example, Ref. 11 uses a precision of 500 bits. This strategy, however, supposes to increase the complexity of the system considerably in order to obtain long average periods. Therefore, it is not an optimum strategy, especially for applications that require encrypting sequences at high speed using a small amount of hardware resources.

Another approach consists on perturbing the orbits by adding some noise to each x_i before each iteration. For example, in Refs. 12 and 13 the least significant bit of each x_i is XORed with a bit generated by a Linear Feedback Shift Register (LFSR). This way, it is possible to prevent the system to stay in a stable orbit and, using the correct pseudorandom sequence, the period can be guaranteed to be bigger than a given number. A possible drawback of this system is that it could be considered that the randomness improvement comes from the LFSR and not from the chaotic system itself. Furthermore, it is well-known that, while LFSRs are good at generating random sequences, they are not secure systems.

In this work, a different approach is used in order to improve the random properties provided by a chaos-based pseudorandom number generator.

III. RANDOMNESS IMPROVEMENT PROPOSAL

Let's consider that we have a stream cipher that, in order to generate the keystream, uses a chaotic map of the form (1). We propose to use, instead of a single value of γ, a set of m different

values of γ: $\gamma_1, \gamma_2, \ldots \gamma_m$. In order to generate the sequence $\{x_i\}$, the value of γ used is continuously changing in a circular way, according to a predefined sequence partition $\{k_i\}$. This way, the first k_1 elements of the sequence are generated as $x_i = f(x_{i-1}, \gamma_1)$, the next k_2 elements are obtained as $x_i = f(x_{i-1}, \gamma_2)$ and so on. When all γ_i have been used, and a total of $\sum_{i=1}^{m} k_i$ elements have been generated, the first value of γ, γ_1 is used again and the process keeps going in a circular fashion.

In the following subsections, we propose and analyze different approaches of choosing the values of m and $\{k_i\}$ in order to optimize the performance of the system.

A. Fixed value of sequence partition k

The first approach consists on generating the same number of elements using each γ_i, i. e. $k_1 = k_2 = \cdots k_m = k$. In the previous case, m=1, we had that, for each value of x_i we always ended up in one of the 2^n possible values of x_{i+1}. In this case, however, we can end up in m different values of x_{i+1} (one for each γ_i). This way, the theoretical maximum period of the generated sequences using this method is increased by a factor of m, $P_M = m2^n$. Along this increment, it can be found experimentally that there is a considerable increment in the mean period length.

The main problem of this approach is that, by using always a fixed value of k and m, there is always a possibility that the sequences stay in a stable short period orbit. In order to avoid this problem, a possible alternative consists on changing constantly the values of k_i, as will be explained in the next subsection.

B. Dynamically changed value of sequence partition k

In this case, a variable number of k_i is generated within a range $k_i \pm \Delta k_i$ and is used to encrypt the message. When all the values of γ_i have been used, a new value of k_1 is generated. This way, the sequences will conserve the characteristic properties of the chaotic map (ergodicity, high sensitivity to the initial conditions), but they will not stay at a stable orbit.

Since the important thing of this method is to introduce small variable values of k_i, it is not necessary to use a perfect random number generator. Therefore, any random number generator, or even a simple algorithm that uses some parts of the plaintext or other parameters of the traffic could be used. For example in a packed stream, the interframe gap or the packet size could be used.

IV. EXPERIMENTAL RESULTS

A. Skew tent map algorithm

In order to test the proposed randomness enhancement technique, a cryptosystem based on a Skew Tent Map (STM)[7] has been implemented on a Virtex 7 FPGA.

The STM is defined by:

$$f(x_i) = x_{i+1} = \begin{cases} x/\gamma, & x_i \in [0, \gamma] \\ (1-x)/(1-\gamma), & x_i \in (\gamma, 1] \end{cases} \tag{3}$$

where $\gamma, x_0 \in (0,1)$.

This map has been chosen because it is a simple system that presents a chaotic behavior for any initial value of x_0 and γ (i.e., it does not present periodic windows) and it can produce values of x_i that are uniformly distributed in the interval (0,1).[14] Therefore, after implementing this system on the FPGA, most of the problems of the generated sequences will be caused by the digitization of the system and not by the intrinsic properties of the chaotic map. Therefore, by proving that our method is capable of improving the randomness of the sequences generated by a STM-based algorithm, it is possible to extrapolate the results to other chaotic cryptosystems. Preliminary results of this work has been accepted for presentation in Ref. 15.

The STM has been implemented using a fixed-point 32-bit precision and only the least significant bit of each x_i has been used to generate the bitstream. In order to test our proposed method, we have generated the sequences using different values of m (number of γ_i) and k_i (number of bits encrypted with each γ_i). Since the divisions can be costly to implement in an FPGA, we have pre-calculated

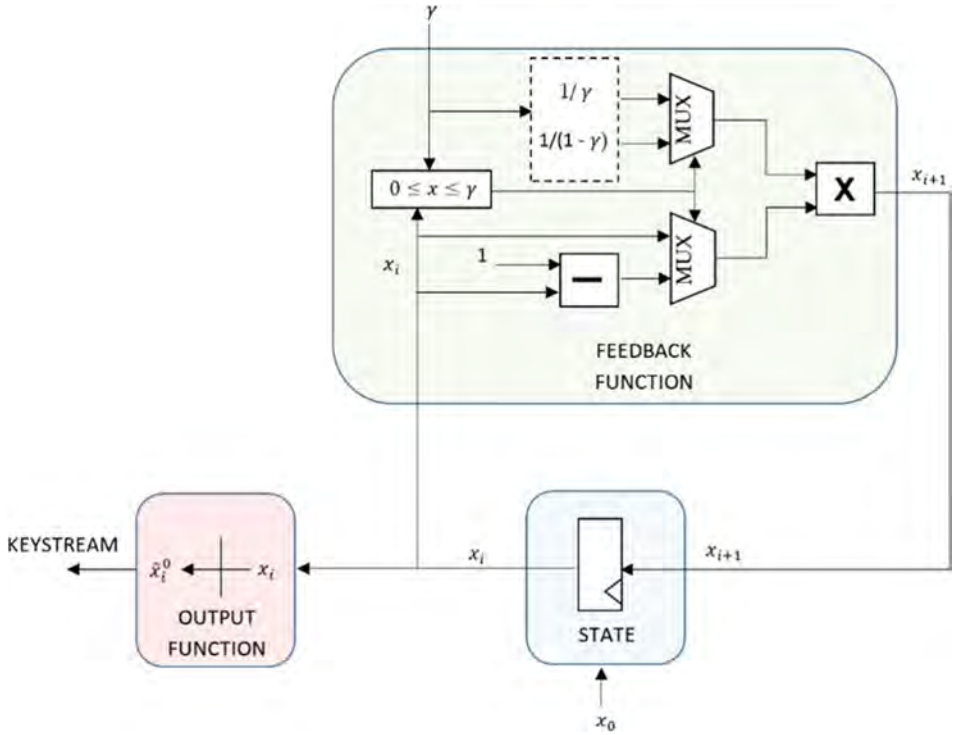

FIG. 3. Block diagram of a Skew Tent Map generator.

and saved the values of $1/\gamma_i$ and $1/(1-\gamma_i)$ and used multiplications instead. A block diagram of the STM generator is shown in Fig. 3 while, a scheme of the complete implementation of the enhanced stream cipher that uses our randomness improvement technique and the STM generator is shown in Fig. 4.

B. Results for fixed value of k

First, we have started using fixed values of k_i ($k_1 = k_2 = \ldots = k$). In order to test the randomness of the sequences generated for each case, 100 sequences have been generated and have been subjected to the National Institute of Standards and Technology (NIST) randomness test[16] with a significance level of 0.01. The NIST test suite provides a list of all the tests, indicating how many sequences have passed each of them. To easily compare the results among different systems, the passing rates for each test (fraction of sequences that have passed each test) have been calculated and, after that, we have calculated the average of all of them. Results are shown in Table I.

As it can be seen, increasing m improves the randomness of the generated sequences. This can be explained by the fact that, by increasing the value of m, the maximum and the mean period of the sequences increases. This effect is especially noticeable for small values of m, since the effect of the increment of the period lengths of the sequences in the randomness test performance is bigger for small periods. On the other hand, by increasing the value of k, the randomness of the sequences also improves. This could be explained by the fact that, when a small value of k is used, groups of bits generated using the same value of γ are separated by a small distance inside the keystream.

Therefore, the NIST tests, might find some correlations among them. However, for bigger values of k, bits generated using the same γ are more separated so it becomes harder to find those correlations.

It can be noticed that, although, the randomness improvement is significant, with the significance level of 0.01, 99% of the sequences should pass the randomness tests. Therefore, the sequences generated using this method are far from random. Although the results could be improved using bigger values of m and k, it would involve an increment of the hardware resources used.

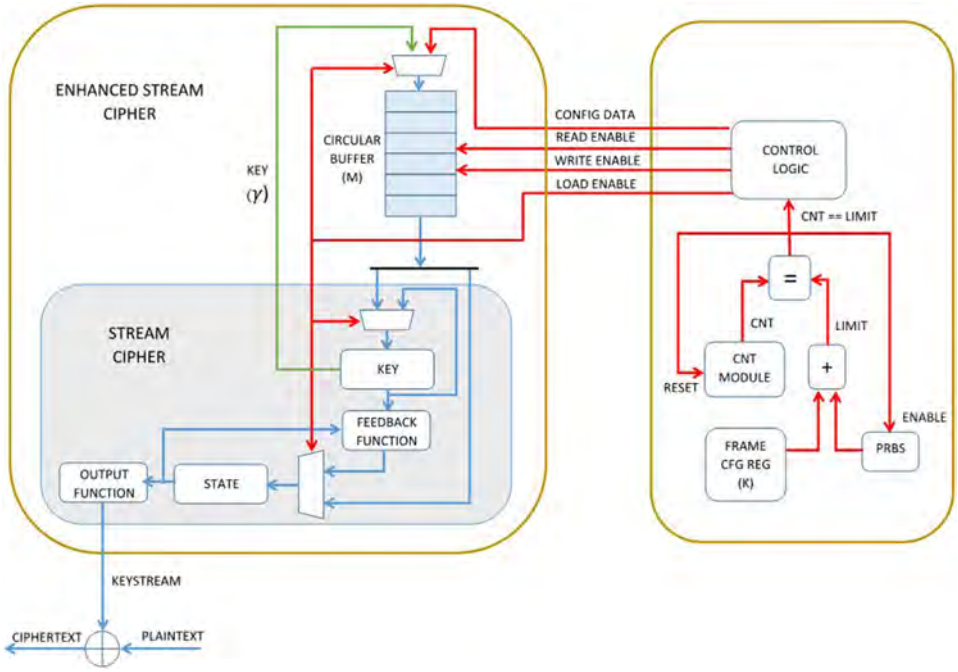

FIG. 4. Scheme of the proposed enhanced stream cipher. The values of γ_i are stored in a FIFO that works as a circular buffer. The control module performs the gamma change every time an amount of bits equal to the configured packet size plus a percentage is transmitted. This bit number limit is set thanks to the LIMIT value, which corresponds to the sum of the packet size value configured in FRAME CFG REG plus a pseudorandom value generated by the PRBS module.

TABLE I. Nist Test Results for fixed values of K.

k	10	100	1,000
m			
1 (32-bit)	0.317	0.317	0.317
2	0.512	0.659	0.750
4	0.552	0.720	0.845
8	0.568	0.777	0.856
16	0.581	0.786	0869

C. Results for variable k

Finally, the k variable case has been tested. We have used the same values of γ_i as before but each k_i is a random integer generated within a certain range. Table II shows the results for different

TABLE II. Nist Test Results for Variable values of K.

k	9-11	90-110	900-1100
m			
1 (64-bit)	0.989	0.989	0.989
2	0.938	0.935	0.857
4	0.971	0.932	0.910
8	0.964	0.944	0.929
16	0.955	0.937	0.901

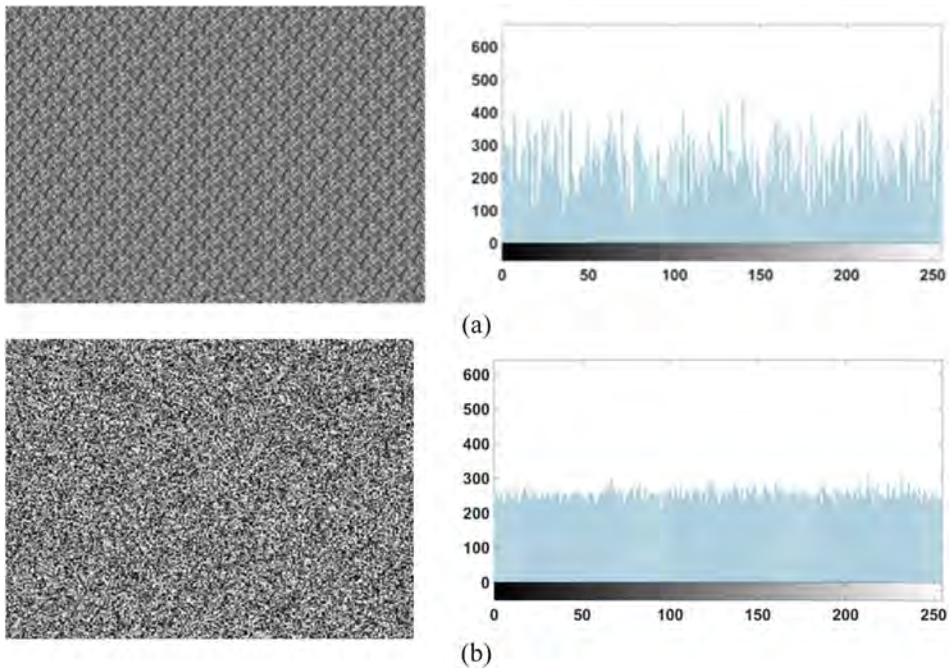

FIG. 5. (a) Binary sequence generated by a 32-bit Skew Tent Map from randomly chosen initial parameters γ, x_0 with its corresponding histogram. Due to its short period, 3048, the sequence presents poor random properties. (b) Binary sequence generated by a Skew Tent Map with our randomness enhancement technique with its corresponding histogram. Its period has not been calculated because it was bigger than the number of bits generated used in this figure, 65536.

ranges. For comparison purposes, the table also includes the results obtained by a system using only one value of γ ($m = 1$) but 64-bit (double) precision.

As it can be seen, the randomness tests results are much better that for the fixed-k case since, as it has been mentioned before, this method helps the system to avoid stable orbits. To visualize the improvement provided by this technique, Fig. 5 represents in a binary matrix and in a histogram a keystream generated by a Skew Tent Map and keystream generated by a Skew Tent Map with our randomness enhancement technique.

In the best case, for $m = 4$ and k=9-11, the randomness improvement is very similar to the improvement obtained by using a 64-bit precision. However, implementing the STM using a 64-bit precision requires far more hardware than implementing the 32-bit precision STM with the proposed randomness enhancement technique (Table III).

It must be noticed that the amount of extra slices needed to implement our technique does not depend on the algorithm used. In this case, since we used one of the simplest chaotic maps, the increment in the number of slices is considerable compared to the number of slices required to implement the Skew Tent Map algorithm ($\sim$ 40%). However, if a more complex chaotic map had been used, the increment in the number of slices could be negligible. However, the number of extra

TABLE III. Implementation Results.

Configuration	32-bit STM	64-bit STM	Enhanced STM
LUTs	369	805	440
Registers	39	70	113
Slices[a]	98	210	139
Total DSPs	11	16	11
RAM blocks	0	0	0
NIST passing rate	0.317	0.989	0.971

[a]The number of slices has been estimated from the number of registers and LUTS assuming unrelated logic.

slices needed to implement a chaotic map with a higher precision increases with the complexity of the algorithm. Therefore, we can conclude that our strategy is a much better approach.

D. Other security aspects

Although this method has been proposed to improve the randomness properties of the generated sequences, it can also improve other aspects related to the security.

First, to prevent brute-force attacks, the key space size of the system should be big enough. In the case of the Skew Tent Map based cryptosystem, implemented using a single value of γ and a 32-bit precision, the key space size, κ_{STM}, is given by all possible initial values of x_0, γ:

$$\kappa_{STM} = 2^{32+32} = 2^{64} \tag{4}$$

According to some guidelines, to be secure, the key space size of a cryptosystem should be at least $\kappa = 2^{112}$.[17,18] Therefore, this system would not be secure from this point of view.

However, with our proposed technique the key space size would be given by all the possible values for the initial element of the sequence, x_0, all possible values of the chaotic parameters γ_1, γ_2, ... γ_m and, finally, all the possible sequence partitions, $n_{\{k_i\}}$. We do not provide a concrete value for $n_{\{k_i\}}$ since it may vary depending on the implementation. Therefore, the key space in this case, $\kappa_{Enhanced-STM}$, would be given by:

$$\kappa_{Enhanced-STM} = n_{\{k_i\}} \times 2^{32(m+1)} \tag{5}$$

Therefore, as it can be seen, the key space size is considerably increased. Indeed, by using a value of $m > 4$, we guarantee that $\kappa > 2^{112}$ and, therefore, the system is secure against brute-force attacks.

On the other hand, some chaotic maps can present a periodic (non-chaotic) behavior for certain values of their chaotic parameters γ. The set of parameters that leads to a chaotic behavior is not usually within a continuous region and it is often unknown.[5]

As an example, a bifurcation diagram of the Logistic Map, which represents several consecutive values of the sequences x_i as a function of the chaotic parameter γ, is shown in Fig. 6. Although most values of $\gamma > 3.57$ present chaotic behavior, there are some white regions where the system exhibits a periodic behavior. Moreover, even in the chaotic regions, there exist an infinite number of initial conditions that lead to periodic cycles.[19] Several methods have been capable to find some of the periodic windows,[20,21] but the whole set of parameters that produces periodic windows remains unknown. Therefore, even by using an infinite precision, the system could fall into a periodic orbit if an unsuitable value of γ was used.

Let's call P_c the probability that a value of γ randomly chosen is "good" (i.e. an orbit generated by that γ exhibits a chaotic behavior) and $P_p = 1 - P_c$ the probability that the value of γ is "bad" (i.e. an orbit generated by that γ exhibits a periodic behavior). Usually, the probability of choosing an unsuitable value of γ will be very small (i.e., $P_p \ll P_c$). However, when this happens, the behavior of the cryptosystem will be very poor. Using our technique, since we are using m different values of γ,

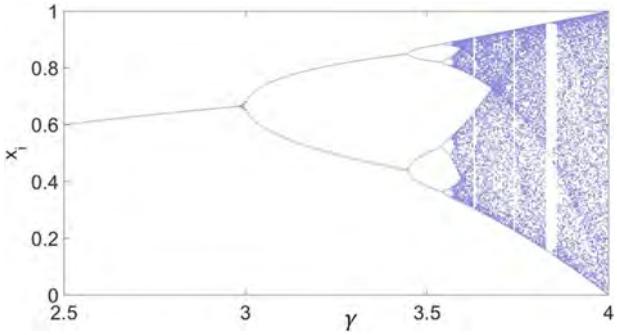

FIG. 6. Bifurcation map of a Logistic Map.

the probability of having l unsuitable values of γ among then will be:

$$P(m,l) = \binom{m}{l} P_p^l \left(1 - P_p\right)^{m-l} \approx \binom{m}{l} P_p^l \left(1 + (l-m) P_p\right) \tag{6}$$

In this case, the probability of having at least one "bad" value of γ will be higher than before:

$$\sum_{l=1}^{m} P(m,l) = 1 - P(m,0) = 1 - \left(1 - P_p\right)^m > P_p \tag{7}$$

However, the probability of having l "bad" values of γ decreases very fast when increasing l. Therefore, even if one or some of the chosen values of γ are "bad", most of them will probably be good. Therefore, the overall behavior of the system will most likely not be as bad as in the previous case.

In conclusion, using our technique, the probability of generating a non-ideal keystream (i.e. with at least one "bad" value of γ) would be higher, but the probability of generating a very poor keystream (i. e. with most of the chosen values of γ "bad") would be lower.

V. CONCLUSIONS

In this work, a new technique for improving the randomness of the binary sequences generated by chaos-based stream ciphers has been proposed. In order to test the system, a stream cipher based on a Skew Tent Map has been implemented on an FPGA. The results show that, by using several different values of the chaotic parameter γ, the quality of the sequences can be considerably improved. This effect is especially notable when, the number of bits k_i generated with each γ_i is constantly changing over time.

The amount of extra resources needed to implement this technique is quite small compared to other proposed techniques and can be negligible if a high complexity chaotic map is chosen.

Finally, a comparison of our system with a system using a single value of γ but a higher precision has been presented. The results show that our algorithm can produce similar results but using much less hardware resources. It must be remarked that, although a Skew Tent Map has been used for testing purposes, the proposed technique could be applied to other similar chaotic maps.

ACKNOWLEDGMENTS

This work has been supported by "Ministerio de Economía y Competividad" MINECO-FEDER (TEC2014-52840-R and TEC2017-85867-R) and "Formación de Profesorado Univertsitario" FPU fellowship to M. Garcia-Bosque (FPU14/03523).

[1] J. Katz and Y. Lindell, *Introduction to Modern Cryptography* (Chapman and Hall, 2008).
[2] M. Stamp, *Information Security: Principles and Practice* (Wiley, New York, 2011).
[3] C. E. Shannon, "A mathematical theory of cryptography," Bell System Technical Memo MM, pp. 45–110-02, 1945.
[4] R. Hasimoto-Beltran, "High-performance multimedia encryption system based on chaos," Chaos: An Interdisciplinary Journal of Nonlinear Science **18**(2), 023110-1–023110-8 (2008).
[5] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," International Journal of Bifurcation and Chaos **16**(8), 2129–2151 (2006).
[6] M. Garcia-Bosque, C. Sánchez-Azqueta, G. Royo, and S. Celma, "Lightweight ciphers based on chaotic Map-LFSR architectures," in *12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, Lisbon, Portugal, 2016.
[7] A. Baranovsky and D. Daems, "Design of one-dimensional chaotic maps with prescribed statistical properties," International Journal of Bifurcation and Chaos **5**(6), 1585–1598 (1995).
[8] B. Harris, "Probability distributions related to random mappings," Annals of Mathematical Statistics **31**, 1045–1062 (1960).
[9] K.-W. Wong, B.-H. Kwok, and C.-H. Yuen, "An efficient diffusion approach for chaos-based image encryption," Chaos, Solitons & Fractals **41**(5), 2652–2663 (2009).
[10] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," Optics Communications **285**(1), 29–37 (2012).
[11] J. Machicao and O. M. Bruno, "Improving the pseudo-randomness properties of chaotic maps using deep-zoom," Chaos: An Interdisciplinary Journal of Nonlinear Science **27**(5), 053116-1–053116-14 (2017).
[12] M. Garcia-Bosque, C. Sánchez-Azqueta, and S. Celma, "Secure communication system based on a logistic map and a linear feedback shift register," in *International Symposium on Circuits and Systems (ISCAS2016)*, Montreal, Canada, 2016.
[13] M. Garcia-Bosque, A. Pérez, C. Sánchez-Azqueta, and S. Celma, "Application of a MEMS-based TRNG in a chaotic stream cipher," MDPI Sensors **17**(3), 1–15 (2017).

[14] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," International Journal of Bifurcation and Chaos **14**, 3119–3151 (2005).
[15] M. Garcia-Bosque, A. Pérez-Resa, C. Sánchez-Azqueta, and S. Celma, "A new technique for improving the security of chaos based cryptosystems," *de International Symposium on Circuits and Systems*, Florence, Italy, 2018. (Accepted).
[16] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "NIST Special Publication 800–22 Rev.1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2010.
[17] ENISA, Algorithms, Key Size and Parameters Report, 2014.
[18] E. Barker and A. Roginsky, "Transitions: recommendation for transitioning the use of cryptographic algorithms and key lengths," NIST Special Publication 800–131 A.
[19] D. P. Feldman, *Chaos and Fractals. An Elementary Introduction* (Oxford University Press, 2012).
[20] W. Tucer and D. Wilczak, "A rigorous lower bound for the stability regions of the quadratic map," Physica D **238**(18), 1923–1936 (2009).
[21] Z. Galias and B. Garda, "Detection of all low-period windows for the logistic map," *de Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, 2015.