BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 65, No. 5, 2017
DOI: 10.1515/bpasts-2017-0078

Hardware Trojans detection in chaos-based cryptography

M. MELOSIK[1]*, P. SNIATALA[1], and W. MARSZALEK[2]

[1]Poznan University of Technology, Department of Computer Science, 3A Piotrowo St., 61-138 Poznan, Poland
[2]Rutgers University, Department of Mathematics, 110 Frelinghuysen Road, Piscataway, NJ 08854, USA

Abstract. The paper deals with the security problems in chaotic-based cryptography. In particular, the 0–1 test for chaos is used to detect hardware Trojans in electronic circuits – generators of chaotic bit sequences. The proposed method of detecting hardware Trojans is based on analyzing the original bit sequences through the 0–1 test yielding a simple result, either a number close to 1, when the examined bit sequence is chaotic, or a number close to 0, when the sequence is non-chaotic. A complementary result is a graph of translation variables q_c and p_c which form a basis of the 0–1 test. The method does not require any extra corrections and can be applied to relatively short sequences of bits. This makes the method quite attractive as the security problems are dealt with at the chaotic generator level, with no need to apply any extractors of randomness. The method is illustrated by numerical examples of simulated Trojans in chaotic bit generators based on the analog Lindberg circuit as well as a discrete system based on the logistic map.

Key words: chaos-based cryptography, hardware Trojans, 0–1 test for chaos, bit generators.

1. Introduction

Modern cryptography techniques use phenomena related to quantum physics, chaotic theory, and are based on algebraic structures of elliptic curves over finite fields. One of the main areas of research in chaotic cryptography is the security of the random binary sequence generators. It is known that chaotic signals can be used as a source of random bit sequences for generator applications [1, 2]. An important issue in such an implementation is to prevent the possibility to predict the generated sequence and to make it impossible to reconstruct the generated bitstream, for example, by the method of synchronization of the generators [2]. Another weak point of chaotic generators is the possibility of generating periodic signals due to a finite length representation of real numbers [3]. These two issues can disturb the proper use of the generated random sequences, and, in consequence, can lead to a predition of the generated bitstreams. Obviously, such problems reduce the security of the whole cryptography system. Malicious hardware (circuit) modifications, known as hardware Trojans (HTs), may result in the above mentioned security problems.

The current embedded systems are implemented as printed circuit boards (PCBs) with typical modules (units) shown in Fig. 1. The number of modules depends on a particular application and specificity of an embedded system [4]. In this paper HTs are assumed to be possible in the hardware modules responsible for the generation of random sequences. As shown in [1–3, 5], one of the acceptable methods of generating random sequences is to use chaotic circuits. Such an approach is straightforward when compared to the use of quantum modules, which require significantly larger areas on PCBs. A random sequence generator can be integrated with a security module or it can itself be a separate module. The security issues of embedded systems, such as PCBs, were first discussed in [6]. However, [6] does not discuss the security problems and hardware Trojan attacks on random sequence generators based on chaotic circuits. This paper deals with such problems based on testing of chaotic signals by the 0–1 test for chaos. The result of such test is two-fold: a single number, close to either 0 or 1, and a visual two-dimensional plot of the so-called translation variables q_c and p_c, that in future might be implemented online in real time.

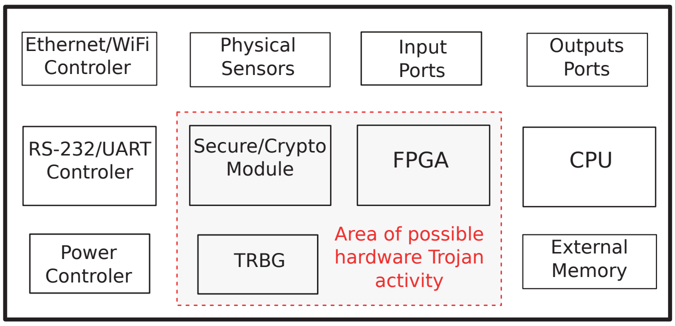


Fig. 1. Modern embedded systems as PCBs

As discussed in [6–10], the security issues and protection against HTs is a key problem in modern electronics and computer engineering. In this paper we discuss HTs that may easily be implemented and integrated with the PCB modules in chaotic analog generators. We also discuss HTs implemented through FPGA in chaotic discrete generators.

*e-mail: michal.melosik@put.poznan.pl

Manuscript submitted 2017-02-28, initially accepted for publication 2017-04-21, published in October 2017.

The possible HT attacks are illustrated through the Lindberg analog chaotic circuit and the logistic equation based discrete bit generator. In order to detect the unauthorized modification of the generator's structure, we propose to use the 0–1 test for chaos. The paper has the following structure. The next section delivers general characterization of HTs. Lindberg's circuit and its typical outputs are also presented and possible hardware attacks on the generator are described. Section 3 presents the 0–1 test for chaos as a tool to check the chaotic generator security. Results of applications of the 0–1 test are presented in section 4, which is followed by a concluding section 5.

2. Hardware Trojans

In order to increase security of modern computer systems, which are vulnerable to countless software viruses, currently the crucial parts of the cryptography systems are implemented as hardware units. Such an approach allows for a stronger protection of the systems, since the hardware layer is separated from the software, and therefore, such systems are resistant to software viruses. Unfortunately, during the last few years, we observed an increased number of hacker attacks on the hardware layer as well [7–9]. Such attacks, or HTs, may take the forms of unauthorized modifications of the values of certain circuit elements in a chosen module of the cryptography system or in the layout (structure) of integrated circuits. Those modifications are usually more difficult to be detected than software viruses. A basic classification of HTs is presented in Fig. 2.

The classification differentiates HTs according to the following criteria: stage where a HT is introduced, level of the attacked system, activation method, HT's action results, and localization of a HT in the infected system. Quite often we need to deal with problems occuring at different locations in the system. Also, HTs can be activated in many different ways, and they can result in various unwanted outcomes. Significant research efforts have been undertaken to detect such security threats. There are several proposed solutions related to digital systems [10]. However, for the cryptography systems utilizing analog chaotic circuits the security issues are still open [6].

2.1. HTs in chaotic generators. The cryptography systems require an intensive use of pseudorandom number generators having strong security properties. One of the possibilities is a hardware implementation of chaotic generators. The generated analog chaotic signal is converted to a bitstream using a simple threshold circuit. A possible technology for realization of such generators is the PCB implementation, which is less expensive than the integrated circuits. In the past, the problem of unauthorized modifications of such circuits was often neglected [1, 6, 11].

Obviously, the layouts of PCBs can be modified and certain elements can be replaced by elements with different parameters (values). Currently, the problem of PCB security becomes a crucial one in modern electronics [6].

Chaotic generators are sensitive to initial conditions. This feature makes them the sources of high entropy random signals. As a result, two separate generators, having the same structure and the same element values can generate different chaotic signals. One of the possible hardware attacks on such generators is to change one or more element values to obtain a periodic signal. According to the classification presented in Fig. 2, such an attack can occur at the circuit assembly stage. This type of modification usually stays active for a long period of time and significantly reduces security of the system. To illustrate the problem, we first use the Lindberg chaotic circuit [12] shown in Fig. 3a in which various HTs are simulated.

The possible hardware attacks in this circuit, which can affect the statistical properties of the generator, could be the modification of the AC source frequency value f_{AC} or of the re-

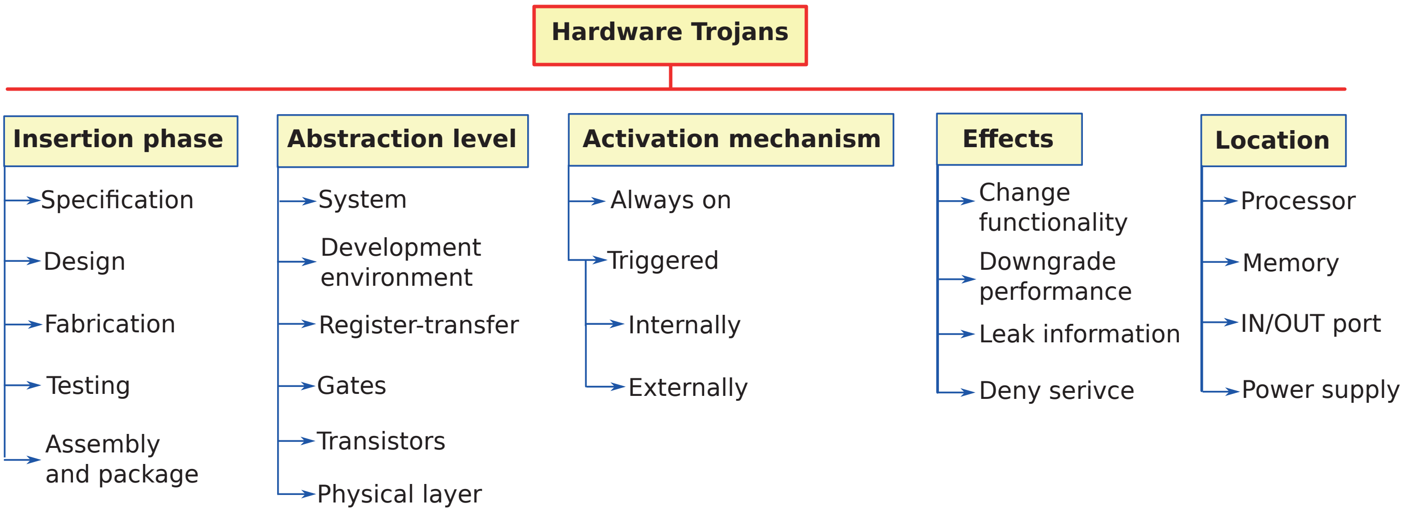
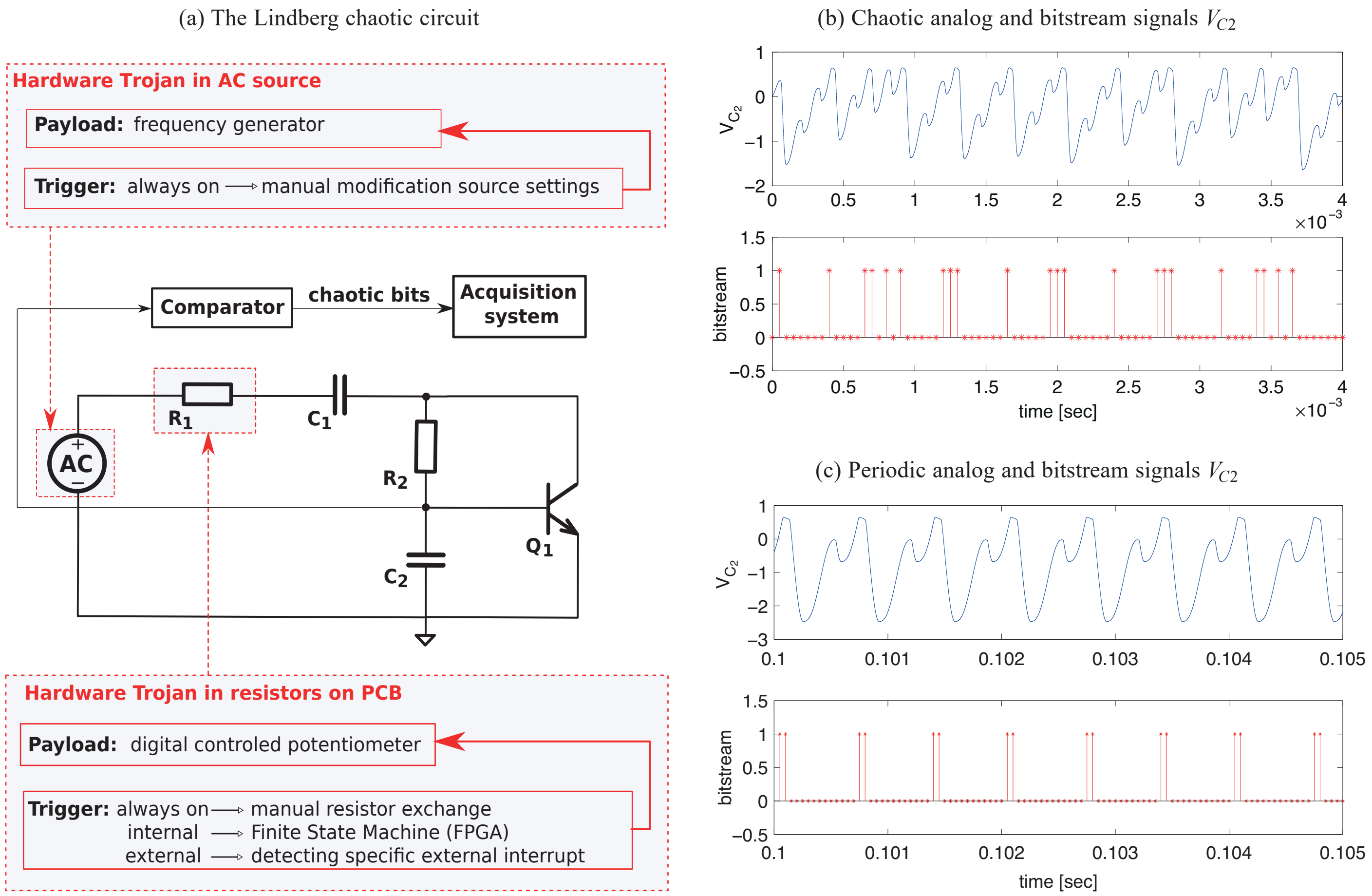


Fig. 2. Classification of the currently known HTs [11]


(a) The Lindberg chaotic circuit

(b) Chaotic analog and bitstream signals V_{C2}

(c) Periodic analog and bitstream signals V_{C2}

Hardware Trojan in AC source
Payload: frequency generator
Trigger: always on → manual modification source settings
Comparator
chaotic bits
Acquisition system
AC
+
–
R1
C1
R2
C2
Q1
Hardware Trojan in resistors on PCB
Payload: digital controled potentiometer
Trigger: always on → manual resistor exchange
internal → Finite State Machine (FPGA)
external → detecting specific external interrupt
V_C2
bitstream
time [sec]
×10^−3
0
0.5
1
1.5
2
2.5
3
3.5
4
−2
−1
−0.5
−3
0.1
0.101
0.102
0.103
0.104
0.105

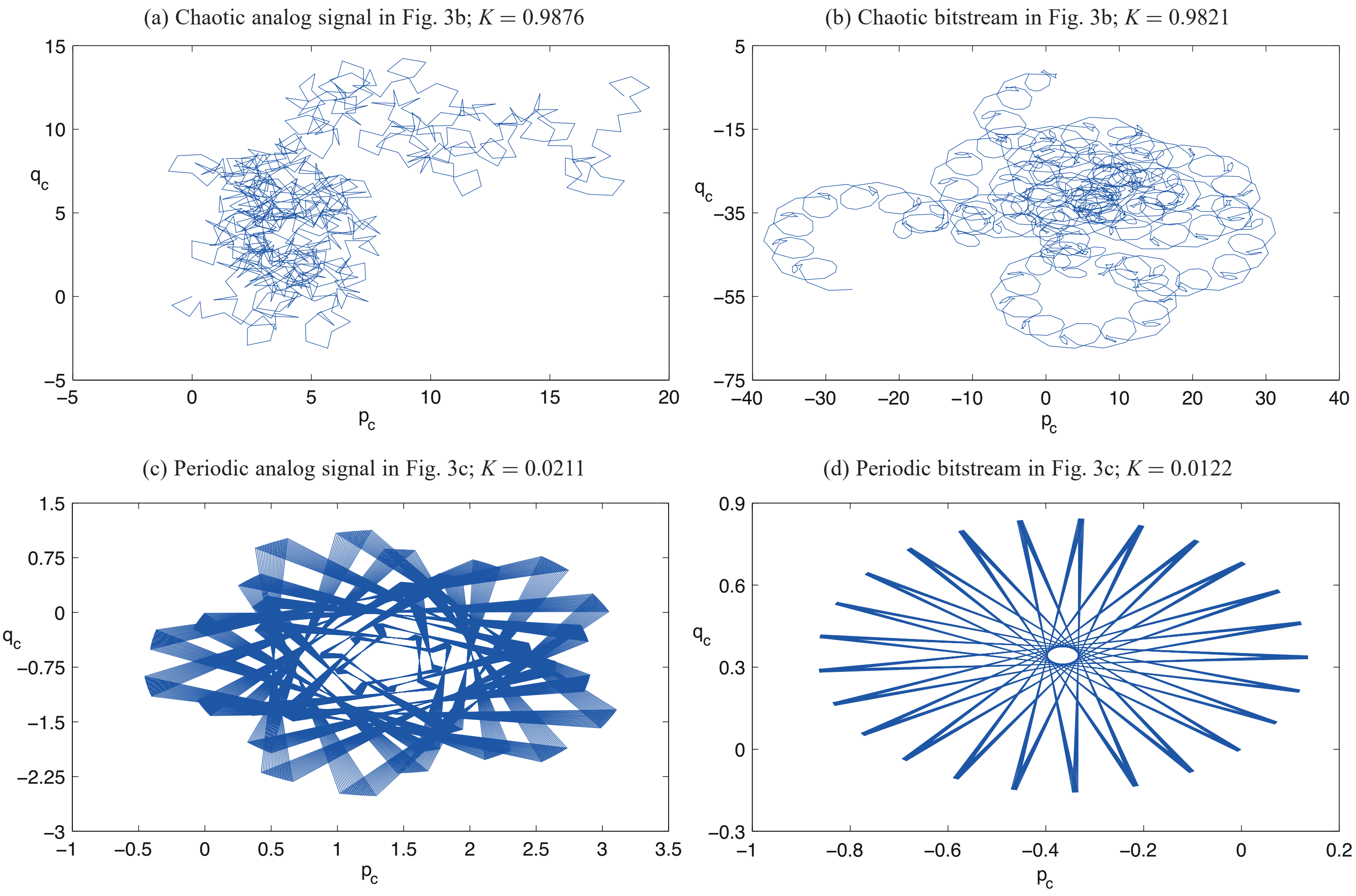
Fig. 3. Hardware Trojan in the Lindberg generator as a result of changing the f_{AC} value from 7 kHz (figure b) to 3 kHz (figure c)

sistor R_1 (see Fig. 3a). The results of such modifications are presented in Figs. 3–5. Fig. 3b shows the chaotic continuous and binary signals obtained in Lindberg's circuits for $f_{AC} = 7$ kHz. Changing that frequency to, for example, $f_{AC} = 3$ kHz results in periodic both continuous and binary signals, as shown in Fig. 3c. A HT may also be in the form of changing the values of parameters. One such possibility in Lindberg's circuit is to change the value of R_1. An example of R_1 modification and its impact on the V_{C_2} signal is shown in Section 4. Similar HTs can be considered in other chaotic generators, for example those based on the Chua circuit [13–15].

2.2. Characteristics of HTs. One can identify two main parts in any implementation of HTs: a payload and a trigger mechanism. The HT's structure and complexity depend on the system in which the HT is applied and the task the HT is supposed to perform. The payload is present in each HT. The trigger mechanism is optional and its presence depends on whether the payload is activated on a constant or temporary basis. The problems of HTs in the context of chaotic-based cryptography have not been yet properly addressed, much less solved. Recent developments in embedded systems and PCBs require detailed and comprehensive analysis of the security threats of HTs [7–9].

HTs do not require complex or costly implementations. Trojans can be implemented based on rather simple circuits. As shown in Fig. 2, HTs can be classified according to:

- Stage of insertion. The basis of HTs can be developed at the stage of designing of an embedded system. Such HT may be a result of modifications done by unauthorized persons having access or being familiar with the main functional characteristics of the embedded system. Another possibility is to implement HT at the stage of prototyping or assembling of embedded systems.
- Level of abstraction. HTs in Lingberg's generators in the forms of elements' changes (resistor and source frequency values) are at the high level of abstraction. Similar HTs can be implemented in FPGA modules, which can also serve as triggering sites.
- Triggering mechanism. HTs can be activated as permanent or temporary. Examples of permanent trojans are the changes of resistors' values at the assembly line. Such HTs require hardware integration along a production line. Temporary modifications are also possible through the digitally controlled resistors from the FPGA level. Such HTs can be activated and deactivated in time. In bit generators such an approach results in periodically obtained chaotic


(a) Chaotic analog signal in Fig. 3b; $K = 0.9876$

(b) Chaotic bitstream in Fig. 3b; $K = 0.9821$

(c) Periodic analog signal in Fig. 3c; $K = 0.0211$

(d) Periodic bitstream in Fig. 3c; $K = 0.0122$

Fig. 4. Plots $q_c - p_c$ for chaotic and periodic signals in Lindberg's generator shown in Figs.3b and 3c

and non-chaotic dynamics. Digitally activated HTs allow an intruder to precisely set up a moment of activation and duration of the modification. HT activation can also be done when special additional conditions are satisfied, for example, through an external event. Most of the current embedded systems in the forms of PCBs are not equipped with security systems that can detect, identify and prevent unauthorized modifications of the systems' layouts [9, 11].

- Results of HT's activity. HTs in analog Lindberg's generators change the generated output signal, lowering the security of the cryptosystem in which the generator is used. This is true even in a case of temporarily activated HTs.
- Site of implementation. HT in a system with the Lindberg's generator is located in the core circuit shown in Fig. 3a and in the FPGA systems responsible for the mechanism of activation.

3. Chaos detecting 0–1 test

A detection of unauthorized modification of the generators can be done by testing the generated bitstream with the statistical test suites for random and pseudorandom number generators for cryptographic applications. Such tests were developed by the US National Institute of Standards and Technology (NIST) [1, 2]. The statistical tests require a bitstream with the minimum of one million bits. In practice the testing bitstream is usually even longer. Additionally, the testing bitstream must go through a preprocessing correction (e.g., von Neumann correction), which eliminates intervals with the domination of the same bits. Thus, performing the whole test is a time-consuming process. In this paper, we propose a new method to protect the security of the chaotic generators. This approach requires bitstreams of much shorter length (only about 5 thousand bits).

The 0–1 test (see [16–25] for details) is a relatively new tool used to test the presence of chaos in analog and digital sequences when a mathematical model (system of equations) is not available. The result of the test has two forms: a single real number K, and a two-dimensional graph with translation variables p_c and q_c [16, 17]. For a chaotic sequence the number K should be close to 1. Regular (non-chaotic) sequences result in numbers K closer to 0. The values of K can be computed by using two different methods: regression or correlation.

For a sequence $\{N_k\}$, $k = 0, \ldots, \overline{N} - 1$, the variables p_c and q_c are computed by the following expressions for a randomly chosen real number $c \in (0, \pi)$

$$p_c(n) = \sum_{j=0}^{n} N_j cos[(j+1)c], \quad q_c(n) = \sum_{j=0}^{n} N_j sin[(j+1)c] \quad (1)$$

with $n = 0, \ldots, \overline{N} - 1$. Then, the mean square displacement $M_c(n)$, $n = 0, 1, \ldots, n_{cut}$, of $p_c(n)$ and $q_c(n)$ is computed with the recommended integer value $n_{cut} \approx (\overline{N} - 1)/10$

$$\begin{aligned} M_c(n) = \lim_{\overline{N}\to\infty} \frac{1}{\overline{N}-1} \sum_{j=1}^{\overline{N}-1} [p_c(j+n) - p_c(j)]^2 + \\ + [q_c(j+n) - q_c(j)]^2. \end{aligned} \quad (2)$$

Next, if the regression method is applied, then the asymptotic growth rate K_c of the mean square displacement is computed as follows

$$K_c = \lim_{n\to\infty} \frac{log M_c(n)}{log\, n}. \quad (3)$$

On the other hand, if the correlation method is applied, then two vectors $\xi = (0, 1, 2, \ldots, n_{cut})$ and $\Delta = (M_c(0), M_c(1), M_c(2), \ldots, M_c(n_{cut}))$ are created. The correlation coefficient K_c is obtained as follows

$$K_c = corr(\xi, \Delta) \equiv \frac{cov(\xi, \Delta)}{\sqrt{var(\xi) var(\Delta)}} \quad (4)$$

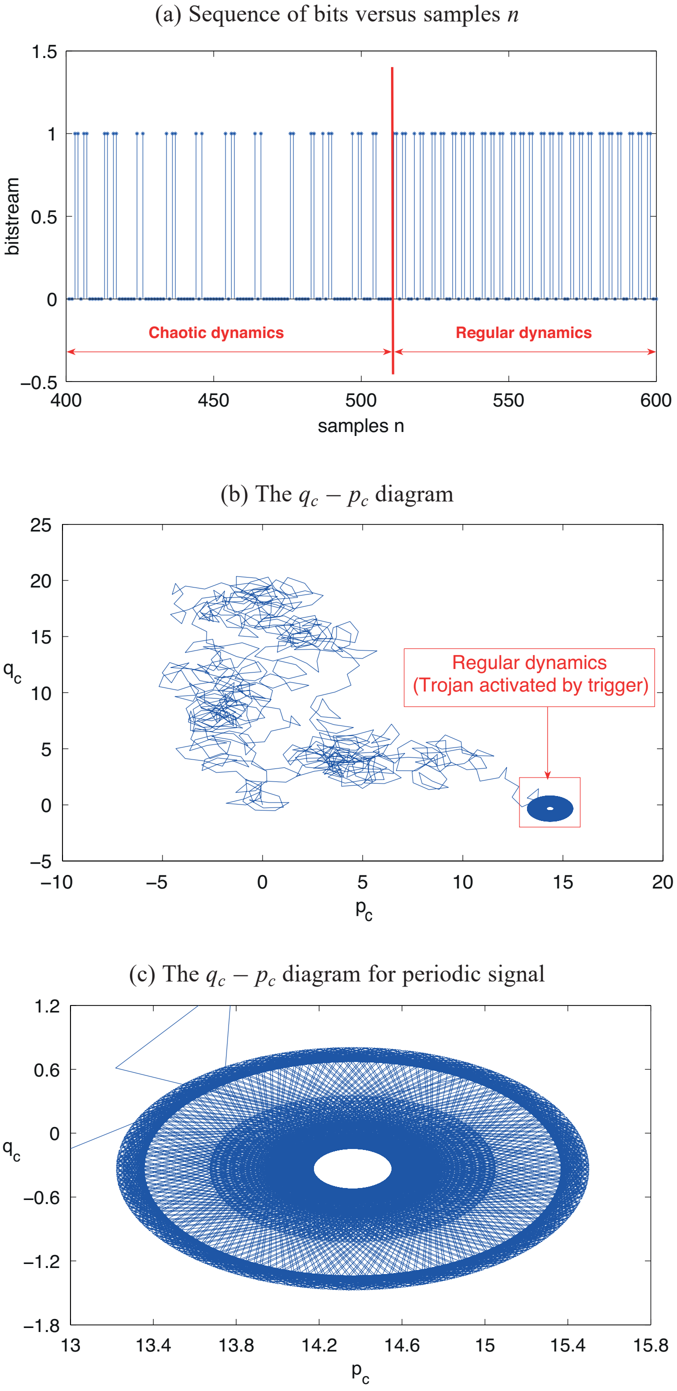
where the *cov* and *var* stand for covariance and variance, respectively [16].

In both methods the above steps are repeated for N_c values of c chosen randomly in the interval $(0, \pi)$. Again, [16] recommends $N_c = 100$. Finally, the median of the N_c values of K_c is the final number K. The $K \approx 1$ indicates a chaotic sequence, while $K \approx 0$ indicates regular (non-chaotic) dynamics. For more details about the 0/1 test, its properties and reliability in the continuous and discrete cases one can see [18–25].

4. Application of the 0–1 test to detect HTs

4.1. HTs in Lindberg's generator. The above 0–1 test can be applied for HT detection. Analyzing an output bitstream can be used to evaluate the security level of the tested system. This can be done at two levels: numerical (through the value of K) and visual (through the $q_c - p_c$ plot).

We have tested this approach for the generators presented in Figs. 3a and also further in Fig. 7 (logistic equation based generator). The results of the asymptotic growth rate K and $q_c - p_c$ plots for Lindberg's generator (modifications of the resistor R_1 and the source frequency f_{AC}) are shown in Figs. 4–6. Figures 4a, b, c, d show the $q_c - p_c$ plots corresponding to the time responses in Figs. 3b and 3c, respectively. The change of f_{AC} from 7 kHz to 3 kHz results in the change of K from $K = 0.9876$ (chaotic analog signal in Fig. 3b (top)) and $K = 0.9821$ (chaotic bitstream in Fig. 3b (bottom)) to $K = 0.0211$ (periodic analog signal in Fig. 3c (top)) and $K = 0.0122$ (periodic bitstream in Fig. 3c (bottom)). The corresponding visual test resuts in the form of the $q_c - p_c$ plots are shown in Fig. 4. The visual test itself seems to be a valuable tool allowing for a real-time on-screen monitoring of the bitstream sequence. The Brownian type of the $q_c - p_c$ plot suggests a chaotic bitstream, while


(a) Sequence of bits versus samples n

(b) The $q_c - p_c$ diagram

(c) The $q_c - p_c$ diagram for periodic signal

Fig. 5. Simulation of a Trojan in Lindberg's generator resulting from changing the value of R_1 from 1 $k\Omega$ to 15 $k\Omega$

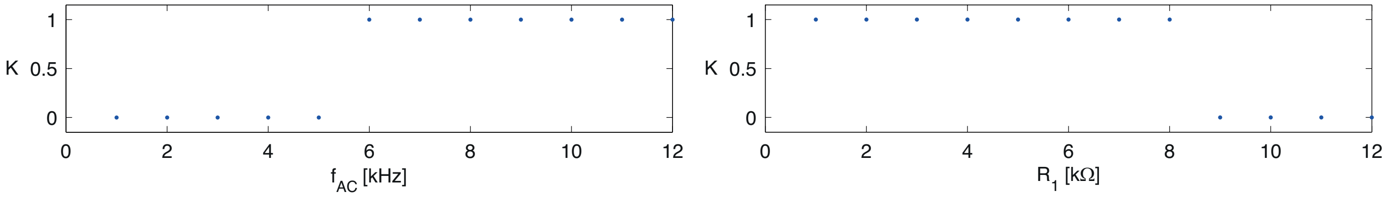


Fig. 6. Results of using the 0–1 test of chaos for the source frequency f_{AC} between 1 and 12 kHz (left) and resistor R1 values between 1 and 12 kΩ (right). HTs are triggered at f_{AC} between 5 and 6 kHz and at the resistance change between 7 and 8 kΩ

a symmetric, regular $q_c - p_c$ plot indicates a periodic bitstream. Another type of HT is illustrated in Fig. 5. The HT is in the form of R_1 modification (see Fig. 3a). Fig. 5a illustrates a change of the nature of signal V_{C2} (see Fig. 3a), when R_1 is suddenly changed from 1 kΩ to 15 kΩ at the discrete sample $n = 510$ (see the horizonatal axis in Fig. 5a). The corresponding $q_c - p_c$ plot is shown in Fig. 5b. The HT activated at $n = 510$ results in a ceasing of the Brownian motion and for $n > 510$ the $q_c - p_c$ plot is restricted to a very small area inside the rectangle in the lower right corner in Fig. 5b. Figure 5c shows the $q_c - p_c$ diagram for the periodic signal for the discrete samples $n > 510$.

Finally, Fig. 6 shows the K values from the 0–1 test as a function of frequency f_{AC} (left) and resistance R_1 (right).

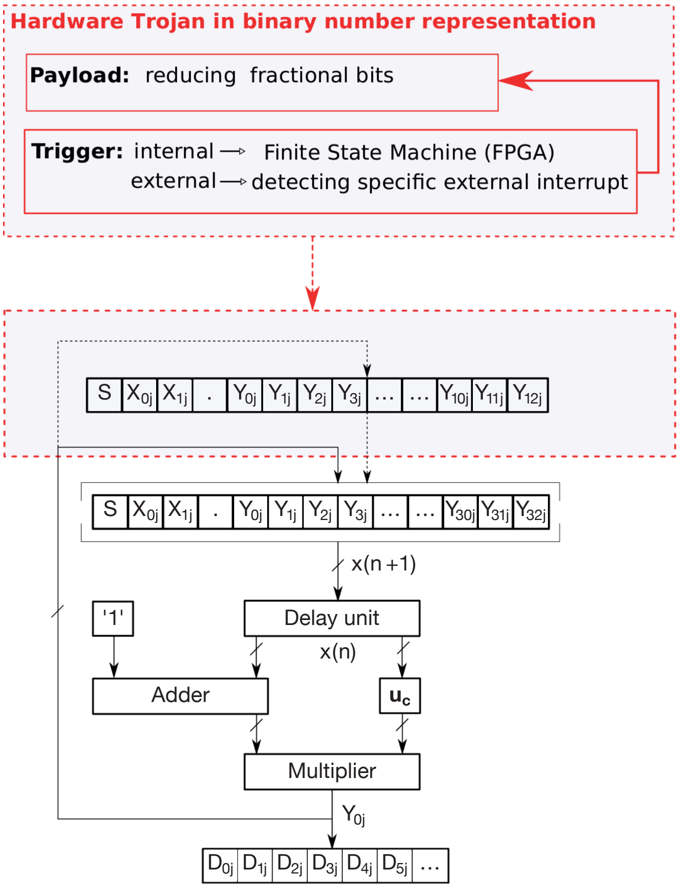

Hardware Trojan in binary number representation
Payload: reducing fractional bits
Trigger: internal → Finite State Machine (FPGA)
external → detecting specific external interrupt
S X0j X1j . Y0j Y1j Y2j Y3j … … Y10j Y11j Y12j
S X0j X1j . Y0j Y1j Y2j Y3j … … Y30j Y31j Y32j
x(n+1)
'1'
Delay unit
x(n)
Adder
uc
Multiplier
Y0j
D0j D1j D2j D3j D4j D5j …

Fig. 7. Diagram of a chaotic generator based on logistic equation $x(n+1) = u_c x(n)[1 - x(n)]$, $u_c \in R$

4.2. HTs in the logistic equation based generator. We have also applied the 0–1 test to evaluate the digital bitstream generator based on logistic equation. Such a generator is sensitive not only to the hardware modification, but the response depends also on the precision of the digital representation of numbers [3, 5]. Nowadays, digital circuits are often implemented in reprogrammable systems (e.g. FPGA), which are chosen by designers as a convenient, inexpensive and fast solution. Unfortunately, embedded systems with these positive features are susceptible to HTs. The possibility of reconfigurability allows modifications of the structure with HTs being activated at predetermined times. Also, system modifications can be used to decrease the accuracy of digital representation of numbers. Such a decrease in accuracy can lead to periodic bitstreams. In order to check effectiveness of the 0–1 test in such a case, the logistic equation based generator shown in Fig. 7 was implemented in FGPA.

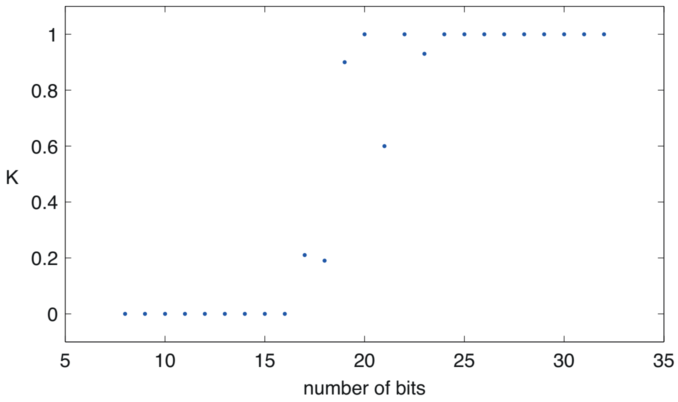


Fig. 8. Values of K as a function of the number of bits in fixed point representation

This next HT is based on the modifications of the accuracy of number representation. The achieved values of K as a function of the number of bits used for digital representations are shown in Fig. 8. The HT in this experiment is in the form of the reduced length of sequence $\{Y\}$ in Fig. 7. The bits are taken from the 7th position in the $\{Y\}$ sequence. The $\{Y\}$ sequence of length 33 results in a chaotic bitstream with K close to 1. This corresponds to the bitstream signal in Fig. 9a for the discrete samples $985 \le n \le 3505$. When the length of sequence $\{Y\}$ is shortened to 13, a periodic bitstream is obtained with K close

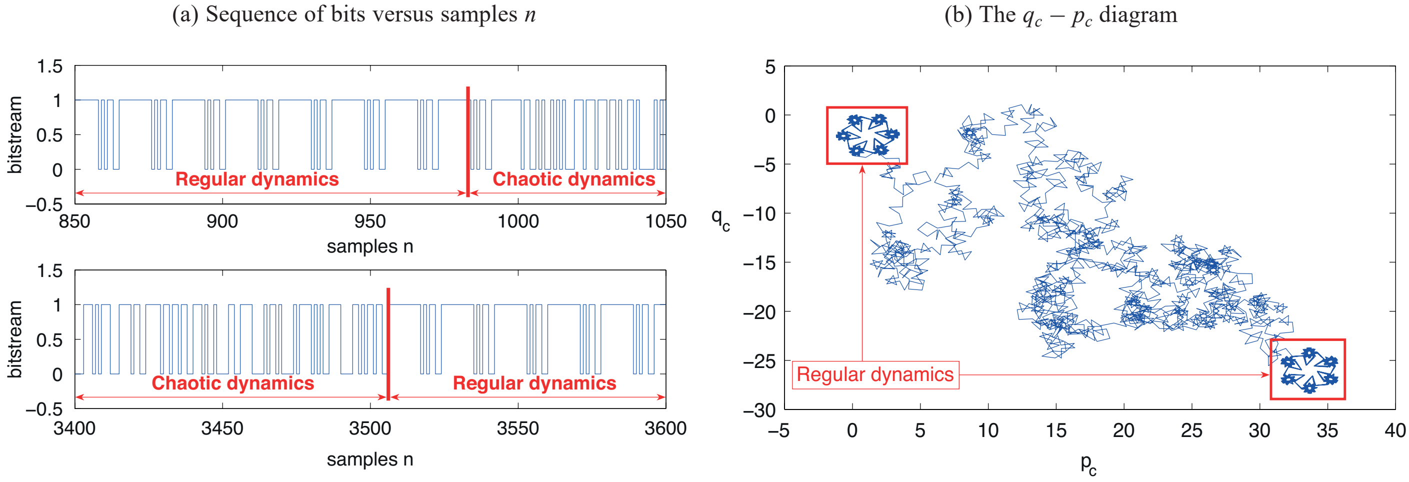


Fig. 9. Hardware Trojan simulation for the logistic-based chaotic generator with a stream of bits taken from the 7th position in sequence $\{Y\}$. The regular sequence is obtained for the sequence $\{Y\}$ of length 13, while the chaotic sequence is a result of using sequence $\{Y\}$ of length 33

to 0. This corresponds to the bistream signal in Fig. 9a for the discrete samples $n < 985$ and also for $n > 3505$. These two periodic bitstream sequences (for $n < 985$ and $n > 3505$) are the sequences in the two small rectangular boxes in Fig. 9b. The chaotic sequence for $985 \leq n \leq 3505$ results in a Brownian-like motion in Fig. 9b. In the case of the logistic equation based chaotic generator the HT structure can be integrated in the FPGA. Similarly, as in the case of Lindberg's circuit based generator, HT may be set to stay active only in selected time intervals. To activate HT one may use a finite state machine, which can sequentially increase and decrease a number of bits needed in a fixed number precision representation. Decreasing the number of bits may result (as shown above in Fig. 9) in a periodic bitstream.

The future of the 0–1 test seems to be promising. The above presented results of our tests confirm the applicability of the 0–1 test to monitor the generated bitstreams. The simplicity of the test, which is based on monitoring of the output sequence in the form of $q_c - p_c$ plot in real-time and online, seems to be attractive for future practical implementations. The $q_c - p_c$ plots could be backed-up by quick calculations of the K parameter when Trojan activity is suspected. Thus, the 0–1 test for chaos offers a two-level security mechanism (tool) to confirm or exclude our suspicion of a HT attact.

5. Conclusions

In the paper we discussed an important and current problem of security of hardware modules used in cryptography systems. We focused our attention on the HT detection in chaotic bitstream generators. This type of circuits is an important part of cryptography systems. Analog and digital generators were analyzed in the context of possible unauthorized modifications of various types. The 0–1 test for chaos was applied to detect the unwanted behavior of the generators. The analysis, performed for Lindberg's circuit and discrete logistic equation based generator, shows the usefulness of the proposed method. Our results were achieved based on bitstreams of rather short lengths of 5000. This fact seems to be an advantage of the 0–1 test when compared to the other known tests for random bit generators, which typically require sequences of 1 million bits [1, 2]. Additionally, the 0–1 test does not require any pre-processing of the bitstream, such as, for example the von Neumann correction used in other tests [1]. The proposed method is useful for both analog and digital implementation of the generators.

Acknowledgements. This paper was recommended for publication by the Scientific Commitee of the National Electronics Conference, Darlowko Wschodnie, Poland, 5–10 June, 2016.

References

[1] M.E. Yalcin, J.A.K. Suykens, and K. Vandewalle, "True random bit generation from a double-scroll attractor", *IEEE Trans. Circuits and Systems I: Regular Papers* 50, 1395–1404 (2004).
[2] L. Kocarev and L. Shiguo (Eds.), *Chaos-Based Cryptography. Theory, Algorithms and Applications*, Springer 2011.
[3] K.J. Persohn and R.J Povinelli, "Analyzing logistic map pseudorandom number generators for periodicity inducted by finite precision floatingpoint representation", *Chaos, Solitions and Fractals*, 45, 23–244 (2012).
[4] P. Sniatala, J. Pierzchlewski, A. Handkiewicz, and B. Nowakowski, "CPLD based development board for mixed signal chip testing", *14th Int. Conf. Mixed Design of Integrated Circuits and Systems*, 21–24 June, 2007, pp. 492–495, Gdynia, Poland.
[5] M. Melosik and W. Marszalek, "A hybrid chaos-based pseudorandom bit generator in VHDL-AMS", *IEEE 57th Int. Midwest Symp. Circuits and Syst. (MWSCAS)*, 2014, pp. 435–438, College Station, TX, USA.
[6] S. Ghosh, A. Basak, and S. Bhunia, "How secure are printed circuit boards against trojan attacks?", *IEEE Design and Test* 32, 7–16 (2014).
[7] S. Bhunia et al., "Hardware Trojan attacks: threat analysis and countermeasures", *Proc. IEEE*, 102, 1229–1247 (2014).

[8] Y. Jin and M. Yiorgos, "Hardware Trojans in wireless cryptographic ICs", *IEEE Design and Test of Computers* 27, 26–35 (2010).

[9] X. Zhang and M. Tehranipoor, "Case study: detecting hardware Trojans in third-party digital IP cores", *IEEE Int. Symp. Hardware-Oriented Security and Trust (HOST)*, doi:10.1109/HST.2011.5954998, 2011.

[10] T. Reece and W. H. Robinson, "Detection of hardware trojans in third-party intellectual property using untrusted modules", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems* 35, 357–366 (2015).

[11] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*, Springer, Berlin, 2011.

[12] E. Lindberg, K. Murali, and A. Tamasevicius, "The smallest transistor-based nonautonomous chaotic circuit", *IEEE Trans. Circuits and Systems-II: Express Briefs* 52, 661–664 (2005).

[13] W. Marszalek and Z.W. Trzaska, "Mixed-mode oscillations in a modified Chua's circuit", *Circuits Syst. Signal Process*. 29, 1075–1087 (2010).

[14] W. Marszalek and Z.W. Trzaska, "Memristive circuits with steady-state mixed-mode oscillations", *Electr. Lett*. 50 (18), 1275–1277 (2014).

[15] H. Podhaisky and W. Marszalek, "Bifurcations and synchronization of singularly perturbed oscillators: an application case study", *Nonl. Dynamics* 69 (3), 949–959 (2012).

[16] G.A. Gottwald and I. Melbourne, "A new test for chaos in deterministic systems", *Proc. Royal Soc. London* 460, 603–611 (2003).

[17] G.A. Gottwald and I. Melbourne, "On the implementation of the 0–1 test for chaos", *SIAM J. Appl. Dyn. Syst*., 8, 129–145 (2009).

[18] M. Melosik and W. Marszałek "On the 0/1 test for chaos in continous systems", *Bull. Pol. Ac.: Tech* 64 (3), 521–528 (2016).

[19] D. Bernardini and G. Litak, "An overview of 0–1 test for chaos", *J. Braz. Soc. Mech. Sci. Eng*., DOI :10.1007/s40430–015–0453- y (2015).

[20] G. Litak, A. Syta, and M. Wiercigroch, "Identification of chaos in a cutting process by the 0–1 test", *Chaos, Solitons and Fractals*, 40, 2095–2101 (2009).

[21] G. Litak, A. Syta, M. Budhraja, and I.M. Saha, "Detection of the chaotic behaviour of a bouncing ball by the 0–1 test", *Chaos, Solitons and Fractals* 42, 1511–1517 (2009).

[22] G. Litak, D. Bernardini, A. Syta, G. Rega, and A. Rysak, "Analysis of chaotic non-isothermal solutions of thermomechanical shape memory oscillators", *Eur. Phys. J. Spec. Top*. 222, 1637–1647 (2013).

[23] L. Zachilas and I. Psarianos, "Examining the chaotic behavior in dynamical systems by means of the 0–1 test", *J. Appl. Math*., 2012, 681296 (2012).

[24] I. Falconer, G.A. Gottwald, I. Melbourne and K. Wormnes, "Application of the 0–1 test for chaos to experimental data", *SIAM J. Appl. Dyn. Syst*. 6, 395–402 (2007).

[25] J. Fouda, B. Bodo, S. Sabat, and J. Effa, "A modified 0–1 test for chaos detection in oversampled time series observations", *Int. J. Bifurc. Chaos* 24, 1450063 (2014).