Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 190 (2021) 370–376

Procedia Computer Science

www.elsevier.com/locate/procedia

2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society

Encryption of pseudorandom number generator logic circuits

Mikhail Ivanov[a], Iliya Chugunkov[a], Bogdana Kliuchnikova[a], Evgenii Salikov[a]

[a]*National Research Nuclear University MEPhI, Kashirskoe highway 31, Moscow, 115409, Russia*

Abstract

The paper presents obfuscation methods for logic circuits of pseudorandom number generators (PRNG) on shift registers with linear and nonlinear feedback, which are based on the PRNG additional logic elements for protection against reverse engineering. The encryption of the PRNG logic circuit changes its design in the way that the device works correctly only if the signals at the additional key inputs of the PRNG take correct values. Even with a small bit capacity of PRNG, a huge number of PRNG implementations with a different number of states and different properties can be provided. The concept of (M + 1)-sequence generator is introduced. The possibility of transforming generators of (M – 1)-sequence and (M – 3)-sequence into generators of (M + 1)-sequence is demonstrated.

Keywords: logic encryption; linear feedback shift register; nonlinear feedback shift register; (M + 1)-sequence; pseudorandom number generator (PRNG).

1. Introduction

In recent years, malicious hardware has become a serious threat to the security of computer systems. Outsourcing in the production of integrated circuits (IC) creates problems associated with the introduction hardware bugs introduction, IC counterfeiting, piracy and unauthorized overproduction.

The most effective opportunities to prevent all of the above threats are provided by logic encryption and design obfuscation technologies [1-3], their implementation can be based on the use of pseudorandom number generators (PRNG) with a non-standard switching diagram, for example, generators of (M – p + 1)-sequences and generators of (M – 2n + 1)-sequences, where p is prime and n is natural [4].

The paper discusses the features of using the logic encryption technology for obfuscation of the logic circuit of binary PRNGs on shift registers with linear and nonlinear feedbacks (LFSR and NLFSR respectively).

10.1016/j.procs.2021.06.046

2. Shift registers with linear and nonlinear feedback

PRNGs on LFSR and NLFSR have been used for a long time to solve various problems of protecting information from accidental and intentional destructive influences [5-11]. The following areas of their use can be highlighted:

- Built-in self-test for VLSI circuits
- Probabilistic testing of large-scale sequential circuits
- Construction the CRC-codes
- Building scramblers and descramblers
- Construction of PRNG with non-standard switching diagrams
- Construction of stream and block ciphers

Fig. 1 shows examples of the most common PRNG schemes on LFSR and NLFSR, where the 4-bit case is considered, i.e. N = 4.

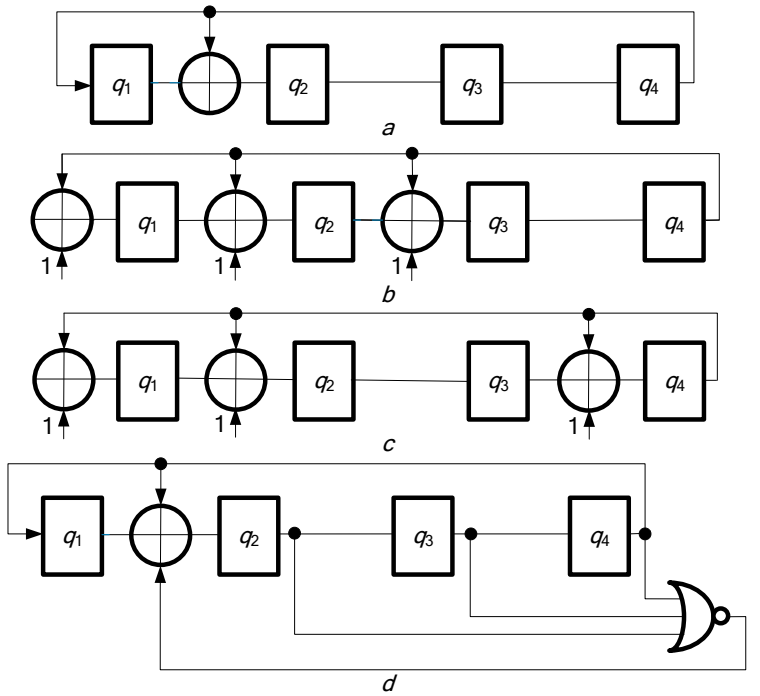

Fig. 1. 4-bit PRNG with M = 2N – 1: (a) M-sequence generator; (b) generator of (M - 1)-sequence; (c) generator of (M - 3)-sequence; (d) generator of (M + 1)-sequence.

Fig. 1 (a) shows a diagram of M-sequence generator, which constructed according to the Galois scheme corresponding to the characteristic polynomial $\varphi(x) = x^4 + x + 1$, which is primitive over the field GF(2). The state transition graph of the device has the form 15-1, in other words, it consists of two cycles, the first is 15 long and it includes all nonzero states of the generator, the second is 1 node long and it includes the "all zeros" state, which passes into itself. The generator operation equations have the form

$$\begin{aligned} q_1^* &= q_4, \\ q_2^* &= q_1 + q_4, \\ q_3^* &= q_2, \\ q_4^* &= q_3, \end{aligned} \tag{1}$$

where addition is performed modulo two, where q_i and q_i^* are the content of the i-th bit of the generator at time t and $t + 1$, $i = 1, 2, 3, 4$.

Fig. 1 (b) shows the generator of the (M - 1)-sequence, which built on the basis of the Galois generator corresponding to the characteristic polynomial $\varphi(x) = (x + 1)(x^3 + x^2 + 1) = x^4 + x^2 + x + 1$, where $\lambda_1(x) = x^3 + x^2 + 1$ is primitive over GF(2). The generator state transition graph has the form 14-2 and for correctly configured device the modulo-2 convolution of the generator memory elements changes their values to the opposite in each cycle. The equations of operation have the form

$$\begin{aligned} q_1^* &= q_4 + 1, \\ q_2^* &= q_1 + q_4 + 1, \\ q_3^* &= q_2 + q_4 + 1, \\ q_4^* &= q_3, \end{aligned} \tag{2}$$

Fig. 1 (c) shows the generator of the (M – 3)-sequence, which built on the basis of the Galois generator corresponding to the characteristic polynomial $\varphi(x) = (x + 1)^2(x^2 + x + 1) = x^4 + x^3 + x + 1$, where $\lambda_2(x) = x^2 + x + 1$ is primitive over GF(2). The generator state transition graph has the form 12-4 and for correctly configured device the modulo-2 convolution of the generator memory elements changes their values to the opposite in each cycle. The equations of operation have the form

$$\begin{aligned} q_1^* &= q_4 + 1, \\ q_2^* &= q_1 + q_4 + 1, \\ q_3^* &= q_2, \\ q_4^* &= q_3 + q_4 + 1, \end{aligned} \tag{3}$$

The concepts of (M – 1)-sequence and (M – 3)-sequence were introduced in [8].

Fig. 1 (d) shows a diagram of an (M + 1)-sequence generator, which built on the basis of the device shown in Fig. 1, (a). The generator state transition graph consists of one cycle is 16 long. When the device is in state 1000, a single signal at the output of the NOR element provides switching of the device to the previously disabled state 0000. If the signal at the output of the NOR element is still equal to one, as a result in the next cycle the device returns back to the main cycle passing to the 0100 state. The generator operation equations are

$$\begin{aligned} q_1^* &= q_4, \\ q_2^* &= q_1 + q_4 + z, \\ q_3^* &= q_2, \\ q_4^* &= q_3 + q_4 + 1, \end{aligned} \tag{4}$$

where z is the signal at the output of the NOR element, and $z = 1$, if $q_2q_3q_4$=000, or $z = 0$ for other cases.

3. Logic encryption

Consider the main idea of logic encryption technology. Encryption of the digital device logic circuit (Fig. 2 (a)) makes it possible to use additional logic elements in the IC structure to hide its original functionality. In other words, this is an attempt to make it as difficult as possible for unauthorized persons to understand the logic of the protected scheme. The encryption of the logic circuit, and in fact its obfuscation, changes the design of the IC in the way that it works correctly only if the signals at the additional key inputs of the device take on the correct values.

The obfuscation scheme assumes the use of an additional key transformation scheme, implemented on the basis of a memory block with protection against interference. This memory block is installed or activated at the final stage of creating an IC before selling it to the end user.

In implementing the logic encryption using a key conversion scheme, this scheme can be implemented based on a substitution box, classical cryptographic primitive or one-way function. In the first case a fixed key is used, in the second case a sequence of key information used.

4. PRNG logic encryption

Fig. 2 (b) shows an example of the implementation of logic encryption, when the device protected from reverse engineering is a PRNG. Fig. 3 shows the simplest example of encryption of a 4-bit generator, built using the Galois scheme. Depending on the values at the key inputs k_1k_0 the device implements one of the four state transition graphs are shown in Table 1.

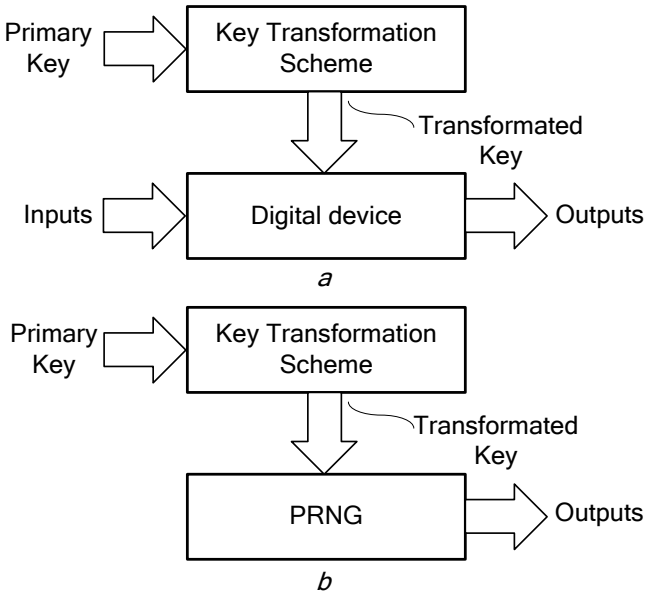

Fig. 2. Digital device logic encryption: (a) general scheme; (b) implementation for PRNG.

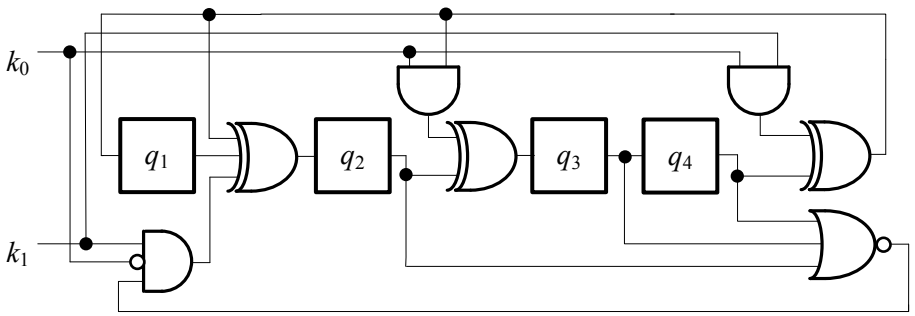

Fig. 3. 4-bit PRNG with M = 2N – 1: (a) M-sequence generator; (b) generator of (M – 1)-sequence; (c) generator of (M – 3)-sequence; (d) generator of (M + 1)-sequence.

Table 1. Dependence of the state transition graph of the device on the value of the key k_1k_0.

Mode	Key	State transition graph
0	0 0	2 cycles are 15 and 1 long
1	0 1	4 cycles are 7,7,1 and 1 long
2	1 0	1 cycle is 16 long
3	1 1	2 cycles are 14 and 2 long

5. Modeling an encrypted PRNG scheme

Fig. 4 shows more complex scheme of an encrypted 4-bit PRNG with 9 key inputs, which means it is capable of performing 29 different functions, including 32 variants of M-sequence generators, 16 variants of (M - 1)-sequences generators, 8 variants of (M - 3)-sequences generators. Construction of the device makes it possible to identify more than 10 unusual modes of operation of the circuit, one of which is shown in Fig. 5. The scheme is interesting in that the generator of the (M + 1)-sequence changes operation mode twice when at $z = 1$ and it changes the rule of the 2-modulo convolution for ... 0 1 0 1 0 1 ... the state of the generations.

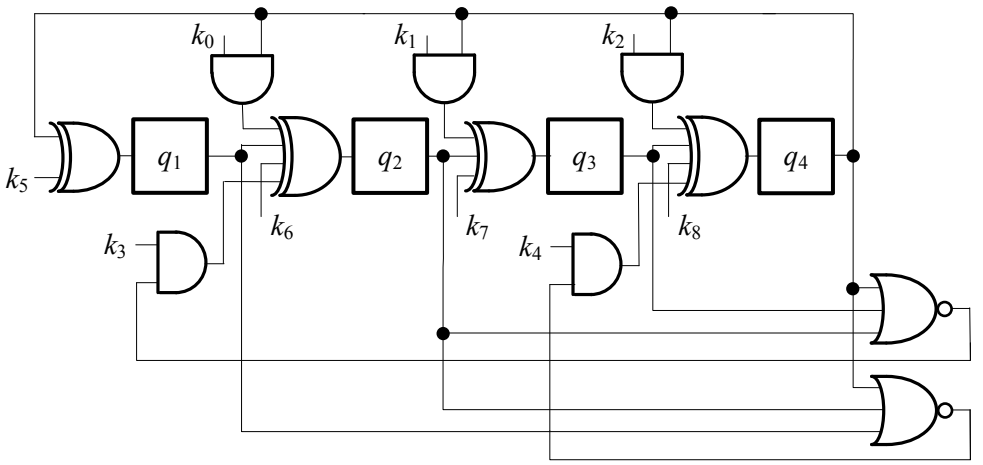

Fig. 4. Encrypted binary 4-bit PRNG circuit.

The device shown in Fig. 4 is built on the basis of the Galois generator corresponding to the characteristic polynomial $\varphi(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + 1$ over GF(2). The equations of operation of the basic generator have the form

$$\begin{aligned} q_1^* &= q_4 + c_1, \\ q_2^* &= q_1 + a_1q_4 + c_2, \\ q_3^* &= q_2 + a_2q_4 + c_3, \\ q_4^* &= q_3 + a_3q_4 + c_4, \end{aligned} \tag{5}$$

where $a_i \in \{0, 1\}$ are the coefficients of the polynomial $\varphi(x)$, $c_i \in \{0, 1\}$ are the control inputs of the device. In the general case, for an arbitrary generator capacity equal to N, the equations take the form

$$\begin{aligned} q_1^* &= q_N + c_1, \\ q_j^* &= q_{j-1} + a_{j-1}q_N + c_j, \quad j = 2, 3, \ldots, N \end{aligned} \tag{6}$$

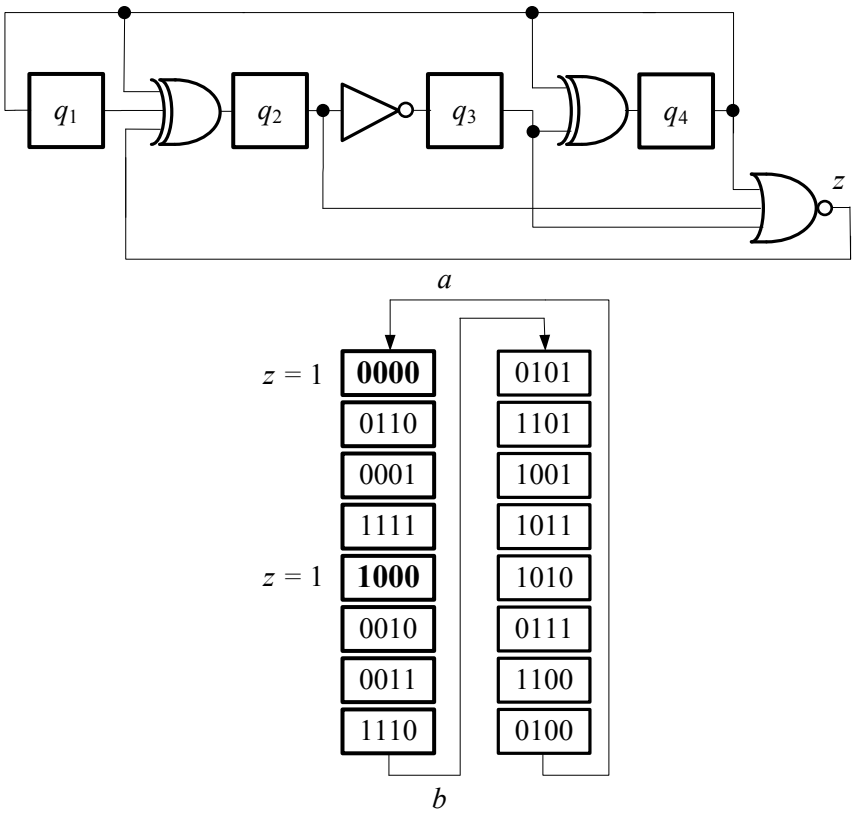

Fig. 5. The logic scheme of the binary 4-bit generator
for $k_8\,k_7\,k_6\,k_5\,k_4\,k_3\,k_2\,k_1\,k_0 = 0\,1\,0\,0\,0\,1\,1\,0\,1$:
(a) circuit scheme of the device; (b) state transition graph.

6. Conclusion

Methods of protection against reverse engineering of logical circuits of generators of pseudo-random numbers on shift registers with linear and nonlinear feedback are considered. It is shown that even with a small bit capacity of generators, it is possible to provide a huge number of PRNG implementations with a different number of states and different properties. With increasing the capacity of generators, the number of primitive polynomials increases, and hence the number of possible variants of generator schemes for M-sequence, (M – 1)- sequence and (M – 3)-sequence types. In the general case, when the capacity of the generator is equal to N, these numbers are respectively equal to

$$\begin{aligned}
\sigma_M(N) &= \frac{2^N \psi\left(2^N - 1\right)}{N}, \\
\sigma_{M-1}(N) &= \frac{2^{N-1} \psi\left(2^{N-1} - 1\right)}{N-1}, \\
\sigma_{M-3}(N) &= \frac{2^{N-1} \psi\left(2^{N-2} - 1\right)}{N},
\end{aligned} \tag{7}$$

where $\psi(\cdot)$ – Euler's number.

Thus, the obfuscating of the device circuit, in addition to solving the problems with malicious hardware, IC counterfeit, piracy and unauthorized overproduction, makes it possible to implement a mechanism of hidden (highly protected) device functions, for example, to protect a technical solution from dual use.

Further development can be associated with increasing the number of generators of (M + 1)-sequences, the addition of modes for generating sequences with a pre-period, as well as the possibility of constructing generators operating in the GF(2^n).

Acknowledgment

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (state assignment project No. 0723-2020-0036).

References

[1] Mukhopadhyay, Debdeep and Rajat Subhra Chakraborty (2014) "Hardware Security: Design, Threats, and Safe-guards." *CRC Press*.

[2] Tehranipoor, Mohammad and Cliff Wang (2012) "Introduction to Hardware Security and Trust." *Springer*.

[3] Becker, Georg, Fyrbiak, Marc and Christian Kison (2017) "Hardware Obfuscation." *Springer*.

[4] Ivanov, Mikhail, Kliuchnikova, Bogdana, Salikov, Evgenii and Andrei Starikovskii (2019) "New Class of Non-Binary Pseudorandom Number Generators." *Proceeding of Intelligent Technologies in Robotics*, 255–262.

[5] Dubrova, Elena (2011) "A Scalable Method for Constructing Galois NLFSRs with Period 2n –1 using Cross-Join Pairs" *Cryptology ePrint Archive*, **632**.

[6] Dubrova, Elena (2011) "A Method for Generating Full Cycles by a Composition of NLFSRs." *Cryptology ePrint Archive*, **632**.

[7] Dubrova, Elena, Teslenko, Maria and Hannu Tenhunen (2008) "On analysis and synthesis of (n, k)-non-linear feedback shift registers", *Design and Test in Europe,* 133-137.

[8] Pesoshin, Aleksei and Kuznecov Mikhail (2007) "Generatory psevdosluchajnyh i sluchajnyh chisel na registrah sdviga", *Kazan, Izd-vo Kazanskogo gos. tehn. un-ta.*

[9] Pesoshin, Aleksei and Kuznecov Mikhail (2012) "Generatori ravnoveroyatnostnih psevdosluchainih posledovatelnostei na registrah sdviga. Izvestiya visshih uchebnih zavedenii. Povoljskii region.", *Tehnicheskie nauki* **1**: 21-28.

[10] Pesoshin, Aleksei and Kuznecov Mikhail (2013) "Generatori sluchainih i psevdosluchainih posledovatelnostei na cifrovih el-ementah zaderjki", *Kazan, Izd-vo Kazanskogo gos. tehn. un-ta*.

[11] Pesoshin, Aleksei, Mikhail, Kuznecov and Vasilya Shirshova (2016) "Generators of the equiprobable pseudorandom nonmaximal-length sequencess based on linear-feedback shift registers", *Automation and Remote Control* **77 (9):** 1622-1632.