In modern software, crypto-algorithms are widely used for both data encryption tasks, and authentication and integrity checks. There are well-known and proven crypto-algorithms. Their cryptoresistance is either mathematically proven or based on the need to solve a mathematically complex problem (factorization, discrete logarithm, etc.). On the other hand, in the computer world, information constantly appears about errors or «holes» in a particular program (including one that uses crypto-algorithms) or that it was broken (cracked). This creates distrust both in specific programs and in the possibility to protect something in general by cryptographic methods not only from special services, but also from ordinary hackers. A promising direction of research in this field is the implementation of a hybrid random number generator with two types of entropy sources in cryptosystems.

The method and means of implementing a hybrid random number generator with two types of entropy sources: external – based on Zener diode noise and internal – based on the uncertainty state of the transistor-transistor logic structure are presented. One of the options for the practical implementation of a random number generator is presented, where two sources are used as a source of entropy: an external source – Zener diode noise and an internal source – the undefined state of the transistor-transistor logic structure. The functional diagram of the proposed random number generator with two types of entropy sources is given. The MATLAB/Simulink model of the proposed random number generator is built, the results of the statistical analysis of the generated random sequences by the NIST SP 800-22 test package are given

Keywords: crypto-resistance, crypto-algorithm, random numbers, pseudo-random numbers, uncertainty zone, entropy sources, cipher stability

UDC 004.94
DOI: 10.15587/1729-4061.2022.265774

DEVELOPMENT OF A HARDWARE CRYPTOSYSTEM BASED ON A RANDOM NUMBER GENERATOR WITH TWO TYPES OF ENTROPY SOURCES

Serhii Yevseiev
Corresponding author
Doctor of Technical Sciences, Professor, Head of Department*
E-mail: Serhii.Yevseiev@gmail.com
Khazail Rzayev
PhD, Associate Professor
Department of Computer Technology and Cybersecurity
Azerbaijan Technical University
G. Javid ave., 25, Baku, Azerbaijan, AZ 1073
Oleksandr Laptiev
Doctor of Technical Sciences, Associate Professor, Senior Researcher
Department of Cyber Security and Information Protection
Taras Shevchenko National University of Kyiv
Volodymyrska str., 60, Kyiv, Ukraine, 01033
Ruslan Hasanov
Doctor of Science, Associate Professor
Department of Radioelectrinics
National Aviation Academy
Mardakan ave., 30, Baku, Azerbaijan, AZ 1044
Oleksandr Milov
Doctor of Technical Sciences, Professor*
Bahar Asgarova
Assistant Professor**
Jale Camalova
Assistant**
Serhii Pohasii
PhD, Associate Professor*
*Department of Cyber Security
National Technical University «Kharkiv Polytechnic Institute»
Kyrpychova str., 2, Kharkiv, Ukraine, 61002
**Department of Computer Engineering
Azerbaijan State Oil and Industry University
Azadlyg ave., 20, Baku, Azerbaijan, AZ1010

Received date 08.07.2022
Accepted date 30.09.2022
Published date 27.10.2022

How to Cite: *Yevseiev, S., Rzaev, K., Laptiev, O., Hasanov, R., Milov, O., Asgarova, B., Camalova, Ja., Pohasii, S. (2022). Development of a hardware cryptosystem based on a random number generator with two types of entropy sources. Eastern-European Journal of Enterprise Technologies, 5 (9 (119)), 6–16. doi: https://doi.org/10.15587/1729-4061.2022.265774*

1. Introduction

Information is the main resource, and it must be protected. One of the areas of hardware and software information protection is cryptography. The level of effectiveness of such protection is estimated by cryptographic strength. The level of cryptographic strength is the ability of a cryptographic algorithm to resist decryption [1]. A strong algorithm is an

algorithm that can not be decrypted for a long time, so long that by the time the information is received, the encrypted data will not be relevant.

There are several encryption methods: symmetric, with one encryption key, and asymmetric, with two keys. The level of cryptographic strength for each type of encryption is different, so for symmetric encryption, this parameter is equal to the key length. The task of ensuring the required cryptographic strength of encryption algorithms is becoming increasingly important due to the development of information technology. As is known, with the help of encryption, the following security states should be ensured: confidentiality, integrity, and identifiability of transmitted information. One of the most productive means of solving this problem is the use of efficient encryption methods. To select the appropriate cryptoalgorithm, it is necessary to master the mathematical apparatus underlying the algorithm, as well as to analyze the possibility of a particular encryption method to withstand modern cryptanalytic attacks. Next, it is important to choose criteria for evaluating and analyzing the cryptographic strength of encryption algorithms. For example, security margin, key expansion speed, protection against runtime attacks, the ability to quickly expand the key, etc.

However, with the advent of high-performance computing technology, the security of cryptographic algorithms is questioned, so the process of improving cryptographic protection systems is always relevant.

The analysis was carried out and the requirements for the operation of cryptosystem hardware were formed. One of the options for the practical implementation of the random number generator is given, and a quantitative assessment of the strength of symmetric encryption algorithms is also given. This is one of the solutions to the urgent scientific problem of improving cryptosystems.

2. Literature review and problem statement

The fundamental rule of cryptanalysis is that the strength of a cipher (cryptosystem) should be determined only by the secrecy of the key. The entire encryption algorithm, except for the value of the secret key, is known to the adversary's cryptanalyst. This is due to the fact that a cryptosystem that implements a family of cryptographic transformations is usually considered as an open system. This approach reflects a very important principle of information security technology: the security of the system should not depend on the secrecy of something that cannot be quickly changed in the event of a leak of classified information. Typically, a cryptosystem is a combination of hardware and software that can be changed only with considerable time and money. Therefore, when improving cryptoprotection, attention should be paid to both software and hardware.

So, in [2] the hardware for encrypting streaming information by methods of indirect steganography is presented. Illustrations of cryptosystem operation algorithms are presented. The hardware that provides cryptographic protection uses software that is based on a classical computer built according to the von Neumann architecture. The weak link is the system architecture itself, since the software only functions within the framework of predefined architectural solutions. Based on this, the availability of such architectural solutions can become a significant problem.

The papers [3, 4] present the disadvantages and advantages of cryptographic protection algorithms. An important advantage of asymmetric algorithms over symmetric ones is that there is no need to pre-transmit the secret key. The main disadvantage is the computational complexity and, consequently, high resource costs compared to symmetric algorithms. Therefore, in practice, asymmetric cryptosystems are used to transmit a secret key, and further information is exchanged using symmetric cryptosystems.

In [5], an analysis of cryptographic protection tools was carried out, which showed that most cryptographic data protection tools are implemented in the form of specialized physical devices. These devices are built into the communication line and encrypt all information transmitted over it. The predominance of hardware encryption over software is due to several reasons. Higher speed. Cryptographic algorithms consist of a huge number of complex operations performed on bits of plaintext. Modern mainframe computers are unsuitable to perform these operations efficiently. Specialized equipment can do them much faster. However, the encryption algorithms and methods themselves are not considered.

In [6], the advantages of hardware cryptosystems are presented. It is easier to physically protect the equipment from outside penetration. A program running on a personal computer is practically defenseless. Armed with a debugger, an attacker can make subtle changes to it to lower the strength of the cryptographic algorithm being used without anyone noticing. As for the equipment, it is usually placed in special containers that make it impossible to change the scheme of its operation. The chip is covered with a special chemical composition, and as a result, any attempt to overcome the protective layer of the chip leads to the self-destruction of its internal logical structure. Although electromagnetic radiation can sometimes be a good source of information about what is happening inside the microcircuit, it is easy to get rid of this radiation by shielding the microcircuit. Similarly, it is possible to shield a computer, but this is much more difficult to do than a miniature microcircuit. However, only physical and chemical means of protection are considered in the work, the cryptosystem software is not considered.

The paper [7] lists the advantages of hardware cryptosystems. Options for an unconditional advantage are presented, for example, encryption equipment is easier to install. Very often, encryption is required where additional computer hardware is completely unnecessary. Phones, fax machines, and modems are much cheaper to equip with hardware encryption devices than to build microcomputers with the appropriate software into them.

The paper [8] shows the potential market of cryptosystems. The modern market for information encryption hardware offers potential buyers three types of such tools. These are self-sufficient encryption modules (they independently do all the work with keys), encryption blocks in communication channels and encryption expansion boards for installation in personal computers. The disadvantage of the devices of the first and second types is that they are highly specialized. At the same timeч, the analysis of encryption methods itself is not given, there are also no encryption algorithms.

The papers [9, 10] consider the issue of software-terminal solutions and cryptographic protection, which are the undisputed leaders in the information security rating. Various types of cryptographic algorithms are considered, such as keyless, one-key and two-key. However, the combination of these two leaders of cryptographic protection in one information system remains rather problematic. For almost all the algorithms presented in these works, there are links to archives with their implementation in C, C++ or Assembler. However, it is not

possible to use them due to the imperfection of the program code itself. Methods of real provision of workable software algorithms are not presented. Therefore, hardware implementation of cryptosystems based on these algorithms is not possible.

The work [11] considers hardware information protection in the form of a built-in computer board. They are special devices that are installed in a computer in order to protect the information processed on it. These modules allow you to encrypt data that is written to a computer drive or transferred to its ports and drives for subsequent recording to external media. The encryption mode can be transparent or pre-encrypted. The board contains a pseudo-random number generator for generating keys and encryption nodes. The devices are highly efficient at encrypting information, but do not have built-in protection against electromagnetic interference. However, the setter itself or the pseudo-random number generator is not described and other options for cryptoprotection are not considered.

The works [12, 13] describe that encryption systems involve the use of both hardware and software packages. Hardware is high-speed, easy-to-install, easily physically tamper-resistant devices. They are built into the communication line and encrypt all information transmitted over it. There are three types of these devices – self-sufficient encryption modules (independently perform all the work with keys), encryption blocks in communication channels and encryption expansion cards for installation in a PC. The software packages are easy to copy, easy to use, easy to modify according to specific needs. The advantages of hardware cryptosystems are described, but the principle of encryption itself is not considered.

In [14–16], the issue of choosing a cryptosystem is considered. Often in practice, a specialist who needs to protect any information from unauthorized reading is faced with the question of which means to give preference to: software or hardware. There are several factors that determine the choice of protection means. These include:

– the value of protected information for third parties;

– the size of protected information;

– the ability to read encrypted information on other devices;

– damage resulting from the loss of information due to cryptosystem failures;

– the price of the cryptosystem;

– the need to hide the very fact of finding valuable information;

– the possibility of using counterfeit software (crack programs for pirated copies, key generators, key emulators).

To select a means of protection, one should compare the strengths and weaknesses of software and hardware cryptosystems based on the above criteria. Advantages of hardware encoders:

– simple and reliable user identification;

– no need to limit the encrypted space, you can encrypt entire disks, not individual files and directories;

– the probability of a device failure is lower than a software failure or damage to the key for the program operation, which is stored in a separate file;

– the inability often even to read the encrypted data for decryption on other devices;

– the impossibility to decrypt data in case of theft of the carrier itself;

– the possibility of using complex, and therefore, time-consuming encryption algorithms with high cryptographic strength;

– a cryptosystem is a set of hardware and software that can be changed only with significant time and money.

Therefore, when improving cryptoprotection, attention should be paid to hardware. Despite a significant number of publications on the development and improvement of hardware cryptographic protection systems, the problem has not been fully resolved.

3. The aim and objectives of the study

The aim of the study is to develop a hardware cryptosystem based on a random number generator with two types of entropy sources. This approach provides protection against threats of hacking and/or bypassing the code of the software implementation of the cryptosystem.

To achieve the aim, the following objectives were set:

– to analyze and form requirements for the operation of hardware cryptosystems such as: pseudo-random number generators (PRNG); true random number generators (TRNG) and hybrid random number generators (HRNG);

– to develop a model of a hardware cryptosystem – an RN generator with two types of entropy sources;

– to simulate the proposed hardware system and get a real picture of the sequence of random numbers at the output of the PRNG/HRNG for different input parameters and effects on it.

4. Materials and methods

In the scientific interpretation, the basic requirements for the cryptographic strength of the system can be expressed as follows:

– the cryptosystem transformation mechanism should not require confidentiality; it must be assumed that it is known to the enemy;

– the stability of the cryptosystem should be determined only by the secret key.

The formation and transmission of a secret key determine the confidentiality of the cryptosystem as a whole. To form a secret key in asymmetric, symmetric and hybrid encryption systems, a wide range of both software and hardware tools are used, called random number generators (RNG).

There are three groups of RNG: pseudo-random number generators (PRNG), true random number generators (TRNG) and hybrid random number generators (HRNG).

The operation of PRNG is based on the use of mathematical models in which a sequence of pseudo-random numbers (PRN) is formed from some initial value called the «initialization vector» or «seed». At the same time, the main requirements for PRNG are: good statistical characteristics, high speed, the ability to recreate the received and predetermine the following sequences. Consider the types of PRNG.

In a simple PRNG, the result is calculated as a function of the current time, data entered by the user, etc. It is used in the formation of static keys and has low cryptographic strength.

Software PRNG are usually developed in high-level programming languages. They have a fairly long period and high speed. In addition, FPSC of this type are easy to modify. Despite these advantages, a powerful computer is required, which limits its use in small-sized applications.

Hardware PRNG combine the advantages of the previously listed types. A distinctive feature of this type is the possibility to use them autonomously (without computers). In addition, depending on the algorithm used and the element base (microcontrollers, FPGA, etc.), they can provide a high

speed of generation, a long period, and allow modifying the software within certain limits. Most modern cryptographic PRNG are built exactly according to this principle. It should be noted that very little is known about hardware PRNG that are successfully used in solving certain cryptographic problems, since most of them are designed for military purposes or patented and kept secret. Despite this, in many countries, work is underway to create various hardware PRNG, the results of which are in the open press. Most of the proposed methods are based on the operation of a linear feedback shift register (LFSR). By itself, LFSR does not have high resistance and can be easily hacked using the Berlekamp-Massey algorithm [10]. Therefore, they serve as building blocks for more complex algorithms. PRNG based on LFSR are widely used, for example, in the A5/1 and A5/2 algorithms of the GSM mobile communication standard, E0 of the Bluetooth wireless data transmission standard, etc.

Separately, the group of PRNG based on elliptic curves should be noted. Several PRNG algorithms based on the properties of elliptic curves have been proposed [11]. As a rule, the implementation of such PRNG is based on the use of already known algorithms for a group of points on an elliptic curve. For example, a linear congruential generator (LCG) over an elliptic curve, elliptic PRNG algorithms based on linear feedback shift registers, etc.

5. Results of the development of a hardware cryptosystem based on a random number generator with two types of entropy sources

5. 1. Formation of requirements for the operation of hardware cryptosystems

All of the listed types of PRNG have two main drawbacks: the periodicity of the generated sequences and their correspondence to a certain mathematical model. In the first case, the creation of a complete database of generated sequences by the analyst leads to the determination of its period and, as a result, to the hacking of the system. In the second case, the determination of several sequence values by the analyst and the application of mathematical methods of cryptanalysis lead to the same result.

To eliminate the shortcomings of PRNG, various sources of entropy are used.

Such devices are combined into a common group called TRNG.

In general, the requirements for TRNG used for cryptographic purposes can be formulated as follows:

– uniform distribution of true random numbers (TRN) in a given interval;

– statistical independence of each TRN from the previous one;

– the impossibility of calculating the next TRN based on previous values;

– high speed of TRN generation;

– the possibility of using TRNG in applications of small size and low power consumption.

There are two approaches to this problem: indirect generation of TRN, using TRN tables, and direct generation of TRN – measurement, as well as processing real physical processes.

TRN tables are pre-formed large arrays of high-quality random numbers stored in electronic media. They have high statistical characteristics and reproduction property. However, the fact that the TRN tables are prepared in advance (as a result of measurements and calculations) and are very large makes their use in high-speed real-time data transmission systems practically impossible.

To overcome these shortcomings, it is recommended to use a TRNG with directly obtaining TRN, i.e. with an entropy source. As the latter, sensors of real physical processes are used:

– thermal, Zener, avalanche, atmospheric, etc. noises;

– optical, electrical, optoelectric, mechanical, etc. chaotic processes;

– quantum processes (radioactive decay, photoelectric effect, phase fluctuations of optical rays), etc.

The main advantage of using noise as entropy sources is that the final device is small and has low power consumption.

The disadvantage of noise-based TRNG compared to PRNG is the low rate of sequence formation due to the relatively low-frequency nature of the physical processes. Also noteworthy is the fact that not all TRN have qualities that can pass statistical randomness tests. To eliminate the first of these shortcomings, the HRNG is used. This approach combines the positive characteristics (speed and randomness) of both classes of RN generators. In this case, the random sequence generated by the TRNG acts as an initialization vector for the PRNG. In other words, random numbers are formed by a pseudo-random algorithm between the moments of TRN generation. The elimination of the second disadvantage depends only on the methods and means of implementing the TRNG.

5. 2. Development of a hardware cryptosystem model

One of the options for implementing the HRNG is considered, where two sources are used as an entropy source: an external source, the noise of a Zener diode, and an internal source, an undefined state of the transistor-transistor logic (TTL) structure.

It is necessary to consider the nature of the noise that occurs in the Zener diode in more detail. When using thermal noise as a source of entropy, the resulting sequences have a strong correlation. In this case, it is impossible to speak about the generation of TRN. The noise that occurs when the Zener diode operates in the tunneling or avalanche breakdown mode has a completely different nature and can be used as an entropy source. Both tunneling and avalanche breakdowns can occur during the reverse connection. The forms of current-voltage characteristic (CVC) for both cases are shown in Fig. 1. It is possible to determine the type of breakdown only experimentally. In addition, there may be a case when both of these breakdowns occur (Fig. 1 – gray zone). Thus, if a tunneling breakdown occurs, then an increase in temperature (t) leads to a shift of the CVC to the right, and in the case of an avalanche breakdown, to the left.

Tunneling breakdown is observed when the electron energy is less than the height of the potential barrier. If the reverse current flowing through the *p-n* junction is small enough, the jumps of carriers through the barrier individually will manifest themselves as voltage jumps and have an ideal random noise nature [9].

In the case of an avalanche breakdown, under the action of a voltage of a back-applied electric field, the carriers receive a sufficiently high kinetic energy for impact ionization. As a result, the number of charge carriers involved in impact ionization increases like an avalanche. The reason for the noise in this case is a decrease in the local breakdown voltage due to the relatively higher concentration of carriers in the vicinity of crystal defects than in other regions of the junction. Such

local zones are called «microplasmas». The current flowing through each microplasma has the form of a short-term pulse and can appear and disappear at random times. The described phenomenon occurs before the transition to the continuous avalanche breakdown mode, which corresponds to the initial section of the avalanche breakdown on the CVC.

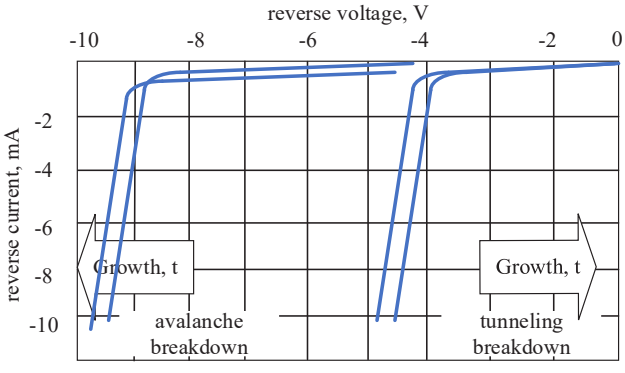

Fig. 1. Current-voltage characteristic of the Zener diode in the case of avalanche and tunneling breakdown

As can be seen from Fig. 1, with the correct choice of the operating point, both tunneling and avalanche breakdowns can be used as a source of entropy. In this case, the operating point must be chosen in the zone preceding the steady-state electrical breakdown, which corresponds to an unsteady electrical breakdown.

The second source of entropy is the uncertainty zone of the transistor-transistor logic (TTL) structure. The essence of this approach is that any signal falling (by amplitude) into the specified zone can be randomly interpreted by the TTL structure as a logical «0» or a logical «1». For the TTL structure, this zone is limited to 0.4–2.4 V.

The functional diagram of the proposed HRNG with entropy sources based on a Zener diode operating in the mode of transient electrical breakdown and an undefined state of the TTL structure can be described as follows (Fig. 2).

The noise generated by the Zener diode (ZD) is amplified by the operational amplifier (OA) and sampled in time by means of the sampler (SM). The sampling period (T_d) is determined by the periodic pulse generator (PG) sequence. As a result of each sampling operation, a pulse with a duration T_d and a random amplitude from 0 to the maximum output voltage of the OA is formed. It should be noted that the T_d value is selected experimentally and depends on the rate of change in the noise amplitude. The duration (τ_d) of sampling pulses can take on an arbitrarily small value and satisfy the condition $\tau_d \leq T_d$. The entry of pulses into the uncertainty zone of the TTL structure is modeled by a solver (SL).

In fact, the SL is a threshold device in which the threshold value is chosen randomly. As a result of the joint work of SM and SL, a sequence of random digits is formed – 0 and 1, which is then converted into a parallel code by means of a serial-to-parallel converter (SPC) and stored in an N-bit buffer memory (BM). The number of sampling operations made by SM for one period of TRNG operation is chosen equal to the capacity of the BM.

After the BM is filled, the generated N-bit word is transferred to the LFSR input and serves as an initialization vector for it. The operation of the entire system is synchronized by the synchronizer (SN). The synchronization process is implemented as follows: the PG is started by the front of the clock pulse and stopped by the cutoff, the filling of the LFSR cells is enabled by the cutoff of the clock pulse and is prohibited by the front. The operation of the SPC and the BM is synchronized by a single pulse generator (SPG). At the output of the SPG, a pulse with duration $\tau_{ui}=\tau_d$ and period $T_{ui}=T_s$ is formed. For the entire period of TRNG operation, only one such impulse is formed, hence the name. The front of a single pulse coincides with the front of the clock pulse from the SN. It should be noted that the speed of the SPC is determined precisely by the parameter τ_{ui}, i.e., the smaller it is, the higher the conversion rate. For optimal operation of the device, it is necessary to select the duration (τ_s) and the period of the clock pulses (τ_s) based on the ratio:

$$\begin{cases} \tau_s = N \cdot T_d, \\ T_S = \Delta \cdot T_{LFSR}, \end{cases} \quad (1)$$

where Δ is the number of values generated by the LFSR after initialization (cell filling), T_{LFSR} is the LFSR clock period.

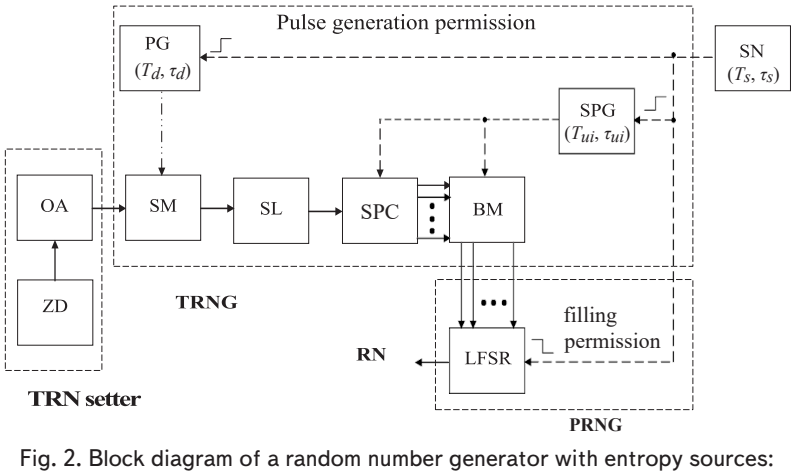

Fig. 2. Block diagram of a random number generator with entropy sources: ZD – Zener diode; OA – operational amplifier; SM – sampler; SL – solver; SPC – serial-to-parallel converter; BM – buffer memory; PG – pulse generator; SPG – single pulse generator; LFSR – linear feedback shift register; SN – synchronizer

5. 3. Modeling of the proposed hardware cryptosystem

The implementation of the presented functional diagram in the Matlab/Simulink system is considered below. The TRN setter is implemented (Fig. 3) on two blocks: Random Number and Gain. In view of computer simulation, it is impossible to talk about the true randomness of the numbers generated by the selected generator, but this approach is quite acceptable for checking the system's performance and forming RN in a relatively short period of time. If desired, tables of true random sequences can be used. It is recommended to reduce the Variance parameter of the generator to 0.001 to simulate the real noise level.

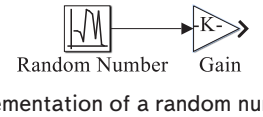

Fig. 3. Implementation of a random number source setter in Simulink

Blocks SM, PG and SL are implemented together (Fig. 4). The operation of SM and PG is simulated by one Zero-Order Hold block. The Sample time parameter of the block corresponds to the value of T_d. In this case, we can assume that $\tau_d = T_d$.

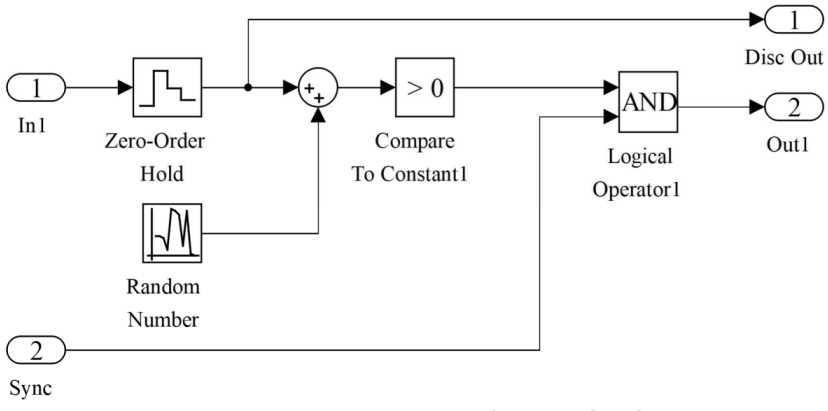

Fig. 4. Implementation of a sampler, SL, and PG in Simulink

The main requirement in the construction of SL is to provide a random interpretation of the input signal level. One option is to sum (Sum block) the pulses obtained as a result of sampling with randomly generated pulses (Random Number block). Due to the presence of pulses of negative and positive polarities in both sequences, some of the pulses will be compensated randomly. Taking into account the peculiarities of the TTL structure, all pulses of negative polarity should be cut off (Compare To Constant block). Logical gate and Logical operator block allow synchronizing the operation of SM, PG and SL.

The implementation of the SPG block is shown in Fig. 5. The task of the SPG is to form a single pulse of a given duration at the moment a clock signal from the SN arrives at its input, regardless of the duration of the latter. In this case, the duration of a single pulse is set by the same-name parameters of the Delay and Delay1 blocks. These parameters are equal to 1. The period of a single pulse is T_S.

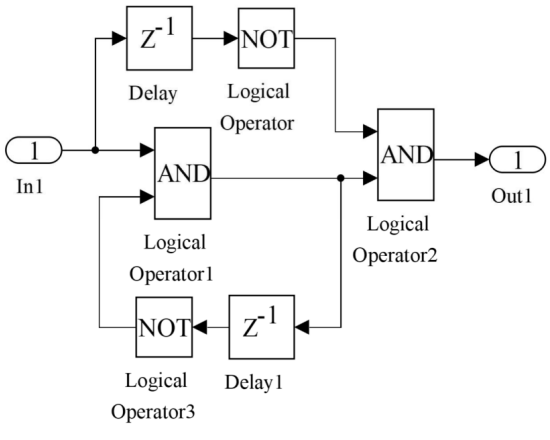

Fig. 5. Implementation of a single pulse generator in Simulink

The SPC and BM blocks are also implemented together (Fig. 6). The circuit is synchronized by pulses from the SPG arriving at the Sync input. Each of the seven Delay-Delay6 blocks is necessary to delay a single pulse for a time interval equal to mT_d, where $m = 1, 2, ..., N-1$ is the ordinal number of the delay line. The sequence of N-digit TRN is fed to the input (In1) of the SPG. Each bit (except for the first digit) of the sequence and the single pulse delayed by mT_d are fed to the inputs of the corresponding AND logic gates (Logical operator 1 – Logical operator 7 blocks). The results are loaded into S-R Flip-Flop – S-R Flip-Flop7 RS triggers. The essence of this approach lies in the fact that a single pulse, delayed by a specified time interval (mT_d), scans and loads the position value in the TRN sequence into the corresponding trigger. The BM is cleared by a single pulse delayed in Delay7.

The implementation of the LFSR block is shown in Fig. 7.

In this case, an 8-bit LFSR was used according to the Fibonacci configuration. The main parts of any LFSR are a shift register and a feedback circuit, the implementation of which will be discussed below. The parallel TRN from the BM output goes to the inputs In1 – In8 of the LFSR, and the clock signal goes to the Sync (Reset seed) input.

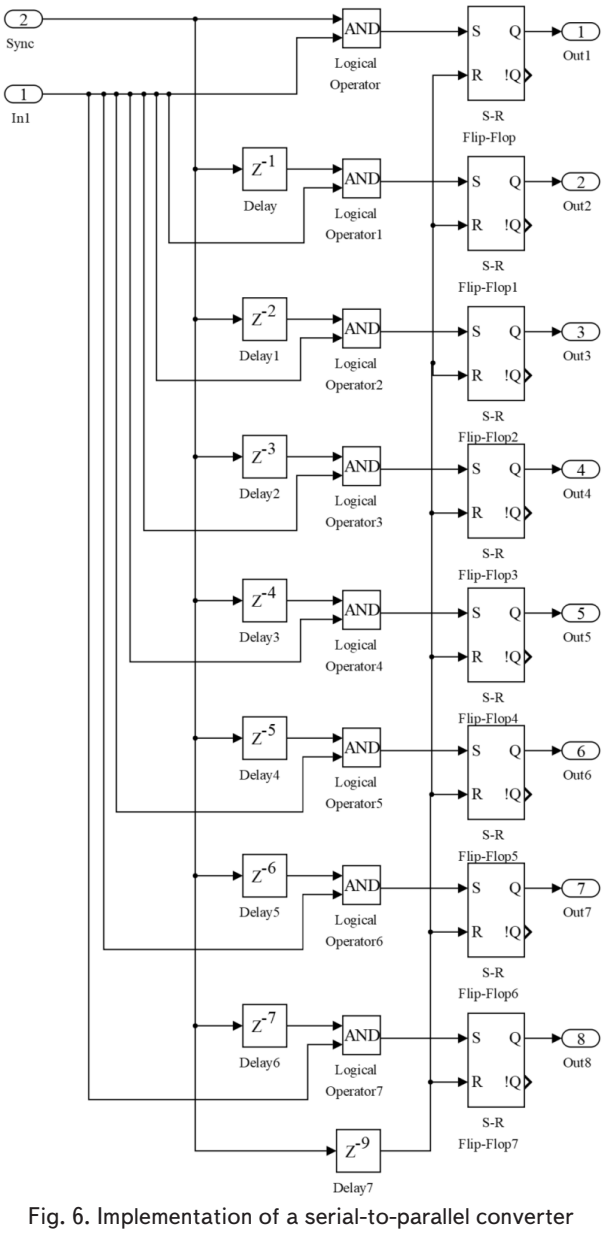

Fig. 6. Implementation of a serial-to-parallel converter and buffer memory in Simulink

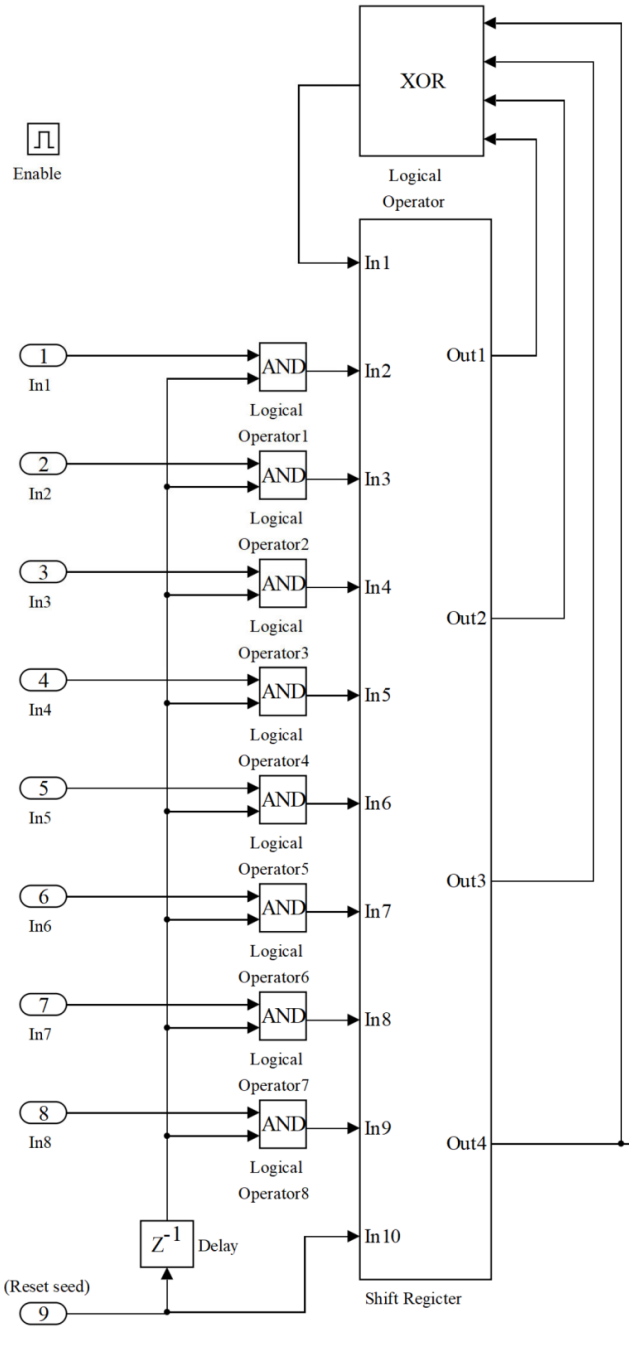

Fig. 7. Implementation of a linear feedback shift register in Simulink

The implementation of the shift register based on delay lines is shown in Fig. 8. Each of the Delay-Delay7 blocks must be set to zero at the cutoff of the clock pulse, i.e., the Algorithm/External reset parameter must be switched to Falling. The delay time of each block is the same for all blocks and is equal to the clock period of the LFSR – T_{LFSR}. This parameter determines the speed of the shift register and LFSR as a whole. The T_{LFSR} value should be chosen according to expression (1). It should also be noted that, according to the LFSR condition, $\Delta \leq 2^N - 1$.

At the cutoff of the clock pulse at the synchronization input (Sync (Reset seed) output), all cells of the shift register are filled with new values received from the TRN setter. Out1 is the output of the entire system, where a sequence of RN is obtained.

Taking into account all the above blocks and sub-blocks, as well as measuring tools, the implementation of the proposed RN generator with two types of entropy sources in Simulink is as follows (Fig. 9).

It should be noted that the PRNG shown in Fig. 1, consisting of only a single LFSR is just one of many options. Despite the simplicity and high speed, this approach does not have high cryptographic strength. Studies have shown that RN sequences generated using a single LFSR fail the following NIST SP 800-22 tests:

1. Maurer's «Universal Statistical» Test.
2. Approximate Entropy Test.
3. Serial Test.
4. Linear Complexity Test.

To eliminate this disadvantage, a combined connection scheme for three LFSRs was used, in which one of them clocks the other two (Fig. 10). Inputs TRN1-TRN8 are designed to enter random numbers – 0 and 1 from the corresponding outputs of the SPC and BM. Inputs IN1-IN16 are used for user input. Thus, this scheme also provides for the use of a 16-bit static user password.

Taking into account the above PRNG scheme, the implementation of the proposed RN generator with two types of entropy sources in Simulink is as follows (Fig. 11).

Thus, the implementation of the proposed RN generator with two types of entropy sources in Simulink is finally obtained. Using this model, modeling was carried out and real results were obtained.

Simulation was carried out to confirm the obtained results. The model of the proposed random number generator built in the MATLAB/Simulink system is shown in Fig. 11. The results of the statistical analysis of the generated random sequences using the NIST SP 800-22 test package for the scheme shown in Fig. 11 are given in Table 1.

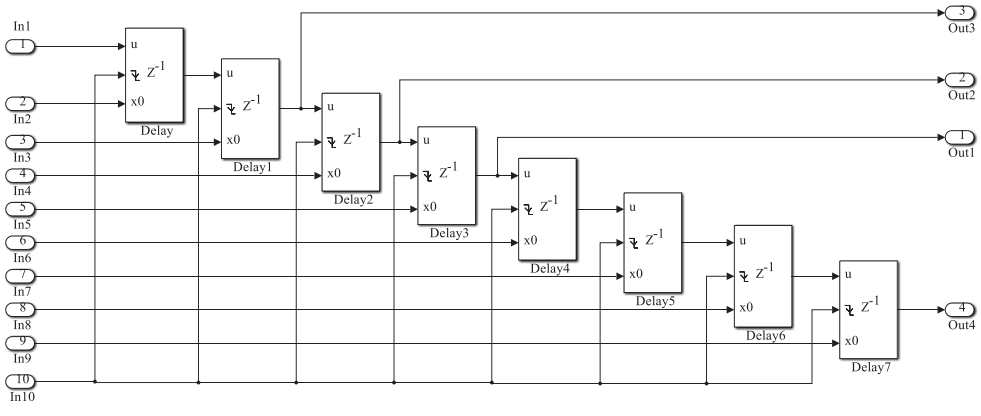

Fig. 8. Implementation of a shift register in Simulink

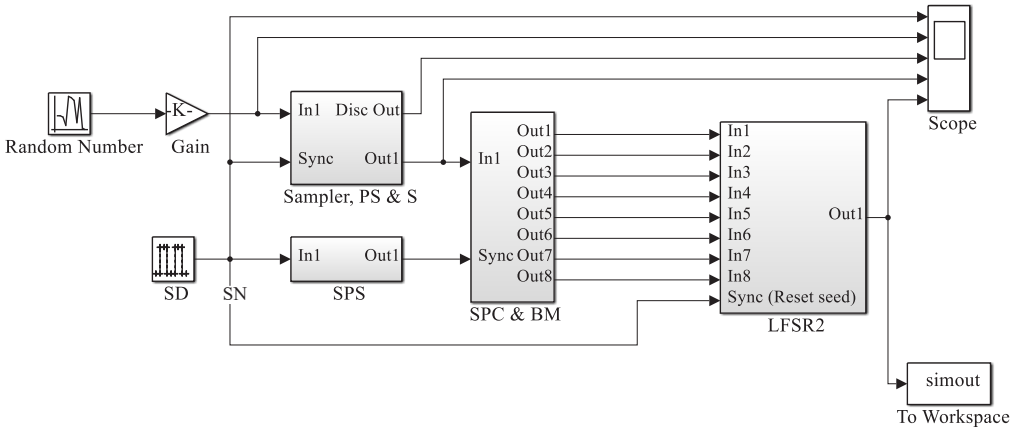

Fig. 9. Implementation of a random number generator with two types of entropy sources in Simulink

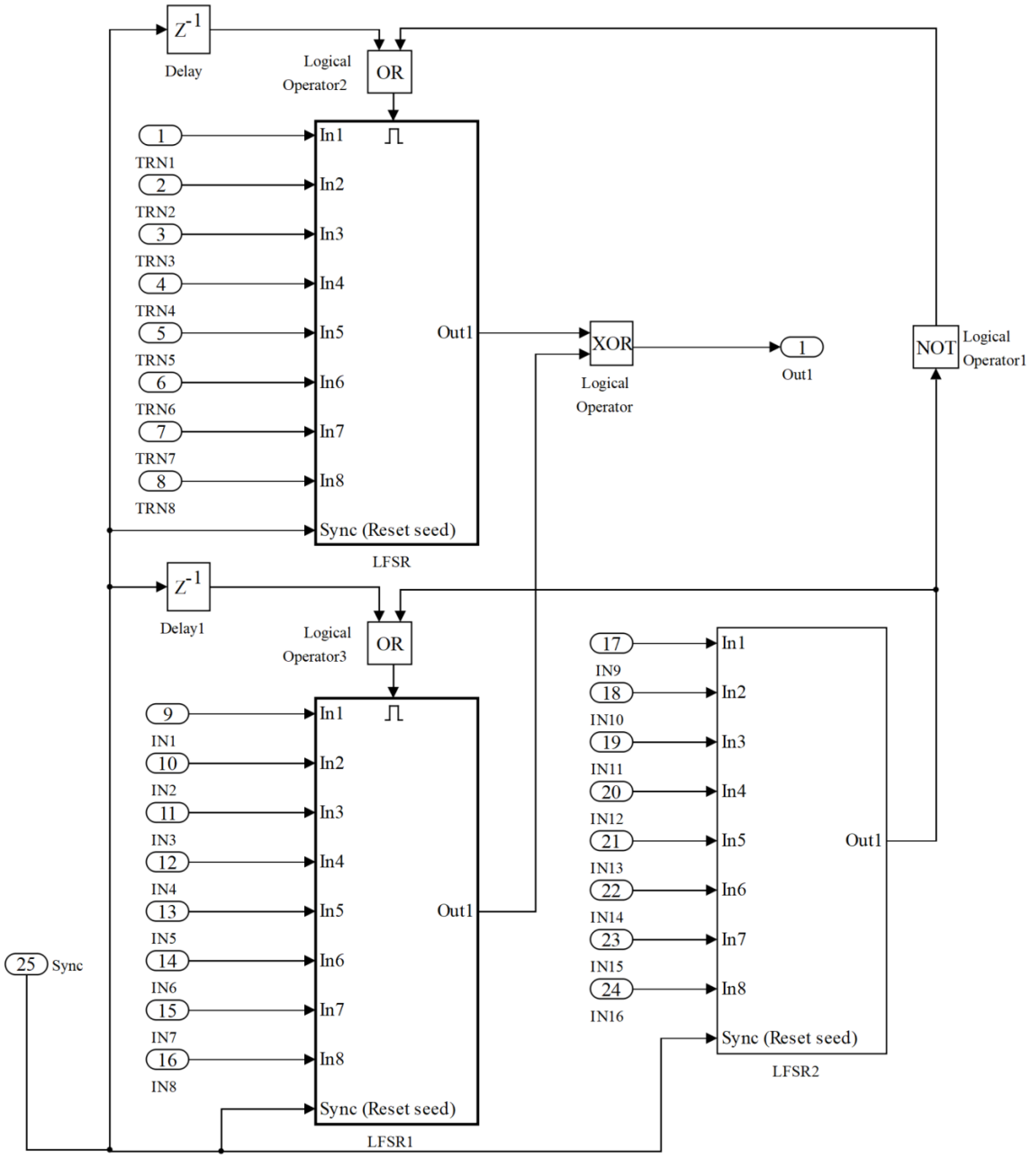

Fig. 10. Implementation of a pseudo-random number generator based on three linear feedback shift registers in Simulink

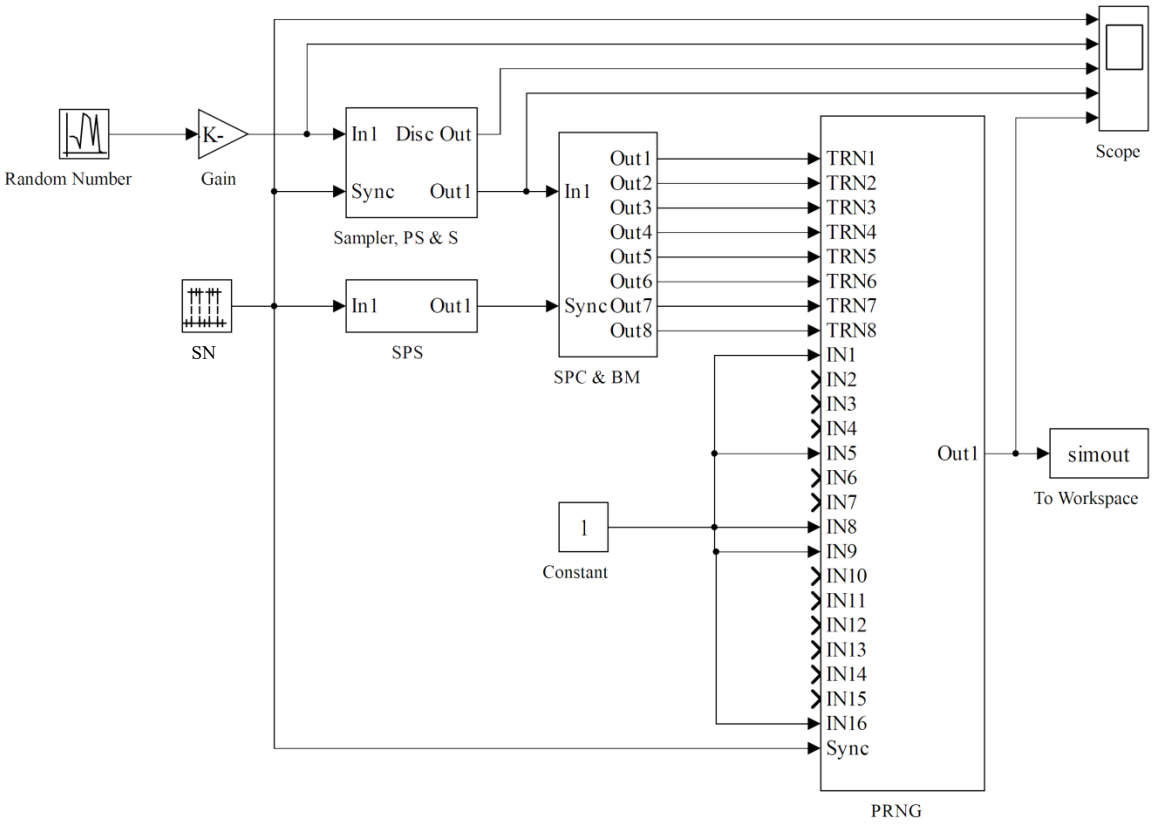

Fig. 11. Implementation of a crypto-resistant random number generator with two types of entropy sources in Simulink

Table 1

NIST SP 800-22 test results

Test name	P-value parameter	Test result
Frequency (Monobits) Test	0.001	passed
Frequency Test within a Block	0.319	passed
Runs Test	0.023	passed
Test for the Longest Run of Ones in a Block	0.011	passed
Binary Matrix Rank Test	0.002	passed
Discrete Fourier Transform (Spectral) Test	0.076	passed
Non-Overlapping Template Matching Test	0.262	passed
Overlapping Template Matching Test	0.005	passed
Maurer's «Universal Statistical» Test	0.631	passed
Linear Complexity Test	0.145	passed
Serial Test	0.071	passed
Approximate Entropy Test	0.843	passed
Cumulative Sums (Cusum) Test	0.001	passed
Random Excursions Test	0.065	passed
Random Excursions Variant Test	0.047	passed

More clearly, the results can be traced by oscillograms; for this, we consider the process of RN generation according to the diagrams shown in Fig. 12. When choosing the parameters of the clock pulse (Fig. 12, *a*), one should proceed from (1).

Taking these values, we get: $N=8$, $T_d=1$, $\Delta=50$, $T_{LFSR}=1$, $\tau_s=8$, and $Ts=50$. The physical meaning of the results obtained is as follows:

– during the time interval $\tau_s=8$, both the TRNG and the PRNG operate;

– at the time $t=8$ (clock pulse cutoff), the TRNG operation stops, and the TRN sequence accumulated in the BM is transferred to the PRNG input and serves as an initialization vector for it;

– during the time interval $T_s-\tau_s=42$, only the PRNG operates, which makes it possible for the TRN setter to switch to a new state that is not correlated with the previous one.

Fig. 12, *b*, *c* shows, respectively, the noise at the output of the TRN setter and the signal after sampling. Fig. 12, *c* shows a diagram of the TRN sequences formed during time «windows» with a duration of $\tau_s=8$. As you can see, in the first cycle, the sequence «10110010» is formed, and in the second – «01001100». The RN sequence obtained at the output of the PRNG/HRNG is shown in Fig. 12, *d*, *e*.

Analysis of Fig. 12, *a*–*e* showed that the results of simulation of the proposed model of a crypto-resistant random number generator with two types of entropy sources in Simulink can be competitive with software cryptoprotection methods.

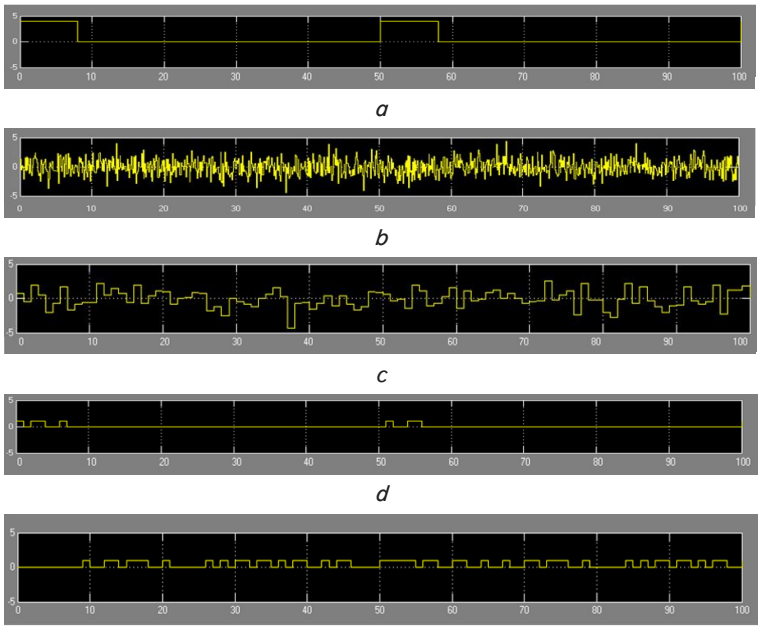

Fig. 12. Oscillograms of the random number generation process: *a* — a sequence of random numbers — clock pulses; *b* — a sequence of random numbers — noise at the output of the operational amplifier; *c* — a sequence of random numbers — noise after sampling; *d* — a sequence of random numbers — a sequence of true random numbers (output of the true random number generator); *e* — a sequence of random numbers — a sequence of random numbers (output of the pseudo-random number generator/hybrid random number generator)

Thus, the proposed approach to the implementation of a hardware cryptosystem can be.

6. Discussion of the results of modeling a hardware cryptosystem

To obtain a space of crypto-resistant encryption keys, devices called HRNG – hybrid random number generators are widely used that combine the advantages of two types of random number generators: PRNG and TRNG. In most HRNG, only one type of entropy source is used, which can be the main factor in reducing cryptographic strength and the appearance of random number periodicity. In the proposed implementation of the HRNG, two radically different sources of entropy are used: external – based on the noise of a Zener diode operating in the transient electrical breakdown mode, and internal – based on the undefined state of the TTL structure.

Tunneling breakdown is observed when the electron energy is less than the height of the potential barrier. In the case of an avalanche breakdown, under the action of a voltage of a back-applied electric field, the carriers receive a sufficiently high kinetic energy for impact ionization.

The PRNG synthesized as a combined connection of several LFSR and the use of an additional static 16-bit user password made it possible to improve the statistical characteristics of the sequences compared to the use of a single LFSR.

The simulation results based on the developed model using the NIST SP 800-22 statistical test package are presented in Table 1.

For clarity, additionally obtained results of modeling the operation of the random number generator with two types of entropy sources, shown in Fig. 12, *a*–*e* testify to the following.

Fig. 12, *b*, *c* shows, respectively, the noise at the output of the TRN setter and the signal after sampling. Fig. 12, *c* shows a diagram of the TRN sequences formed during time «windows» with a duration of τ_s=8. As you can see, in the first cycle, the sequence «10110010» is formed, and in the second – «01001100». The RN sequence obtained at the output of the PRNG/HRNG is shown in Fig. 12, *d*, *e*.

Unlike random number generators using a single linear feedback shift register, a pseudo-random number generator is proposed, which made it possible to improve the statistical characteristics of the generated sequences. This becomes possible due to the fact that the generator is synthesized in the form of a combined connection of several linear feedback shift registers and the use of an additional static 16-bit user password.

7. Conclusions

1. The analysis was carried out and the requirements for the operation of hardware cryptosystems such as pseudo-random number generators, true random number generators and hybrid random number generators were formed. The shortcomings of existing cryptosystems are revealed. As an option to eliminate the shortcomings of cryptosystems, a true random number generator with direct generation of true random numbers is proposed.

2. A model of a hardware cryptosystem was developed and built – a random number generator with two types of entropy sources: external – based on the noise of a Zener diode operating in the mode of unsteady electrical breakdown and internal – based on an undefined state of the transistor logic structure.

3. The simulation of the proposed model of a crypto-resistant random number generator with two types of entropy sources was carried out. As a result, a real picture of the sequence of random numbers at the output of the true random number generator/hybrid random number generator is obtained for various input parameters and effects on it.

4. The obtained results indicate that the pseudo-random number generator allowed improving the statistical characteristics of the sequences compared to the use of a single linear feedback shift register. The improvement was achieved as a result of the synthesis of the pseudo-random number generator in the form of a combined connection of several linear feedback shift registers and the use of an additional static 16-bit user password.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

References

1. Lienkov, S., Zhyrov, G., Pampukha, I., Chetverikov, I. (2019). Block Encryption Algorithm for Digital Information Using Open Keys for Selfgeneration of Closed Random Private Keys. 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). doi: https://doi.org/10.1109/atit49449.2019.9030509
2. Pampukha, I., Zhyrov, G., Druzhynin, V., Chetverikov, I., Lienkov, S., Komarova, L. (2021). Description and Application of Network and Terminal Security Device Based on the Block Algorithm of Cryptographic Transformation of Information Using Random Keys. 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT). doi: https://doi.org/10.1109/atit54053.2021.9678870
3. Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O., Korol, O., Milevskyi, S. et. al.; Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O. (Eds.) (2021). Synergy of building cybersecurity systems. Kharkiv: PC TECHNOLOGY CENTER, 188. doi: https://doi.org/10.15587/978-617-7319-31-2
4. Iovane, G., Amorosia, A., Benedetto, E., Lamponi, G. (2015). An Information Fusion approach based on prime numbers coming from RSA algorithm and Fractals for secure coding. Journal of Discrete Mathematical Sciences and Cryptography, 18 (5), 455–479. doi: https://doi.org/10.1080/09720529.2014.894311
5. Long, M., Chen, Y. (2019). Average throughput and BER analysis for energy harvesting communications. IET Communications, 13 (3), 289–296.
6. Glory, F. Z., Ul Aftab, A., Tremblay-Savard, O., Mohammed, N. (2019). Strong Password Generation Based On User Inputs. 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). doi: https://doi.org/10.1109/iemcon.2019.8936178
7. Lemire, D. (2019). Fast Random Integer Generation in an Interval. ACM Transactions on Modeling and Computer Simulation, 29 (1), 1–12. doi: https://doi.org/10.1145/3230636
8. Chakrabarty, D., Sarma, B. K. (2017). Comparison of degree of randomness of the tables of random numbers due to Tippet, Fisher & Yates, Kendall & Smith and Rand corporation. Journal of reliability and statistical studies, 10 (1), 27–42. Available at: https://www.journal.riverpublishers.com/index.php/JRSS/article/view/2205/1526
9. Ewert, M. (2018). A Random Number Generator Based on Electronic Noise and the Xorshift Algorithm. Proceedings of the 2018 VII International Conference on Network, Communication and Computing – ICNCC 2018. doi: https://doi.org/10.1145/3301326.3301359
10. Kim, J., Nili, H., Truong, N. D., Ahmed, T., Yang, J., Jeong, D. S. et. al. (2019). Nano-Intrinsic True Random Number Generation: A Device to Data Study. IEEE Transactions on Circuits and Systems I: Regular Papers, 66 (7), 2615–2626. doi: https://doi.org/10.1109/tcsi.2019.2895045
11. Kyrychok, R., Laptiev, O., Lisnevskyi, R., Kozlovskyi, V., Klobukov, V. (2022). Development of a method for checking vulnerabilities of a corporate network using Bernstein transformations. Eastern-European Journal of Enterprise Technologies, 1 (9 (115)), 93–101. doi: https://doi.org/10.15587/1729-4061.2022.253530
12. Petrivskyi, V., Shevchenko, V., Yevseiev, S., Milov, O., Laptiev, O., Bychkov, O. et. al. (2022). Development of a modification of the method for constructing energy-efficient sensor networks using static and dynamic sensors. Eastern-European Journal of Enterprise Technologies, 1 (9 (115)), 15–23. doi: https://doi.org/10.15587/1729-4061.2022.252988
13. Vlasyk, H., Zamrii, I., Shkapa, V., Laptiev, S., Kalyniuk, A., Laptieva, T. (2021). The Method of Solving Problems of Optimal Restoration of Telecommunication Signals. 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory (ATIT). doi: https://doi.org/10.1109/atit54053.2021.9678649
14. Laptiev, O. et. al. (2019). The Method of Hidden Transmitters Detection based on the Differential Transformation Model. International Journal of Advanced Trends in Computer Science and Engineering, 8 (6), 2840–2846. doi: https://doi.org/10.30534/ijatcse/2019/26862019
15. Laptiev, O., Tkachev, V., Maystrov, O., Krasikov, O., Open'ko, P., Khoroshko, V., Parkhuts, L. (2021). The method of spectral analysis of the determination of random digital signals. International Journal of Communication Networks and Information Security (IJCNIS), 13 (2). Available at: https://www.ijcnis.org/index.php/ijcnis/article/view/5008
16. Ruban, I., Martovytskyi, V., Lukova-Chuiko, N. (2016). Designing a monitoring model for cluster super-computers. Eastern-European Journal of Enterprise Technologies, 6 (2 (84)), 32–37. doi: https://doi.org/10.15587/1729-4061.2016.85433