Hidden Embedding of Encrypted Data into Images via Hamming Coding and Pseudorandom Positioning

Nataliia Kukharska
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0000-0002-0896-8361

Roman Martyniuk
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0009-0008-0603-2771

Orest Polotai
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0000-0003-4593-8601

Serhiy Semenyuk
Department of Security of Information Technologies
Lviv Polytechnic National University
Lviv, Ukraine
ORCID 0000-0002-8143-5887

***Abstract*—The article proposes a method for embedding encrypted data into digital images, in which blocks of p bits are integrated in accordance with the principles of the Hamming code. For each p-bit block, a group of $2^p - 1$ container elements is constructed, and only a single element within each group is modified. This design provides high embedding efficiency while ensuring minimal distortion of the carrier image. The positions of the container elements are determined by a pseudorandom sequence derived from the fractional part of the square root of a prime number, which enhances the method's resistance to statistical detection. Prior to embedding, the message undergoes encryption using a bitwise XOR operation, thus preserving confidentiality even in cases of partial steganogram recovery or exposure of the embedding positions. Experimental results confirm the viability of the proposed generalized approach for the development of scalable, low-visibility, and cryptographically resilient steganographic systems.**

Keywords—information security, computer steganography, data hiding, digital image, LSB method, Hamming code, pseudorandom number generator, square-root calculation of prime numbers, insertion efficiency.

I. INTRODUCTION

Transmitting information over open communication channels poses significant risks to message confidentiality, as such channels do not provide sufficient protection against interception or unauthorised access. In modern networked environments, where the volume of digital exchanges continues to increase, and traffic-analysis techniques are becoming increasingly sophisticated, even basic channel monitoring may enable attackers to access the content of private communications. As a result, confidential data – ranging from personal messages to sensitive corporate information – can be stolen, exposed, or exploited to facilitate subsequent attacks. For this reason, ensuring confidentiality remains one of the fundamental objectives of securing information transmitted across open networks.

To ensure the security of message content transmitted over potentially vulnerable channels, it is essential to employ methods that not only hinder interception but also prevent adversaries from analysing the content should they gain access to the transmitted data. Under such conditions, cryptographic and steganographic techniques become particularly significant, especially when they are integrated within a single system. The combined use of these methods creates a form of symbiosis in which each approach reinforces the other: encryption guarantees that the message content remains inaccessible even if intercepted, whereas steganography reduces the likelihood that an attacker will detect the presence of confidential, encrypted data. This combined strategy establishes a multi-layered protection model capable of effectively countering a wide range of attacks.

In modern information transmission systems, multimedia files are regarded as one of the most promising containers for steganographic data hiding. Their widespread use, substantial information capacity, and diversity of formats create favourable conditions for embedding secret messages in a manner that does not arouse suspicion among external observers. Among the various types of multimedia objects, such as audio, video, and text documents, digital images warrant particular attention.

Images possess a range of characteristics that make them particularly suitable and effective as data-hiding containers. They constitute one of the most widely distributed forms of digital content and are continuously shared across social networks, messaging platforms, and web applications. Consequently, their transmission, even when they contain embedded confidential information, does not draw undue attention, including from potential adversaries, and is generally perceived as a routine and unremarkable aspect of digital communication.

Secondly, digital images contain a substantial amount of redundant information arising from the characteristics of colour models, encoding schemes, and the internal pixel-based structure of the data. This redundancy creates opportunities for subtle modifications that have minimal or no impact on visual quality, making the embedded information virtually imperceptible to the human eye.

Another important advantage of using images as carriers of hidden messages is the wide variety of available formats – from uncompressed raster formats such as PNG and BMP to compressed formats like JPEG – each possessing distinct properties. These differences enable the selection of optimal steganographic techniques depending on desired criteria such as robustness, embedding capacity, or imperceptibility. Owing to this flexibility, images remain one of the most versatile and

extensively studied media containers in the field of steganography.

Taken together, these factors highlight the relevance and high effectiveness of using digital images as containers for hidden messages. They enable an optimal balance between natural perceptibility, embedding capacity, robustness against attacks, and the overall quality of camouflage, making images a key component of modern steganographic systems.

A wide variety of methods are employed to implement steganographic data hiding in digital images, each offering specific advantages, limitations, and appropriate areas of application [1-4]. The most common techniques operate in the spatial domain, embedding information directly into pixel values, or in the frequency domain, where data is inserted by modifying coefficients obtained after transforming the image into a spectral representation (for example, via discrete cosine or wavelet transforms). In addition, several advanced approaches rely on adaptive models, statistical characteristics, or fractal properties of images, enabling higher resistance to steganalytic attacks and improved visual quality after embedding.

Despite the wide range of available techniques, spatial-domain methods remain particularly popular, especially those that modify insignificant pixel components. Among these, the least significant bit (LSB) approach occupies a prominent position due to its simplicity, high embedding capacity, and minimal impact on the visual quality of the host image. By altering only the least significant bits of colour channels, information can be embedded in a manner that is imperceptible to the human eye, making LSB one of the foundational and most extensively studied steganographic techniques.

However, despite its simplicity and efficiency, the classical LSB method has several notable limitations, including low resistance to statistical steganalysis, susceptibility to transformation-based attacks, and potential distortions introduced during image processing. To address these shortcomings, a variety of enhanced approaches have been developed that improve the invisibility, robustness, and overall adaptability of LSB-based embedding schemes.

One important direction for improving LSB-based techniques involves adaptive methods that take local image characteristics into account during data embedding. Unlike the classical approach, which applies modifications uniformly across the entire image, adaptive algorithms identify regions with high texture or noise and utilise these areas as the most suitable for concealing information. This strategy minimises visual artefacts and reduces the likelihood that embedded data will be detected through statistical analysis.

Another class of enhancements involves multi-level or multi-bit embedding schemes, in which information is inserted not only into a single least significant bit but into several lower-order bits, often guided by pseudo-random sequences or quality-control parameters. This approach increases the embedding capacity of the container while preserving an acceptable level of imperceptibility.

Pseudorandom LSB algorithms, which employ pseudorandom sequence generators to determine the pixel positions for embedding information, have also gained widespread use. This approach considerably complicates steganalysis, as the placement of data is no longer linear or easily predictable.

A distinct category is formed by robust LSB methods that include additional protection mechanisms, such as the use of error correction codes (e.g., Hamming codes) or the combination of LSB with steganography in spectral domains. These enhancements not only improve concealment but also increase the algorithm's resilience to common image processing operations, including scaling, filtering, and compression.

Collectively, these advanced strategies significantly enhance the classical LSB method, achieving a more effective balance between imperceptibility, embedding capacity, and resistance to contemporary steganalysis techniques.

The aim of this work is to develop an approach based on a multi-bit LSB algorithm for concealing confidential messages in digital images and to integrate the use of Hamming codes and pseudorandom sequence generators. This study analyses the results obtained using a pseudorandom sequence generator constructed from a square-root calculation of prime numbers.

II. Related Work

In our previous research [5-6], we explored the application of this generator in steganographic and cryptographic contexts. Specifically, in [5], we proposed an algorithm for embedding encrypted information into WAV audio files to enable secure transmission over open networks. The use of this generator made it possible to increase the variability of the method of replacing the least significant bits.

The study [6] presents a method for embedding and extracting digital watermarks, based on a combination of the Hamming coding algorithm and a pseudorandom number generator derived from the square root of a prime number. Experiments involving various types of attacks, such as copying, collaging, text insertion, and object removal, demonstrated the high efficacy of the proposed scheme in detecting unauthorized modifications to images: 100% of altered pixels were correctly identified, no modifications were missed, and the false positive rate remained very low (<0.14%). Based on these results, the method was deemed reliable and recommended for information security applications, including digital image authentication and integrity protection.

The work presented in [7] describes steganographic algorithms in which information embedding is based on binary and ternary Hamming codes. A key advantage of these methods is their increased embedding efficiency, defined as the ratio of the number of bits in the hidden message to the number of modified bits in the container. In contrast to the classical LSB approach, where average efficiency is approximately two message bits per modification (since roughly half of the least significant bits of the container already match the message bits), Hamming code-based methods achieve even higher efficiency. They enable multiple message bits to be embedded by modifying only a single bit of the container, thereby significantly reducing distortion. These algorithms will serve as a theoretical foundation for further development and enhancement of the data-embedding approach using a pseudorandom number generator.

III. Proposed Method Theory

We will embed the confidential message into digital images, which in this case will act as a container – a carrier of hidden data. The container is a set of numerical values that describe the pixels of the image. For 8-bit grayscale images, this is an $M \times N$ matrix, where each element is a brightness value from 0 to 255. In the case of colour RGB images, the

container takes the form of an $M \times N \times 3$ tensor, in which each pixel is represented by a triplet of R, G, and B components in the same range of values. At the bit level, each pixel component is represented as an 8-bit sequence, the least significant bits of which will be used to embed the message bits.

Let us introduce some notation.

p – the number of bits in a message block.

The number of container bits required to embed a single message block is calculated using the formula $2^p - 1$.

Matrix M has a size of $p \times (2^p - 1)$ and contains in its columns all non-zero binary vectors of length p (arranged in a specific order). For example, for $p = 3$:

$$M = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \quad (1)$$

A. Message Embedding

Step 1. Let us represent the message as a sequence of K bits (for example, by encoding it in UTF-8 and then converting the bytes into a bit stream).

If necessary, let us supplement it with service bits (padding) so that the length of the resulting sequence is a multiple of p, which is necessary for the further formation of information blocks.

After that, we will perform a cryptographic transformation of the message using a stream cipher. To generate the key pseudorandom sequence, a generator based on the binary representation of the square root of a prime number is employed [5, 8]. To obtain pseudo-random bits, we will consider the fractional part of the corresponding irrational number. We will use the resulting bits as a gamma, combining them with the message bits using the XOR operation.

After encryption, we will split the resulting bit stream into blocks of p bits in length. We will consider each such block as a message vector m in the embedding algorithm described below.

Step 2. To hide p bits of a single message block from the container, we will select a sequence of $2^p - 1$ bytes (or channel values).

To increase the stability of the algorithm, we suggest selecting the byte positions pseudo-randomly using a generator based on a square root calculator for a prime number (another prime number can be considered here, not necessarily the one in step 1), but this time in decimal form [5, 6, 9, 10]. Such a generator should ensure that a large number of decimal digits of the root are obtained – in our case, at least $\frac{K(2^p-1)}{p}$ digits after the decimal point. At the same time, each digit must be determined with high accuracy, since any calculation error can compromise the pseudo-random nature of the resulting sequence. In this regard, rough approximation methods can only be used to determine the initial value, while subsequent calculations must be performed with a high degree of accuracy.

For example, let us calculate the square root of the number 3 and obtain the infinite irrational number $\sqrt{3} = 1.732050807568877$ ... Hereinafter, we will only consider its fractional part. Let us divide it into fragments of fixed length, for example, 5 digits each. As a result, we will obtain a sequence of numbers: 73205 32050 20508 05080 ... . We will interpret each of them as the position number of the corresponding container byte, the least significant bits of which will be involved in hiding the confidential message.

Monograph [4] emphasizes that pseudorandom numbers must be unique, as repeated values can lead to the overwriting of already embedded bits, resulting in the distortion of the hidden information. When the message size is substantially smaller than the container capacity, the probability of such collisions is low. Nevertheless, to fully eliminate this risk, it is recommended to track previously used positions and verify, prior to embedding each message block, that the selected bytes are being used for the first time.

The described approach ensures a non-contiguous (non-sequential) distribution of container bytes used for embedding. To further enhance the uniformity and unpredictability of byte selection, we propose dividing the fractional part of the irrational number into subsequences of variable length rather than fixed-length fragments. This is achieved by generating an additional pseudorandom sequence using a generator based on the square root of a different prime number. Each digit obtained from this sequence, after scaling to the range 3–6 (with the upper limit generally determined by the container size), is interpreted as the length of the next subsequence, the numerical value of which corresponds to a byte position. This method introduces variability in fragment lengths and ensures a more uniform, unpredictable, and stegoanalysis-resistant distribution of byte positions in the container, thereby increasing the overall security of the embedding algorithm.

Step 3. From each byte of the container, whose position is determined by the pseudo-random method described above, we will extract the least significant bit. By collecting these bits, we will form a coverage vector c with length $2^p - 1$.

If the confidential message is large and the container file has a limited size, there may be a situation where there are not enough available positions to embed all data blocks. In this case, a multi-bit approach can be used, in which not only the least significant bits but also the next most significant bits are used to embed the data. This allows you to move to the second or third layer of concealment, increasing the total capacity of the container.

Of course, using the senior bits may slightly increase the risk of noticeable distortions, so this approach should be used taking into account the characteristics of the image and the acceptable level of modifications. However, it is the multi-level scheme that makes it possible to scale the algorithm and work with a larger volume of confidential data when necessary.

Step 4. Let's calculate $m' = Mc$ modulo 2 (bitwise XOR). The vector m' is the message that corresponds to the current LSB without modifications.

Step 5. Let the desired message (for this block) be m (length p). Let us calculate:

$$v = m' - m \quad (2)$$

The vector v shows what correction needs to be made so that $Ms = m$.

Step 6. If v is a zero vector, then $Mc = m$ and nothing needs to be changed: $s = c$.

Otherwise, we need to find the column of the matrix M that matches v. Let this column have index j. It is necessary to flip the bit at position j of vector c (i.e., change the LSB of the

corresponding byte) to obtain the stego-vector s with the desired property $Ms = m$.

Step 7. Change the corresponding byte in the container so that its LSB takes on a value equal to the new bit s. This is usually done by performing a +1 or −1 operation on the byte value to change the LSB and minimise visible artefacts. Both operations are acceptable.

Repeat steps 2–7 for each block of the message until all the bits of the message are embedded in the container.

B. Extracting the hidden message

During extraction, we process the same pixels or bytes of the container that were used to embed the data. From each of them, we read the corresponding least significant bits and form a vector c. Next, we process this vector according to the rules of the Hamming code – we calculate the syndrome and, based on it, determine the information bits hidden in the current block. By repeating this procedure for all the blocks involved, we obtain a bit sequence containing the encrypted hidden message.

After that, we move on to the decryption stage. Using the same pseudo-random sequence generator as during encryption, we reproduce the key stream and perform a bitwise XOR operation between it and the obtained sequence. As a result, we obtain the original bit sequence of the message, which we convert into bytes and then into a text representation (for example, in UTF-8 encoding).

IV. Study of the Effectiveness of the Scheme

Fig. 1(a) and Fig. 1(c) show digital images with embedded messages. The length of the hidden text is 108 characters, and the size of one block is 3 bits. Red dots indicate modified pixels (i.e., those in which the LSB has been changed).

A pseudorandom number generator based on calculating the square root of 5 was used to select the positions of the container pixels. Fig. 1(a) shows an image in which the positions for embedding were determined by sub-sequences of 3 digits, and fig. 1(c) shows an image in which the positions were determined by sub-sequences of 5 digits.

Fig. 1(b) and Fig. 1(d) show the corresponding embedding maps: red indicates pixels in which the LSBs have been modified, and blue indicates pixels whose least significant bits were involved in the hiding process but did not undergo any changes.

As observed, when pixel positions are determined using three-digit subsequences, the selected positions tend to cluster locally. This results in a less uniform and more predictable distribution of modified pixels across the image. Frequent repetitions of pseudorandom values in this scenario increase the likelihood of collisions and render the embedding process more susceptible to statistical analysis.

In contrast, employing five-digit subsequences considerably expands the range of possible positions, leading to a more even distribution of modified pixels without pronounced clustering. This method of index formation enhances the unpredictability of pixel selection and improves the overall steganographic robustness of the algorithm.

Fig. 2 show the distribution of pixels involved in the information embedding process for different values of parameter p.

In the case of $p = 5$ (Figs. 2(a), 2(b)), it can be seen that the coloured dots evenly cover the entire image, and the number of modified pixels (marked in red) is relatively small – only 173. This is because to hide each 5-bit block, it is necessary to change no more than one least significant bit among 31 pixels. Under these conditions, the insertion efficiency is five bits of information per modification. Therefore, even despite the significant number of container positions involved (blue and red dots together), the actual changes remain minimal and have virtually no effect on the visual quality of the image.

For $p = 8$ (Figs. 2(c), 2(d)), the blue dots exhibit a noticeably denser distribution, reflecting the fact that an 8-bit message block requires a substantially larger number of container positions – 255 pixels. Concurrently, the number of red dots decreases to 108, indicating that fewer actual modifications occur compared to the case of $p = 5$. The embedding efficiency also increases significantly: a single modification allows the insertion of 8 information bits, making the method less perceptible to the human eye and more resistant to statistical detection.

Thus, analysis of the figures confirms the following patterns:

- As the parameter p increases, the embedding efficiency rises, since more message bits are concealed per modification.
- The total number of modified pixels decreases relative to the size of the hidden message.
- However, the number of container positions required for embedding grows as longer blocks necessitate more positions in the image.

Table 1 shows the values of MSE, SNR, and PSNR metrics for different values of parameter p (3, 5, and 8). This allows us to evaluate how block size affects the quality of steganographic embedding, the degree of distortion, and the visual imperceptibility of changes.

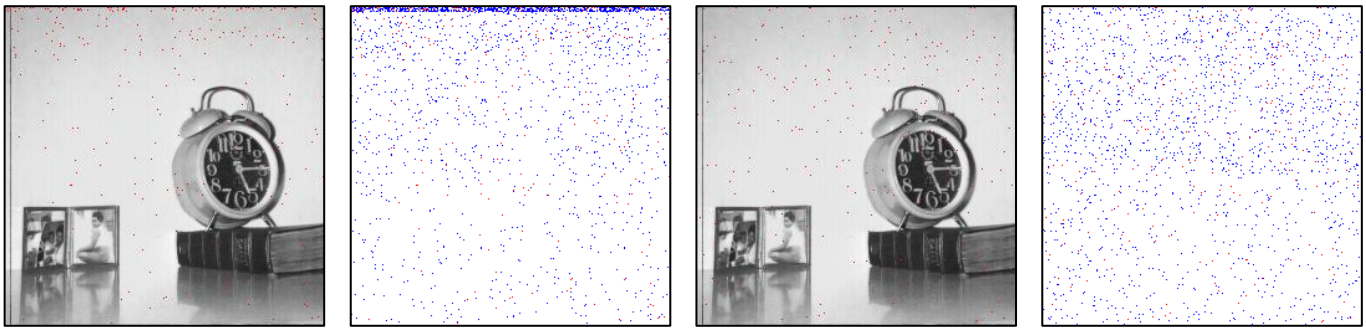
TABLE I. Parameters that show the Efficiency of the Scheme

Parameters	**Block size**		
	$p = 3$	$p = 5$	$p = 8$
MSE	0.0035	0.0023	0.0015
SNR (dB)	70.3112	72.1004	73.9910
PSNR (dB)	72.6595	74.4487	76.339

The results exhibit a clear trend: as p increases, the quality metrics improve. The MSE values decrease steadily (from 0.0035 to 0.0015), indicating a reduction in the root-mean-square deviation between the original image and the steganographic container.

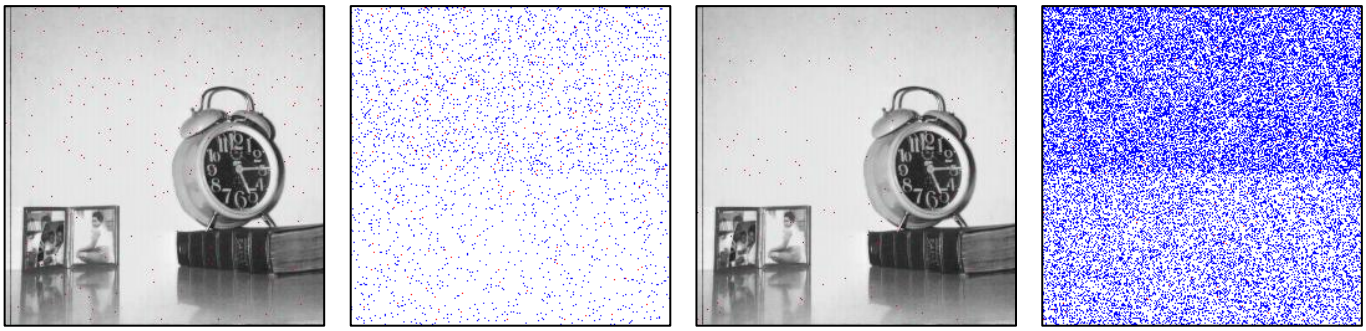
Concurrently, both SNR and PSNR values increase. The SNR increases from 70.3dB to 74.0dB, indicating an improvement in the signal-to-noise ratio. The change is even more pronounced for PSNR, which increases from 72.66dB at $p = 3$ to over 76.34dB at $p = 8$. These PSNR values correspond to steganographic methods that produce minimal distortion, with the embedded changes remaining visually imperceptible.

Therefore, analysis of Table 1 shows that increasing the p parameter improves embedding quality and reduces container distortion, making the algorithm more efficient and even less noticeable.



(a) (b) (c) (d)

Fig. 1. Digital images with embedded messages: (a) – embedding positions were determined by sub-sequences of 3 digits, (c) – by sub-sequences of 5 digits, (b) and (d) – corresponding bit distribution maps



(a) (b) (c) (d)

Fig. 2. Digital images with embedded messages: (a) – block length $p = 5$, (c) – block length $p = 8$, (b) and (d) – corresponding bit distribution maps

V. CONCLUSION

This paper proposes and investigates a method for covertly embedding encrypted data into digital images, combining pseudorandom scattering of container positions with the use of Hamming codes to minimize distortion of the image structure. Bitwise XOR encryption ensures the confidentiality of the original message, while embedding information exclusively in the least significant bits preserves the visual quality of the container.

Experimental results demonstrated that for each information block of length p, only one pixel out of $2^p - 1$ modified pixels was altered, indicating high embedding efficiency and low perceptibility. Comparison of embedding maps confirmed that the length of the subsequences derived from the fractional part of the square root of a prime number, which determines pixel positions, directly affects the uniformity of their distribution.

The proposed approach provides a foundation for constructing multi-level data protection systems and further enhancing steganographic algorithms.

A promising direction for future research is the extension of the method through the use of ternary Hamming codes and pseudorandom generators based on the calculation of prime-number square roots in the ternary number system. This approach would enable the handling of larger data blocks and enhance the flexibility of the embedding process. Furthermore, employing ternary structures has the potential to increase embedding efficiency by reducing the number of modified container elements while also improving resistance to steganalytic attacks.

REFERENCES

[1] P.C. Mandal, I. Mukherjee, G. Paul and B.N. Chatterji, “Digital Image Steganography: A Literature Survey”, Information Sciences, 2022, vol. 609, pp. 1451-1488.

[2] V. Gnanalakshmi and G. Indumathi, “A review on image steganographic techniques based on optimization algorithms for secret communication”, Multimedia Tools and Applications, 2023, vol. 82, pp. 44245–44258.

[3] I. Kazmidi and V. Zubok, “Image steganography – classic and promising methods: a study”, Theoretical and Applied Cybersecurity, 2025, vol. 7, No. 1, pp. 14-19.

[4] G. Konakhovych, D. Progonov and O. Puzyrenko, Komp'juterna steghanoghrafichna obrobka j analiz muljtymedijnykh danykh. Kyiv: Centr navchaljnoji literatury, 2018.

[5] N. Kukharska, A. Lagun and O. Yashchyk, “Crypto-steganographic system based on the solver of the square root of a prime number,” Proceedings – International Conference on Advanced Computer Information Technologies, ACIT, 2024, pp. 535–538.

[6] N. Kukharska, A. Lagun, S. Semenyuk, O. Polotai, O. Hospodarskyy and M. Ozhha, “Embedding a digital watermark into an image using a Hamming code and a solver of the square root of a prime number,” Proceedings – International Conference on Advanced Computer Information Technologies, ACIT, 2025, pp. 537–541.

[7] D. Lerch, “Binary Hamming Codes in Steganography”. [Online]. Available: https://daniellerch.me/stego/codes/binary-hamming-en/

[8] A. Horpeniuk and N. Luzhetska, “Generator of pseudo-random numbers of the square root of simple numbers,“ Bulletin of the National university “Lviv Polytechnic“ Automation, measurement and control, 2013, No. 753, pp. 45–50.

[9] V. Dimitrievska Ristovska and V. Bakeva, “Comprasion of the results obtained by pseudo random number generator based on irrational numbers,” Mathematical Modeling, 2017, Vol. 1, No. 4, pp. 167–170.

[10] V. Dimitrievska Ristovska, “Pseudo random generator based on irrational numbers“, ICT-Innovations 2017, Data Driven Innovations, Conference Web Proceedings, 2017, pp. 105–113.