Turkish Journal of Science & Technology
19(2), 443-450, 2024

Research Paper
https://doi.org/10.55525/tjst.1511432

A New Cryptographic Key Planning Algorithm Based on Blum Blum Shub

Songül KARAKUŞ[1*], Fırat ARTUĞER[2]
[1] Computer Engineering Department, Engineering-Architecture Faculty, Bitlis Eren University, Bitlis, Turkey
[2] Computer Engineering Department, Engineering Faculty, Munzur University, Tunceli, Turkey
[*1] skarakus@beu.edu.tr, [2] firatartuger@munzur.edu.tr

(Geliş/Received: 06/07/2024; Kabul/Accepted: 28/09/2024)

Abstract: The Blum Blum Shub (BBS) algorithm is one of the known powerful pseudo random number generators. This algorithm can be used for key generation. BBS is basically based on the product of two large prime numbers and a seed value. The selection of these values is a critical issue. In this study, a new approach is proposed to overcome this problem. In the proposed approach, a prime number pool is first created. At this point, the user sets a start and end value. The primes in this range are generated and stored in an array. Then, two primes are randomly selected from this prime number pool with chaotic maps. The positions of these prime numbers in the array are recorded. The seed value is taken as the sum of the positions of these two primes. In other words, the parameters to be selected will be randomly selected in the ranges that the user will enter at that moment. In this study, two random bit sequences were obtained in this way. These sequences are 1 million bits long. NIST SP 800-22 tests were applied to these sequences and the sequences successfully completed all tests.

Key words: Blum blum shub, key generator, RNG, NIST SP 800-22 test, chaotic map.

Blum Blum Shub Tabanlı Yeni Bir Kriptografik Anahtar Planlama Algoritması

Öz: Blum Blum Shub (BBS) algoritması bilinen güçlü sözde rastgele sayı üreteçlerinden bir tanesidir. Bu algoritma anahtar üretiminde kullanılabilir. BBS temelde iki büyük asal sayının çarpımına ve bir tohum değerine dayanmaktadır. Bu değerlerin seçilmesi oldukça kritik bir konudur. Bu çalışmada bu problemin üstesinden gelmek için yeni bir yaklaşım önerilmiştir. Önerilen yaklaşımda öncelikle asal sayı havuzu oluşturulmaktadır. Bu noktada kullanıcı bir başlangıç ve bitiş değeri belirlemektedir. Belirlenen bu aralıktaki asal sayılar üretilerek bir diziye kaydedilir. Daha sonra bu asal sayı havuzundan kaotik haritalar ile rastgele iki asal sayı seçilir. Seçilen bu asal sayıların dizideki konumları kaydedilir. Tohum değeri ise bu iki asal sayının konumları toplamı olarak ele alınmıştır. Yani seçilecek olan parametreler o an kullanıcının gireceği aralıklarda rastgele bir şekilde seçilecektir. Bu çalışmada bu şekilde rastgele iki bit dizisi elde edilmiştir. Bu diziler 1 milyon bit uzunluğundadır. Elde edilen bu dizilere NIST SP 800-22 testleri uygulanmış olup diziler tüm testleri başarı ile tamamlamıştır.

Anahtar kelimeler: Blum blum shub, anahtar üreteci, RNG, NIST SP 800-22 test, kaotik harita.

1. Introduction

Today, with the rapidly developing internet technologies, the amount of data is increasing exponentially day by day. Among these data, especially personal and private ones must be protected while being transmitted or stored over networks. At this point, the concept of confidentiality emerges. The most basic technique to ensure the confidentiality of data is encryption. In other words, to ensure the confidentiality of the data, it is necessary to encrypt it with an effective encryption algorithm. Encryption approaches have been developing rapidly from the past to the present. New encryption algorithms are needed according to the increasing amount of data and various requirements in different applications. Therefore, the importance of studies on encryption approaches is increasing day by day [1]. Encryption algorithms are generally classified in two ways. The first one is the stream cipher approach. In these approaches, bits are encrypted one by one as a stream. These algorithms are highly secure but become difficult to implement as the amount of data increases. The other approach is block cipher. In block cipher, the data is divided into equal blocks and each block is encrypted separately. The encrypted blocks are then combined to obtain encrypted data. The most important advantage of these algorithms is that they work effectively, i.e. fast, no matter how large the data is. These structures are frequently used today and are of vital importance [2]. Today, the block cipher standard is the AES [3] algorithm. In both stream cipher algorithms and block cipher algorithms, all processes are known. The only unknown parameter in these algorithms is the key value. Especially in stream ciphers, since the data is passed directly through the XOR component with the key value, the key must be secure and secret [4]. How to obtain secure key values to be used in these encryption algorithms is a critical issue. Studies on this subject are increasing day by day. In other words, the complexity of the algorithms used to ensure data security is not sufficient alone. At the same time, the keys used must also be secure [5]. Random

[*] Corresponding author: skarakus@beu.edu.tr. ORCID Number of authors: [1] 0000-0003-1999-0203, [2] 0000-0002-4096-0458

number generators (RNG) are used to obtain key values securely. In other words, these keys must be obtained randomly.

RNG structures are frequently used in many fields, especially in computer science. The need for random numbers has increased in recent years. Physics, biology, simulation, and security fields are some of them. Especially in Monte Carlo simulations, random numbers are frequently used [6]. However, if random numbers are to be used in cryptological applications, they should be chosen very carefully. RNG structures are classified in two ways. The first one is true random number generators (TRNG). Here, numbers are obtained from physical sources such as mouse movements, processor run times, and radioactive disturbances. With these structures, secure numbers that are completely random can be obtained. However, these structures are often difficult to implement. The other structure is pseudo random number generators (PRNG). These are approaches in which random numbers are generated by mathematical and algorithmic processes. These techniques are easier to develop and implement. However, it is necessary to be very careful when generating these numbers. To decide that these numbers are random, various tests must be performed. Because there is an initial value called seed in PRNG structures. When this value is estimated, the same numbers can be generated again. This is an advantage for the reproduction of the sequence. However, this value should not be obtained by attackers. The capture of these values by attackers makes it easier for systems to exploit security vulnerabilities [7]. The BBS algorithm is one of the important PRNG structures. However, for this algorithm to be used in key generation, its parameters must be selected effectively. In this study, chaotic maps are used in the selection of these parameters. Chaotic maps are used in the generation of random numbers and many cryptographic applications [8]. The BBS algorithm has been used in many cryptographic studies. Some of these studies are mentioned below.

In a study by Joey [9], it was aimed to increase the random number sequence and the bits received per iteration to increase the efficiency of the BBS. To achieve this goal, it is proposed to modify the second-order generator of BBS with a matrix generator by squaring a 2x2 matrix that produces four outputs per iteration. Arroyo and Delima [10] proposed a method that combines Affine cipher and BBS algorithm. In the proposed method, the BBS algorithm is used to generate one of the secret keys by modifying the Affine cipher. To increase the unpredictability of the ciphertext, a random key sequence is generated by the BBS algorithm. According to the results obtained, it is found that the proposed method is more secure in general. Surbakti, Fauzi, and Khair [11] proposed a hybrid system using Rivest Shamir Adleman (RSA) and BBS algorithms to improve the security of database files in Binjai Regional Public Service Agency. In their proposed system, RSA algorithm is used to encode the content of the text file. Since RSA algorithm uses two different keys in encryption and decryption processes, BBS algorithm is used for key generation. As a result, the RSA and BBS hybrid system was found to work well. Rambe, Nababan, and Nasution [12] developed a method using RC5 and BBS algorithms in their research to increase message security. In this method, the key generated using BBS is integrated into the RC5 encryption algorithm to prevent key duplication and to make key generation more randomized. According to the results obtained from the tests, it is seen that the processing speed is independent of the number of characters in the plaintext and the encrypted file size with the combination of RC5 and BBS is lower than the encryption using the standard RC5 algorithm. In their study, Ndruru and Zebua [5] proposed a method based on BBS algorithm with Beaufort encryption in the process of encryption and decryption of pixels in JPG format digital images to ensure the security of digital images. BBS algorithm was used in key generation. According to the results obtained, it was seen that the keys generated by the BBS algorithm are random and do not cause repetition and make it difficult for others to recognize the original image because it increases the color weakness of the original image. Delima and Arroyo [13] used the BBS algorithm to produce a more secure ciphertext by changing the key generation process of traditional Nihilist encryption. According to the simulation results, this proposed hybrid system offers a secure encryption and decryption process. Saini and Sehrawat [14] also aimed to protect sensitive information during transmission and provide efficient and reliable decryption at the receiver side. In their proposed system, they transformed the MNIST dataset into a key generation source and combined it with the power of modern cryptographic methods. An additional layer of security is added to the cryptographic algorithm passing through the first ciphertext by using XOR and BSS. It is shown that this proposed two-layer encryption system exhibits superior performance compared to existing systems such as AES, DES, RSA, and ElGamal and can be applied for various security uses by passing NIST tests with key sizes of 128, 256, and 512 bits. In a study by Najwan [15], three types of pseudo-random number generators were used for image encryption to protect personal images from unauthorized access. These are Linear Feedback Shift Register (LFBSR), Nonlinear Feedback Shift Register (NLFBSR), and BBS. In these algorithms applied to color images, large random keys were applied. When these algorithms were compared, BBS was found to have better performance than LFBSR and NLFBSR. Laia, Zamzami, and Sutarman [16] used the DES algorithm and BBS pseudo-random number generator to generate external keys for encryption and decryption of messages. Thus, they made it difficult to guess the random number. According to the results obtained, it was determined that the combination of DES and BBS successfully encrypts and decrypts messages. In addition, the use of BBS as an external key generator in the DES algorithm did not affect the processing time. Alagaw,

Muhammed, and Geto [17] have carried out a study using a modified BBS algorithm and key flow values to improve the security of the Playfair cipher. They evaluated the performance of the application in Matlab environment in terms of avalanche effect, frequency analysis, key generation, key exchange, and resistance to brute force attack. According to the results obtained, they found that the algorithm they proposed has a good performance. Ardhianto et al. [18] investigated the effect of the Euler number applied to the BBS key generator and extended Vigenere on the utilization of information confidentiality. They used different key lengths of 32 and 64 bits using short astronomical observation speeches sent via 1 Kb telegrams. To measure the performance of the study, they calculated the entropy value of the Extended Vigenere output. According to the results obtained, there is a significant increase in information confidentiality. Sina et al. [19] proposed a method using BBS random number generator and LSB steganography together with 3DES encryption method to secure messages. In this hybrid method, they used triple DES to encrypt messages with .txt extension, BBS random number generator to determine the location of the message to be hidden in the cover image, and LSB steganography method as the hiding method. With 20 hiding operations, they hid a maximum of 150 characters and obtained an average PSNR value of 88.61 after 20 message hiding operations.

As mentioned above, the BBS algorithm has been used many times in different ways. The aim of this study is to add a new one to these methods. The proposed algorithm offers an innovative perspective on the use of the BBS algorithm in the generation of values such as keys for cryptographic applications. In this study, chaotic maps are utilized to achieve this. Chaotic maps are one of the popular topics of recent years. There are many chaotic maps in the literature. This is one of the effective aspects of the proposed work. Thanks to the diversity of these maps, any number of secure random numbers can be obtained with the proposed method. In addition, the proposed approach is simple, comprehensible and at the same time difficult to predict. It is thought that this study will be a source of inspiration for researchers in using the BBS algorithm in different ways. In the rest of this paper, in the second section, the idioms of the BBS algorithm are explained. In the third section, the proposed generator algorithm is presented with a step-by-step flow diagram. In the fourth section, the results of the analyses are given and NIST SP 800-22 tests are applied to the obtained bit sequences. In the fifth section, the results are discussed.

2. Blum Blum Shub Algorithm (BBS)

The BBS algorithm is a frequently used algorithm for generating random numbers. It was designed by Lenore Blum, Manuel Blum, and Michael Shub in 1984 for use in public key infrastructure [20]. BBS is based on the product of two prime numbers and a seed value. In this algorithm, two primes p and q, and a seed value called "seed" are determined. Then, using these values, the desired number of random numbers or bit sequences are obtained in a very simple way as given in Equation 1 and Equation 2.

$$m = p * q \tag{1}$$

$$x = seed^2 \mathrm{mod}(\mathrm{m}) \tag{2}$$

In addition, the flow diagram of the BBS algorithm is given in Table 1. As stated in Table 1, after the prime numbers and seed value are selected, any number of random numbers and ones can be generated.

Table 1. Pseudocode of the BBS algorithm.

```
begin
generate p prime number
generate q prime number
generate "seed" value
x_i = seed
m = p * q
for [0, n]:
        x_{i+1} = x_i^2 mod (m)
        print ( x_{i+1})
        x_i = x_i + 1
end for
end
```

3. Proposed Key Generator Algorithm

In this study, a new key planning algorithm is designed using the BBS algorithm. In this algorithm, the first, two large prime numbers are selected. Then these primes are multiplied to obtain a value of m. Then the numbers are generated according to the formula given in equation 2. This algorithm is quite simple. However, when the

prime numbers and seed value are chosen efficiently, unique numbers can be generated. The main problem here is how to choose the seed value and prime numbers. This point is particularly emphasized in this study. Chaotic maps are used to overcome this problem. The steps of the proposed algorithm are given below.

Step 1. The user is asked to enter a prime number range.
Step 2. A prime number pool is generated in the entered range.
Step 3. With the selected chaotic map, a prime number is randomly selected from the prime number pool (p value) and its position is recorded.
Step 4. With the other selected chaotic map, a prime number is randomly selected from the prime number pool (q value), and its position is recorded.
Step 5. The seed value is obtained by summing the location information of the calculated p and q values. (seed= location p+ location q)
Step 6. Using these values, bit sequences are generated with the BBS algorithm.
Step 7. The generated bit sequences are converted into keys of desired lengths.

As given above, in the proposed method, the user is first asked for the starting and ending values for the prime number pool. All primes between these values entered by the user are generated and stored in an array. In this way, the prime number pool will change every time the algorithm runs. This will provide an important gain, especially in security applications. Because each user will determine the prime number range randomly. This will increase the unpredictability of the algorithm. After the prime number pool is created, p and q values should be selected from this pool. These values should also be unpredictable. Therefore, in this study, it is shown that these values can be selected with chaotic maps. These values can be selected with different chaotic maps or with the same maps. In this study, different chaotic maps are used for p and q values. In this way, two primes are randomly selected from the prime number pool. After the primes are selected, the seed value needs to be determined. One of the most critical parameters of pseudo-random number generators is the seed value. This value must be chosen randomly and cannot be generated again. Because if the seed value is known, the bit sequence can be generated again. In this study, the seed value is taken as the sum of the positions of prime numbers. Since the prime number pool and the number of primes will change each time the algorithm runs, it is very difficult to estimate the seed value used here. The system model of the proposed algorithm is given in Figure 1.

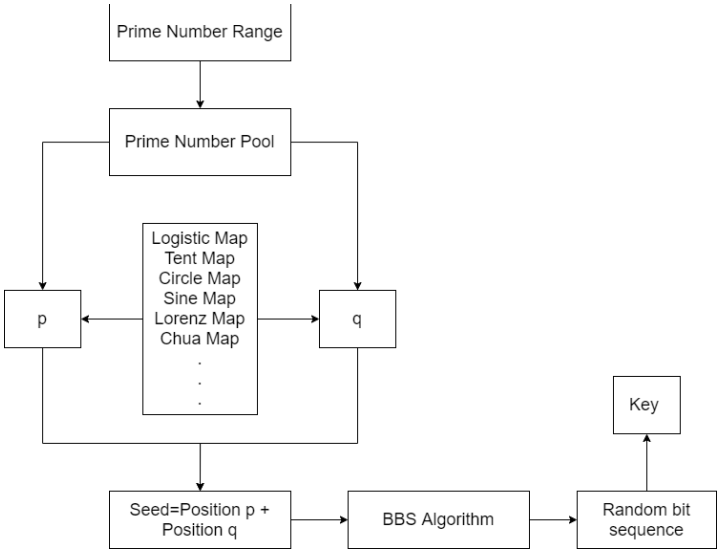

Figure 1. System model of the proposed key planning algorithm.

As seen in Figure 1, different maps can be used to select the values of p and q. Some chaotic maps are given here as examples. However, it is seen that there are more than a hundred chaotic maps in the literature. In addition, studies in which new chaotic maps are obtained are very popular today. One of the most important advantages of this study is the variety of chaotic maps. Thanks to this diversity, the unpredictability of the algorithm will increase. In addition, the seed value is completely random and is determined according to the values entered by the user at that moment. In other words, the algorithm designers and users will not be able to predict the seed value.

4. Analysis Results

The BBS algorithm is simple but effective. It should be noted that the prime numbers and the seed value are chosen randomly. In this study, the problem of selecting these values is solved with chaotic maps. There are intervals where chaotic maps in the literature show random behavior. It is known that the values produced in these intervals are random. With the algorithm proposed in this study, 2 different bit sequences are obtained by using 2 different chaotic maps. The first value, i.e. p, is obtained with the logistic map and the second value, i.e. q, is obtained with the tent map. The mathematical model of the logistic map is given in Equation 3 and the mathematical model of the tent map is given in Equation 4.

$$x_{n+1} = ax_n(1 - x_n)\,, \quad x_n \in [0,1]\,, \quad a \in [3.5, 4] \tag{3}$$

$$x_{n+1} = \begin{cases} ax_n & x_i < 0.5 \\ a(1 - x_n) & x_i \geq 0.5 \end{cases}, \quad x_n \in [0,1]\,, \quad a \in [1,2] \tag{4}$$

The fixed value intervals used in these chaotic maps are the intervals where the map shows random features. These are usually shown by bifurcation diagrams. Bifurcation diagrams of logistic and tent map are given in Figure 2.

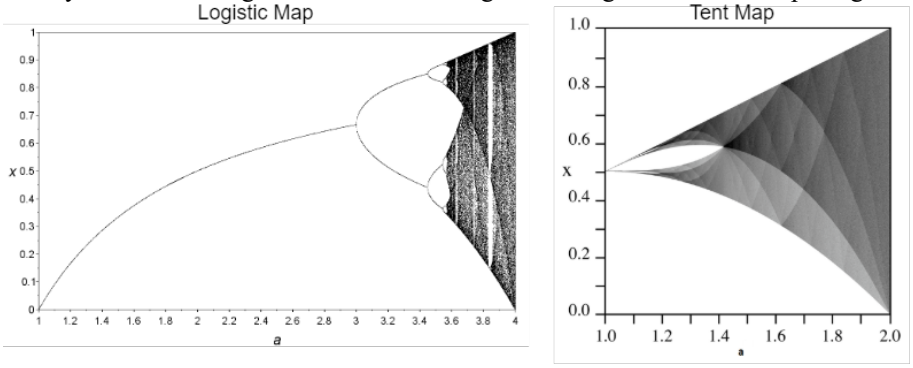

Figure 2. Bifurcation diagrams of logistic and tent maps.

To test the performance of the proposed algorithm, 2 different sequences were generated. The parameters used in the generation of these sequences are given randomly. These parameters are given in Table 2.

Table 2. Values obtained with the proposed algorithm.

Parameters	1. sequence	2. sequence
1. value entered by the user	87814024245	123416789
2. value entered by the user	87814052154	123452345
Total number of prime numbers generated	1061	1946
Number of primes "p" selected by logistic map	87814027403	123433087
Number of primes "q" selected by tent map	87814028789	123419927
Index of the 1. value	124	873
Index of the 2. value	177	181
Seed (index 1+index 2)	301	1054

Using the parameters in Table 2, two different bit sequences of length 1 million were generated. There are various methods to test whether these bit sequences are random or not. The most well-known among these methods is the NIST SP 800-22 test [21]. In this study, the NIST SP 800-22 test was applied to the bit sequences obtained. The NIST SP 800-22 test includes a total of 15 different tests. In each test, a p value is calculated. This p value is expected to be between 0 and 1. As this value approaches 1, the degree of randomness of the sequence increases. The NIST SP 800-22 tests are briefly explained below.

Test 1- Frequency Test (Monobit): The distribution of 1 and 0 values is examined. In other words, the number of 1 and 0 values in the obtained bit sequence is checked. These values are expected to be close to each other. Otherwise, the sequence is not randomly distributed.

Test 2- Frequency Test within a Block: 1 and 0 distributions are examined within various selected blocks. The numbers 1 and 0 are also expected to be close within the blocks.
Test 3- Run Test: The number of 0 and 1 blocks is examined. Here, it is determined whether the number of 0 and 1 streams is random.
Test 4- Longest Run of Ones in a Block: Here the array is first divided into blocks. Then, the number of consecutive 1s and 0s in each block is examined. It is examined whether there is a deviation between the values here and the expected values.
Test 5- Binary Matrix Rank Test: A matrix is obtained with bit blocks, each of which is expressed as a row. Then the rank of this matrix is calculated, and it is observed whether there is a dependency between the blocks.
Test 6- Discrete Fourier Transform (Spectral) Test: Discrete Fourier transform of the bit sequence is taken and its period is observed. The aim here is to detect repeating patterns.
Test 7- Non-Overlapping Template Matching Test: It examines whether a block of a certain length repeats in the sequence. If this block repeats, a new block is searched by shifting one bit from this block.
Test 8- Overlapping Template Matching Test: This test has the same logic as the previous test. The main difference here is that it repositions when a similar pattern is detected.
Test 9- Maurer's Universal Statistical Test: This test examines whether the bitstream can be compressed without data loss. If the bit sequence can be compressed properly enough without data loss, it does not show high randomness.
Test 10- Linear Complexity Test: Measures the complexity of the array by looking at its LFRS (linear feedback shift register) length. If the lengths obtained here are large enough, the array is random.
Test 11- Serial Test: The frequency of overlapping blocks in the bit array is checked. A low number of overlapping blocks is desired here.
Test 12- Approximate Entropy Test: Here the entropy of blocks of length a and a+1 is checked. Frequency is calculated as in the previous test. The frequency calculated here should not be greater than the expected value.
Test 13- Cumulative Sums (Forward) Test: The bit string is divided into blocks of the same length and the balance of 1, 0 values is checked.
Test 14- Cumulative Sums (Reverse) Test: This test is applied to the previous test with the bit string reversed.
Test 15- Random Excursions Test (State: +4): The bit string is divided into blocks of the same length and the balance of 1, 0 values is checked. The excursion is random.

The NIST SP 800-22 test results of the bit sequences generated by the proposed algorithm are given in Table 3. Looking at this table, it is seen that both bit sequences successfully completed all NIST SP 800-22 tests. The point to be considered when determining the parameters in Table 2 is to choose the prime numbers large. In the analysis results, it was seen that the bit sequences could not successfully complete the NIST SP 800-22 tests when the primes were chosen small. However, it is observed that the algorithm works effectively when the primes are at least 9 digits. With the proposed study, key values of desired lengths can be obtained. These keys can be used in existing [22] or newly developed encryption algorithms. In addition, reference [23] can be examined for analyzing the encryption speed of keys of different lengths obtained using the proposed method or different methods.

Table 3. NIST SP 800-22 test results.

PRNG		Sequence1	Sequence2
No	**Test**	**p-value**	**p-value**
1	Frequency Test (Monobit)	0.58919	0.20766
2	Frequency Test within a Block	0.32079	0.11212
3	Runs Test	0.49126	0.71168
4	Longest Run of Ones in a Block	0.43902	0.67749
5	Binary Matrix Rank Test	0.85397	0.17649
6	Discrete Fourier Transform (Spectral) Test	0.02636	0.04446
7	Non-Overlapping Template Matching Test	0.43851	0.71955
8	Overlapping Template Matching Test	0.90420	0.08871
9	Maurer's Universal Statistical test	0.22543	0.52938
10	Linear Complexity Test	0.26034	0.65046
11	Serial test	0.23690	0.32660
12	Approximate Entropy Test	0.01569	0.30734
13	Cumulative Sums (Forward) Test	0.51638	0.35201
14	Cumulative Sums (Reverse) Test	0.75238	0.35137
15	Random Excursions Test (State: +4)	0.56452	0.20341
Number of Successful Tests		15	15

5. Conclusions

In this study, a new key generator architecture is proposed for use in encryption algorithms. Random number generators are one of the most widely used methods to generate keys. This is because keys should not be guessed by attackers and should not be deterministic. BBS algorithm is one of the known effective pseudo-random number generators. In this study, it is considered that it can be used in key generation. BBS is a pseudorandom number generator based on the product of two different large prime numbers and a seed value. This algorithm is generally capable of generating high quality random numbers. However, the parameters to be used need to be chosen carefully. This paper proposes a new approach to overcome this problem.

In the proposed approach, the user is first asked to enter a range. The primes in this range are generated and saved in an array, and a prime number pool is created. Then, prime numbers are randomly selected from this prime number pool with two different chaotic maps to be used in the BBS algorithm. The positions of these selected primes are also recorded. The seed value is taken as the sum of these two positions. Thus, every time the algorithm runs, the prime number range, prime numbers, and seed value will be randomly generated. Even the user will not know the primes and the seed value. This significantly increases the randomness and privacy of these values.

To test the performance of the proposed method, two random intervals are given, and 2 different bit sequences are generated by selecting the values in these intervals. These bit sequences are 1 million long. This is because this number is generally used in the literature. NIST SP 800-22 test suite was applied to the obtained bit sequences. This test suite contains a total of 15 tests and is frequently used. It was observed that the bit sequences obtained with the proposed approach successfully completed all NIST SP 800-22 tests. It should be noted that the prime numbers should be chosen large. When primes larger than 9 digits were chosen, the bit sequences passed all the tests. However, when smaller primes were chosen, not all NIST SP 800-22 tests were passed. Thus, this study provides a roadmap for those who will use the BBS algorithm for key value.

References

[1] Liu J, Wang Y, Han Q, Gao J. A sensitive image encryption algorithm based on a higher-dimensional chaotic map and steganography. Int J Bifurcat Chaos 2022; 32(01), 2250004.

[2] Katz J, Lindell Y. Introduction to modern cryptography: principles and protocols. Chapman and hall/CRC, 2007.

[3] Daemen J, Rijmen V. The design of Rijndael. New York: Springer-verlag 2002; 2.

[4] Van Oorschot PC, Menezes AJ, Vanstone SA. Handbook of applied cryptography. New York: CRC Press, 1996.

[5] Ndruru E, Zebua T. Generate Beaufort Cipher Key Based on Blum-Blum Shub For Secure Digital Image. Instal: Jurnal Komputer 2021; 13(01).

[6] Malik K, Pulikkotil J, Sharma A. Comparison of pseudorandom number generators and their application for uncertainty estimation using Monte Carlo simulation. Mapan 2021; 36(3): 481-496.

[7] Schindler W. Random number generators for cryptographic applications, Cryptographic Engineering. In: Koç ÇK, editör. Springer, Boston, MA, 2009; 5-23.

[8] Artuğer F, Özkaynak F. A new chaotic system and its practical applications in substitution box and random number generator. Multimedia Tools Appl 2024, 1-15.

[9] Joey FL. Modification of Blum-Blum-Shub Generator (BBS) with a 2× 2 Matrix and the First Digit Property of Generated Random Numbers and Bits. AMCI 2023; 12(2): 120-129.

[10] Arroyo JCT, Delima AJP. An Improved Affine Cipher using Blum Blum Shub Algorithm. IJECS) 2020; 9(3): 3295-3298.

[11] Surbakti TB, Fauzi A, Khair H. Rivest Shamir Adleman (RSA) Hybrid Algorithm System and the deep Blum Blum Shub (BBS) Algorithm Securing E-Absence Database Files. INJECSI 2023; 1(2): 53-61.

[12] Rambe BM, Nababan EB, Nasution MK. Performance Analysis Of The Combination Of Blum Blum Shub and Rc5 Algorithm in Message Security. JITE 2024; 7(2): 409-423.

[13] Delima AJP, Arroyo JCT. An Enhanced Nihilist Cipher Using Blum Blum Shub Algorithm. IJATCSE 2020, 9(3): 3270-3174.

[14] Saini A, Sehrawat R. Enhancing Data Security through Machine Learning-based Key Generation and Encryption. ETASR 2024: 14(3): 14148-14154.

[15] Najwan AH. Color Images Encryption using Cipher System with different types of Random Number Generator. IJIRCCE 2017; 5(5).

[16] Laia O, Zamzami EM. Analysis of combination algorithm data encryption standard (DES) and Blum-Blum-Shub (BBS). In Journal of Physics: Conference Series 2021. 5th International Conference on Computing and Applied Informatics (ICCAI 2020), 1-2 December 2020; Medan, Indonesia. pp. 1-7.

[17] Muhammed AJ, Woldiegiworgies TA, Tsegaye GG. Security Enhancement of Playfair Cipher Using Modified Blum Blum Shub Algorithm and Keystream Values. Research Square 2024; 1-12.

[18] Ardhianto E, Redjeki RS, Supriyanto E, Murti H, Wahyudi EN. Adopsi Generator Kunci Euler Number dan Pembangkit Kunci Blum Blum Shub untuk Meningkatkan Confidentiality Level pada Extended Vigenere. Infotek 2024; 7(1): 1-11.

[19] Sina DR, Kiu GA, Djahi BS, Pandie ES. Aplikasi Keamanan Pesan (. Txt) Menggunakan Metode Triple DES Dan Metode Kombinasi LSB Dan BLUM-BLUM-SHUB. J-Icon 2022; 10(2): 204-209.

[20] Omorog CD, Gerardo BD, Medina RP. Enhanced pseudorandom number generator based on Blum-Blum-Shub and elliptic curves. In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) 28-29 April 2018, Penang, Malaysia: IEEE. pp. 269-274.
[21] Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, et. al. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. National Institute of Standarts and Technology, 2010. pp. 23-87.
[22] Etem T, Kaya T. Trivium Algoritması Kaynaklı Rastgele Permutasyon Üretimiyle Görüntü Şifreleme Uygulaması. FÜMBD 2022; 34(2): 687-697.
[23] Etem T, Kaya T. Görüntü Şifreleme için Trivium-Doğrusal Eşlenik Üreteci Tabanlı Bit Üretimi. FÜMBD 2020; 32(1): 287-294.