Applied Mathematics and Computational Intelligence
Volume 12, No. 2, Aug 2023 [120-129]

Modification of Blum-Blum-Shub Generator (BBS) with a 2×2 Matrix and the First Digit Property of Generated Random Numbers and Bits

Farah L. Joey

Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
Department of Computer Techniques Engineering, Al-Esraa University, Baghdad, Iraq

Corresponding author: farahlateefjoey@gmail.com

Received: 7 July 2022
Accepted: 14 July 2023

ABSTRACT

The Blum-Blum Shub algorithm (BBS) that uses quadratic generator produces one output per iteration. For this reason, the proposed paper seeks to improve the efficiency of the BBS by increasing output, that is, longer sequence of random numbers and bits taken per iteration. To do so, this study proposes modification to the quadratic generator of the BBS with a matrix generator by squaring matrix of 2 × 2 which generates more outputs, namely, four outputs per iteration. Subsequently, occurrence-difference of 0s and 1s of these random bits was investigated for both generators. Results show that the occurrence-difference of 0s and 1s of the matrix BBS generator is decreasing as number of iterations are increasing represented by its linear trendline with negative slope. Furthermore, the comparison was made for the first digits of such random numbers for pseudo-oscillation trend per iteration for both generators. Though random numbers obtained from different generators, pseudo-oscillation trends of the first digits for both are considerably quite similar.

Keywords*:* Blum prime, Blum integer, Blum-Blum Shub algorithm, Psesudorandom numbers, Pseudorandom bits, PRNG, PRBG.

1 INTRODUCTION

Random numbers contribute significantly to every aspect of cryptography which are generated from a set containing all possible values such that all the numbers in the set are uniformly distributed [1], [2]. However, it is very difficult to generate true random numbers on computers because computers are designed to be deterministic. For this reason, random numbers are of two forms: true random numbers (TRNs) and pseudo-random numbers (PRNs). True random numbers depend on a physical source of randomness to generate non-regeneratable sequences of random numbers [3]. While pseudorandom numbers require a generator (called pseudorandom number generator (PRNG)) represented by deterministic algorithm or function based on a concept of initial condition called seed to PRNG to produce unique regenerated sequences. Additionally, pseudorandom bit generator (PRBG) is also deterministic generating binary sequence randomly [4, 5].

In 1982, M. Blum explored an application of Blum integers in pseudorandom bit and so these integers were named for him [6]. In following years, PRBG called the Blum Blum Shub (BBS) algorithm was created by L. Blum, M. Blum and M. Shub which uses quadratic generator. To understand the foundations of the BBS algorithm, the definitions of Blum prime number and Blum integer should be illustrated first. Note that Blum prime number is defined by a prime number p with $p \equiv 3(mod\ 4)$. Moreover, a positive n is a Blum integer if it is the product of two distinct Blum prime numbers p and q such that $q \equiv p \equiv 3(mod\ 4)$.. As for the BBS algorithm, its prime is Blum integer denoted by n. This algorithm uses scalar as co-prime x_S and its seed X_0 is defined by $X_0\ \equiv (\ x_S)^2 mod\ n$. Additionally, the quadratic generator of the BBS generates bits based on the output of squaring terms in a sequence producing one output per iteration [7-9].

In this work, the interested study is to modify the BBS so that it could produce more than one output per iteration. The proposed method is replacing the quadratic generator of the BBS with a new generator using the 2×2 matrix. This paper is organized as follows: The first section presents essential concepts related to security of the BBS followed by steps taken to generate the BBS sequence. Next section focuses on modification of the BBS. Subsequently, results and discussion of the study and finally, conclusion.

2 BLUM-BLUM SHUB ALGORITHM

Security of the BBS is dependent on the difficulty of the quadratic residuosity problem as well as the difficulty to factor large numbers n made up of Blum primes [10]. Therefore, this section presents the definition of quadratic residue and theorem concerning on function f used in the BBS and steps taken to generate random bits of the BBS [11].

Definition 1: If there is an integer $0 < x < n$ such that the congruence of $x^2 \equiv r(mod\ n)$ has a solution, then r is said to be quadratic residue $(mod\ n)$.

Note that if the congruence does not have a solution, then r is called quadratic nonresidue.

Theorem 1: Let $n = pq$ be the product of two Blum primes. The function f such that

$$f:\quad QR_n \longrightarrow QR_n$$

$$x \longrightarrow x^2(mod\ n).$$

The function f described by Theorem 1 is that f is a permutation defined from a set to the same set. Moreover, f is a bijection since each quadratic residue has exactly one square root which is also a quadratic residue.

The BBS generator takes the parity bit of each term in the generated sequence where random bits are created by:

1. Choose two distinct Blum prime numbers p and q for $n \ni n = p \times q$.
2. Choose a scalar x_S as co-prime to n randomly.
3. Define a scalar seed $X_{0\ S}$ as $X_{0\ S} = (x_S)^2 mod\ n$.
4. Generate the first iteration $X_{1\ S} = (X_{0\ S})^2 mod\ n$.
5. Convert $X_{1\ S}$ to $Z_{1\ \mathrm{S}}$ such that parity($X_{1\ S}$) $= X_{1\ S}\ mod\ 2$.
6. Create binary sequence as output of $Z_{1\ S}$ as $\{X_{1\ S}\ mod\ 2\}$.

Next is Figure 1 that shows a block diagram of Blum-Blum Shub algorithm.

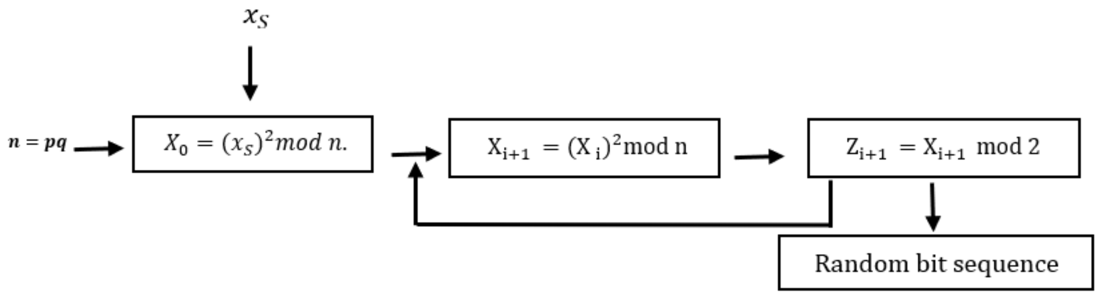
x_S
$n = pq$
$X_0 = (x_S)^2 mod\ n.$
$\mathrm{X_{i+1} = (X_i)^2 mod\ n}$
$\mathrm{Z_{i+1} = X_{i+1}\ mod\ 2}$
Random bit sequence

Figure 1: Block diagram of Blum-Blum-Shub algorithm that describes steps taken to generate the BBS sequence.

Example:

Step 1 Let $\mathrm{p = 3, q = 7, n = pq = 21}$.

Step 2: Let $x_S = 100$.

Step 3: Scalar seed $X_{0\,\mathrm{S}}$ is $X_{0\,S=}(x_S)^2 mod\ n$: $X_{0\,S=}(100)^2 mod\ 21 = 4$.

Step 4: The first iteration is $X_{1\,S=}(X_{0\,S})^2 mod\ n$: $X_{1\,S=}(4)^2 mod\ 21 = 16$.

Step 5: Evaluate $Z_{1\,S}$ such that $Z_{1\,S}$ =party $(X_{1S}) = X_{1\,S} mod\ 2$: $16\ mod\ 2 = 0$.

Step 6: Binary sequence: $\{0\}$.

3 METHODOLOGY

This study proposes modification to the BBS algorithm replacing its quadratic generator with a matrix M of 2×2 denoted by x_M and its seed $S_{0\,M}$ is defined by $S_{0\,M} = (x_M)^2 mod\ n$. This new generator is called 2×2 matrix generator. Subsequently, sequence of random bits is created by:

1. Choose two distinct Blum prime numbers p and q for n such that $n = p \times q$.
2. Choose a matrix M of 2×2 as co-prime to n randomly with a condition of $\gcd(a_{ij}, n) = 1$.
3. Define a matrix seed $S_{0\,M}$ as $S_{0\,M} = (x_M)^2 mod\ n$.
4. Generate the first iteration $S_{1\,M} = (S_{0\,M})^2 mod\ n$.
5. Convert $S_{1\,M}$ to $Z_{1\,M}$ such that parity $(S_{1\,M}) = S_{1\,M}\ mod\ 2$.
6. Create binary sequence according to output of $Z_{1\,M}$ as $\{S_{1\,M}\ mod\ 2\}$.

Block diagram of the modified BBS algorithm using the 2×2 matrix generator is shown in Figure 2 followed by an example.

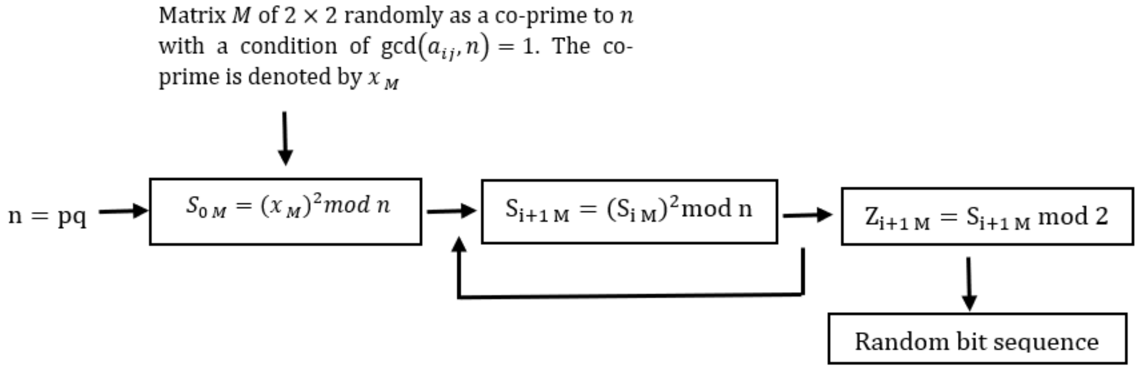
Matrix M of 2×2 randomly as a co-prime to n with a condition of $\gcd(a_{ij}, n) = 1$. The co-prime is denoted by x_M
$\mathrm{n = pq}$
$S_{0\,M} = (x_M)^2 mod\ n$
$\mathrm{S_{i+1\,M} = (S_{i\,M})^2 mod\ n}$
$\mathrm{Z_{i+1\,M} = S_{i+1\,M}\ mod\ 2}$
Random bit sequence

Figure 2: Block diagram of the modified BBS algorithm using the 2 × 2 matrix generator.

Example:

Step 1: Let $p = 3, q = 7, n = pq = 21$.

Step 2: Let a matrix M be $x_M = \begin{pmatrix} 13 & 20 \\ 11 & 17 \end{pmatrix}$.

Step 3: Matrix M seed is $S_{0\,M} = (x_M)^2 mod\ n$:

$$S_{0\,M} = \begin{pmatrix} 13 & 20 \\ 11 & 17 \end{pmatrix}^2 mod\ 21 = \begin{pmatrix} 389 & 600 \\ 330 & 509 \end{pmatrix} mod\ 21 = \begin{pmatrix} 389\ mod\ 21 & 600\ mod\ 21 \\ 330\ mod\ 21 & 509\ mod\ 21 \end{pmatrix} = \begin{pmatrix} 11 & 12 \\ 15 & 5 \end{pmatrix}.$$

Step 4: The first iteration is $S_{1\,M} = \ (S_{0\,M})^2 mod\ n$:

$$S_{1\,M} = (S_{0\,M})^2 mod\ 21 = \begin{pmatrix} 11 & 12 \\ 15 & 17 \end{pmatrix}^2 mod\ 21 = \begin{pmatrix} 301 & 192 \\ 240 & 205 \end{pmatrix} mod\ 21 = \begin{pmatrix} 7 & 3 \\ 9 & 16 \end{pmatrix}.$$

Step 5: Evaluate $Z_{1\,M}$ such that $Z_{1\,M}$ = parity $(S_{1\,M}) = S_{1\,M}\ mod\ 2$:

$Z_{1\,M}$ = parity $(S_{1\,M}) = S_{1\,M}\ mod\ 2$: $\begin{pmatrix} 7\ mod\ 2 & 3\ mod\ 2 \\ 9\ mod\ 2 & 16\ mod\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.

Step 6: Binary sequence: {1 1 1 0}.

4 RESULTS AND DISCUSSION

Firstly, the results illustrate the comparison of occurrence of 0s and 1s in the bit sequence obtained from the quadratic BBS generator and the 2 × 2 matrix BBS generator per iterations. Second step is comparing occurrences of the first digit of random numbers in the generated sequence per iterations [12, 13]. Subsequently, the next step is comparing such random numbers based on the Benford's law per iterations [14, 15].

4.1 Occurrence-Difference of 0s and 1s in the Bit Sequence

One of the important requirements for the PRNG is that occurrence of 0s and 1s in the long output bit sequence should be almost the same as mentioned in [16]. In this context, this means that the PRNG is considered to perform better when the difference between occurrence of 0s and 1s is closer to 0. For this reason, this study is interested to compare occurrence of 0s and 1s in the output bit sequence obtained from the quadratic BBS and the 2×2 matrix BBS per 500, 600, 650 and 700 iterations as shown in Table 1 and Figure 3. Next is Figure 4 showing comparison of difference between occurrence of 0s and 1s in the output bit sequence generated by the quadratic BBS and the 2×2 matrix BBS per iterations.

Table 1: Comparison of occurrence of 0s and 1s in output bit sequence generated by the quadratic BBS and the 2 × 2 matrix BBS per 500, 600, 650 and 700 iterations.

Iterations	**0s and 1s generated by the quadratic BBS**		**Difference**	**0s and 1s generated by the 2x2 matrix BBS**		**Difference**
	0	**1**		**0**	**1**	
500	267	233	34	848	1152	304
600	295	305	10	1237	1163	74
650	330	320	10	1246	1354	108
700	375	325	50	1391	1409	18

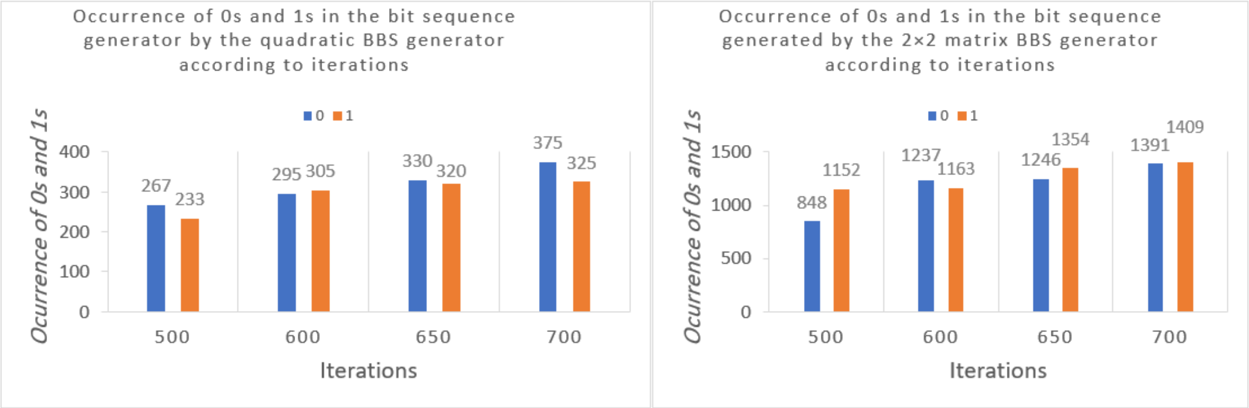

Figure 3: Comparison of occurrence of 0s and 1s in the output bit sequence generated by the quadratic BBS (left) and the 2 × 2 matrix BBS (right) per 500, 600, 650 and 700 iterations.

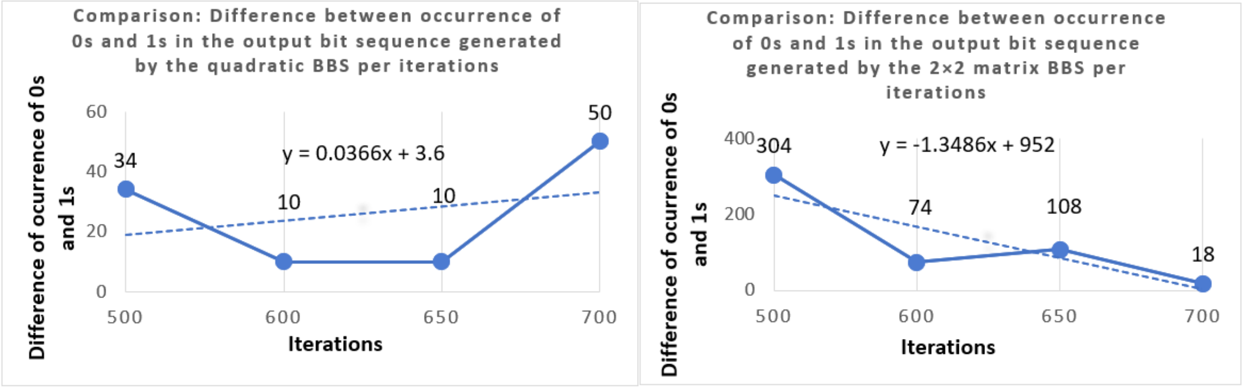

Figure 4: Comparison of difference between occurrence of 0s and 1s in the output bit sequence generated by the quadratic BBS (left) and the 2 × 2 matrix BBS (right) per 500, 600, 650 and 700 iterations.

From Table 1, Figures 3 and Figure 4, 600 and 650 iterations produced by the quadratic BBS show minimum difference between occurrence of 0s and 1s in the output bit sequence and its maximum difference happened during 700 iterations. Additionally, the linear trendline of the quadratic BBS about the difference has positive slope. While the minimum difference happened during 700 iterations for the 2 × 2 matrix BBS and its 500 iterations resulted in maximum difference. It is observed that the 2 × 2 matrix BBS performs such that the difference is decreasing as number of iterations are increasing and so its linear trendline has negative slope.

4.2 Pseudo-Oscillation Trend of the First Digit of Random Numbers

The underlying reason for this section is to exhibit an internal property of coherency of generated random numbers which is the mechanism of generator-development. The results show the occurrence of the first digit of each random number generated by the quadratic BBS and the 2 × 2 matrix per 500, 600, 650 and 700 iterations.

Though sequences of random numbers were generated by different generators, plots concerning the occurrence of the first digit of these numbers exhibit a pseudo-oscillation. Based on Figure 5, the first digit 1 occurs maximally for both generators followed by other first-digits that oscillate between 17-50 and between 99-265 for the quadratic BBS and the 2 × 2 matrix BBS, respectively. The 2 × 2 matrix BBS generated an oscillation of the first digits that appears to be sharper compared to the first digits produced by the quadratic BBS.

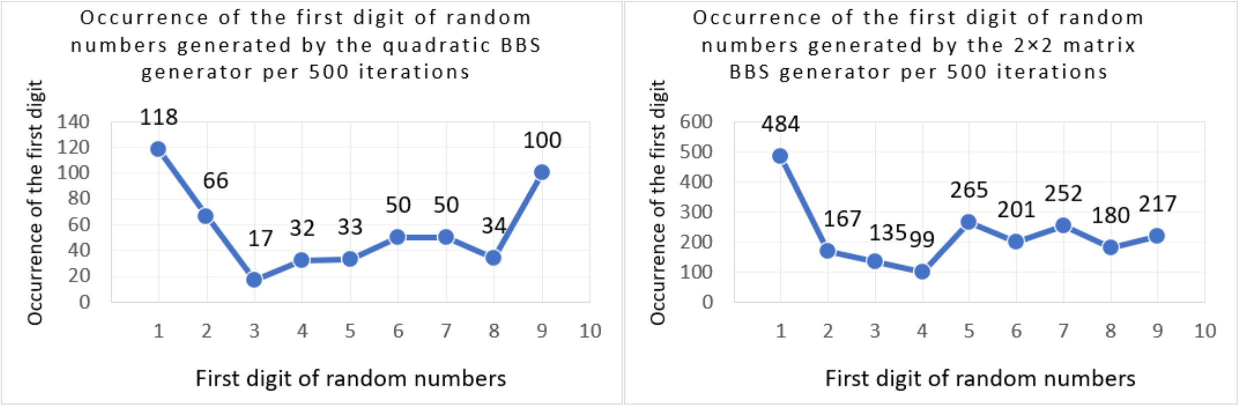

Figure 5: Comparison of occurrence of the first digit of random numbers generated by the quadratic BBS (left) and the 2x2 matrix (right) per 500 iterations.

Next is Figure 6 showing plots for 600 iterations of the first digits obtained from both generators resulted in an extremely small oscillation between the first digits that nearly appears as a flat line while the first digit 1 still occurred greatly.

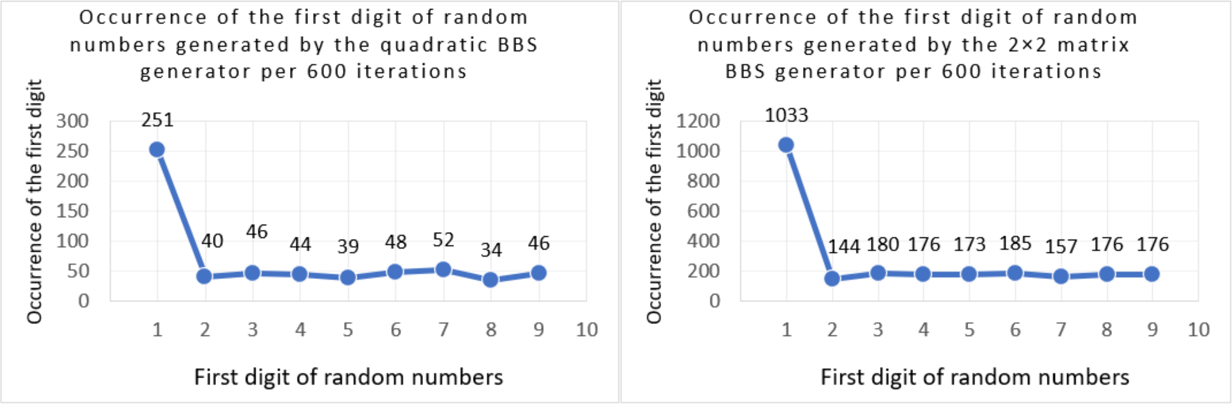

Figure 6: Comparison of occurrence of the first digit of random numbers generated by the quadratic BBS (left) and the 2x2 matrix (right) per 600 iterations.

Results of occurrence of the first digit of random numbers per 650 iterations produced by both generators are shown in Figure 7. The maximum occurrence is presented by the first digit 1 subsequently followed by a slight greater oscillation between other first digits compared to plots in Figure 6.

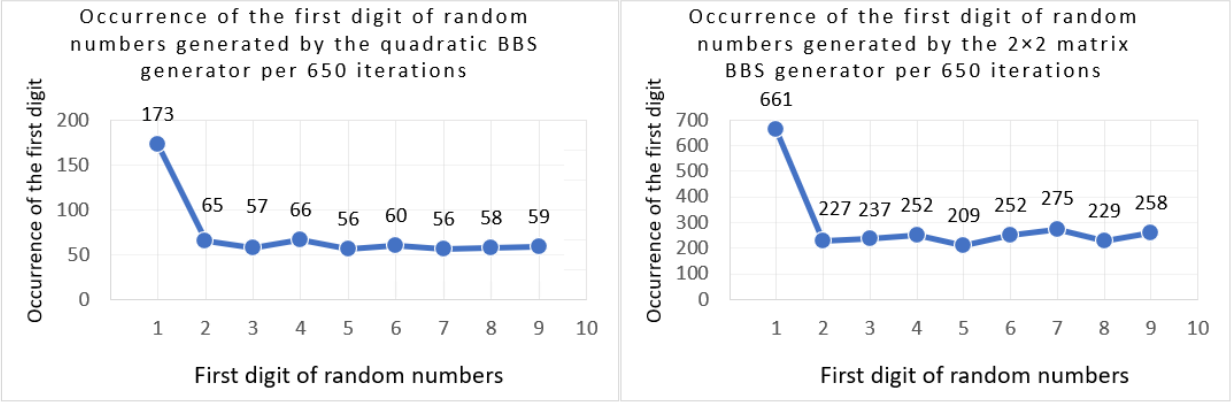

Figure 7: Comparison of occurrence of the first digit of random numbers generated by the quadratic BBS (left) and the 2x2 matrix (right) per 650 iterations.

In Figure 8, plots of occurrence of the first digit obtained from random numbers generated by both generators per 700 iterations show that maximum occurrence of the first digit is 2 followed by obvious oscillation between other first digits.

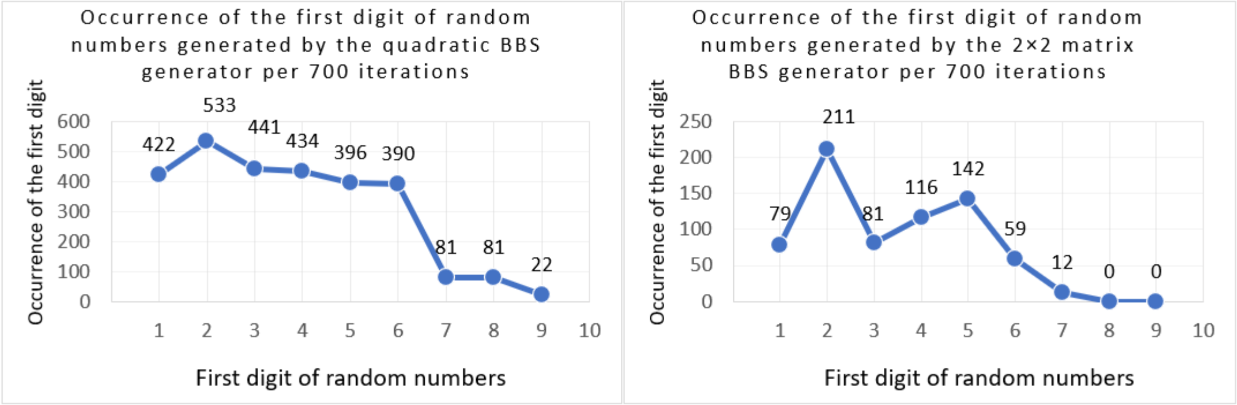

Figure 8: Comparison of occurrence of the first digit of random numbers generated by the quadratic BBS (left) and the 2x2 matrix (right) per 700 iterations.

5 CONCLUSION

In this approach, the modification is made to the quadratic BBS generator using the 2 × 2 matrix that is by squaring the matrix which increases output per iteration. The matrix BBS performs in such a way that number of iterations taken by it is inversely proportional to occurrence-difference of 0s and 1s obtained from the generated binary sequence. In addition, the matrix generator performs just as well as the original where pseudo-oscillation trends of the first digits of the generated random numbers are considerably quite similar with the original.

ACKNOWLEDGEMENT

The author would like to thank Universiti Putra Malaysia for the support in writing this manuscript.

REFERENCES

[1] K. Landsman, “Randomness? What randomness?”, *Found. Phys.*, vol. 50, no. 2, p.p. 61–104, 2020, doi:10.1007/s10701-020-00318-8.

[2] C. D. Omorog, B. D. Gerardo and R. P. Medina, "Enhanced pseudorandom number generator based on Blum-Blum- Shub and elliptic curves”*, IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE),* pp. 269-274, 2018, doi:10.1109/ISCAIE.2018.8405483.

[3] O. Laia, E. M. Zamzami and Sutarman, “Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS)”, *J. Phys.: Conf. Ser.* 1898 012017, 2021, doi:10.1088/1742-6596/1898/1/012017

[4] W. Stallings, ”Cryptography and Network security principles and practices”, Prentice Hall, Fifth Edition, 2011.

[5] L. Pasqualini and M. Parton,” Pseudo Random Number Generation: a Reinforcement Learning approach”, *Procedia Computer Science,* vol. 170, p.p. 1122–1127, 2020.

[6] L. Blum, M. Blum, and M. Shub, “Comparison of Two Pseudo-Random Number Generators”, *In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds) Advances in Cryptology. Springer, Boston, MA.,* 1983, doi: 10.1007/978-1-4757-0602-4_6.

[7] A. Sidorenko and B. Schoenmakers, “Concrete security of the Blum-Blum-Shub pseudo-random generator” *in: Lecture notes in computer science, Cryptography and Coding, Springer-Verlag, Berlin,* vol. 3796, pp. 355–375, 2005, doi:10.1007/11586821_24.

[8] L. Blum, M. Blum and M. Shub, “A simple unpredictable pseudo-random number generator”, *SIAM Journal on Computing*, vol.15, no. 2, 364-383, 1986, doi:10.1137/0215025. ISSN 0097-5397.

[9] E. Kadhim, U. Hussein, and S. Hadi, “AES Cryptography Algorithm Based on Intelligent Blum-Blum-Shub PRNGs”, *Journal of Engineering and Applied Sciences,* vol. 12 no. 10, pp. 9035-9040, 2017.

[10] P. Junod, “Cryptographic secure pseudo-random bits generation: the Blum-Blum-Shub generator” 1999. http://crypto.junod.info/bbs.pdf (accessed 22.11.2016).

[11] B. David,”Introduction to cryptography with java applets”, Grinnell College, London, 2003.

[12] R.A. Raimi, “The first digit problem”, *Am. Math. Mon*., vol. 83, no. 7, pp. 521-538, 1976.

[13] R.A. Raimi, “The peculiar distribution of first digits”, *Sci. Am*., vol. 221, pp. 109–121, 1969.

[14] F. Benford, “The law of anomalous numbers”, *Proceedings of the American Philosophical Society,* 1938, vol. 78, no. 4, p.p. 551–572.

[15] L. Sun, T. S. Anthony, H. Z. Xia, J. Chen, X. Huang and Y. Zhang, "Detection and classification of malicious patterns in network traffic using Benford's law," *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia*, 2017, pp. 864-872, doi: 10.1109/APSIPA.2017.8282154.

[16] Y.D. Vybornova, “Implementation of Blum-Blum-Shub generator and study of its key features”,*IN SITU*. Vol. 4, no. 4, pp. 36-38, 2015.