Modified Blum-Blum-Shub Generator for Generation of Pseudo-random Number

Farah L. Joey[1, 2], *Wah June Leong[2] C. Y. Chen[2] M. L. Othman[3]

[1]Department of Mathematics and Computer Applications, College of Sciences,Al-Nahrain University, Jadriya, Baghdad, Iraq.

[2]Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

[3]Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Abstract

Pseudo-random number generators (PRNGs) are considered as mathematical algorithms, that produce sequences of random numbers. They can be used in many applications such as operation research, statistics and cryptography which require random numbers. For this purpose, modifications have been applied to Blum-Blum-Shub(BBS) PRNG to increase the bit-sequence while lower number of iterations is required to achieve better randomness. Besides that, randomness hypothesis test is performed to study the randomness behavior, and our results indicate that all algorithms have fulfilled the randomness test except the cubic BBS algorithm.

Keywords: Blum prime, Blum integer, Blum-Blum-Shub algorithm, Randomness hypothesis test.

1 Introduction

Random numbers play a vital role for example, in operation research, statistics and cryptography which are generated as unpredictable valued numbers. They are generated from two kinds of generators: First,random number generators (RNGs) are generated from uncertain process that its output represents as random. Therefore, it needs some process to remove the noise for clarifying the results. Second, pseudo-random number generators (PRNGs) aregenerated as a deterministic process whichneeds unpredictable random seed and some well-known mathematical formula[1]. In simulation and computer programing, PRNGs are commonly used because they are deterministic and fast in implementation so that in cryptography, the outputs of PRNG's would be used as a key-encryption for encoding and decoding. [2,3,4]. Therefore, random numbers are values arranged as a sequence manner while the bit random sequence consists of {0, 1} binary digits.Subsequently, generated random numbers subject to randomness metric to investigate the randomness behavior such as the randomness hypothesis test.

In 1982, M. Blum discovered a method of finding an integer in pseudo-random bit and called Blum integer [5]. Following that in 1986, Blum et al. [8] proposed pseudo random number generator (PRNG) called Blum-Blum-Shub (BBS) that produces unpredictable sequences of random numbers which uses quadratic generator. Blum-Blum Shub (BBS) algorithm is considered one of the most popular random number

generators for secured encryption with high security because of the large prime factorizing number (n). Though it is a simple algorithm, it requires large numbers to be secured [6]. Moreover, Blum prime number is defined by a prime number p with $p \equiv 3(mod\ 4)$ and a positive integer$n$ is a Blum integer denoted by the product of two distinct Blum prime numbers p and q such that $q \equiv p \equiv 3(mod\ 4)$. So that the BBS algorithm uses Blum integer denoted by n, anditusesa scalar $x_S$as co-prime to n and its seed X_0 is defined by $X_0 \equiv (x_S)^2 modn$. Additionally, the quadratic generator of the BBS$X_{i+1} \equiv (X_i)^2 modn$ generates the output of random numbers converting to bit sequence [7, 8, 9]. Joey [10] modified the BBS algorithm by replacing its quadratic generator with a 2×2matrix, M 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 a2×2 matrix generator, which could produce four times more further one output periteration.

In this work, modifications on the BBS algorithmsuch as cubic, quartic, arithmetic sequence, 3×3 matrix andmodified cubic and quadratic functionalgorithms' generators are considered. This paper is organized as follows: section 1 presents the basic concept ofBBS algorithm concept. In section 2 the randomness hypothesis test is described. Then wefocus on modifying the algorithms andit follows by applying therandomness hypothesis test on the randomness of the modified BBS algorithms. Subsequently, results and discussion of the study are in section 5 and finally, conclusion is made on overall finding of the study.

2 Blum-Blum Shub Algorithm

BBS algorithm is one of the pseudorandom number generators PRNG that appeared in 1986[11] depending on the concept of number theory. The BBS generator is based on the difficulty of quadratic residue and the large number of modulo n which is calculated by choosing two Blum prime numbers$p$ and q such that$n = p \times q$. In this section, the definition and theorem regarding to the function f employed in BBS algorithm and the process for generating random bit sequences of the BBS.

Definition 1 [12]: If there is an integer $0 < x < n$ such that the congruence of $x^2 \equiv r(modn)$ has a solution, then r is said to be quadratic residue $(modn)$.

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

Theorem 1 [12]: Let $n = pq$be the product of two Blum primes. The function f such that

$$f: \quad QR_n \longrightarrow QR_n$$
$$x \longrightarrow x^2(modn).$$

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.

Next, the parity bit sequence has been converted from the random numbers generated by BBS for each iteration, and the following steps and Figure 1 illustrates the BBS algorithm:

Choose two distinct Blum prime numbers p and q for $n \ni n = p \times q$.

1. Choose a scalar x_S as co-prime to n randomly.
2. Define a scalar seed $X_{0\,S}$ as $X_{0\,S} = (x_S)^2 modn$.
3. Generate the first iteration $X_{1\,S} = (X_{0\,S})^2 modn$.
4. Convert $X_{1\,S}$ to $Z_{1\,S}$ such that parity$(X_{1\,S}) = X_{1\,S} mod\ 2$.
5. Create binary sequence as output of $Z_{1\,S}$ as $\{X_{1\,S} mod\ 2\}$.

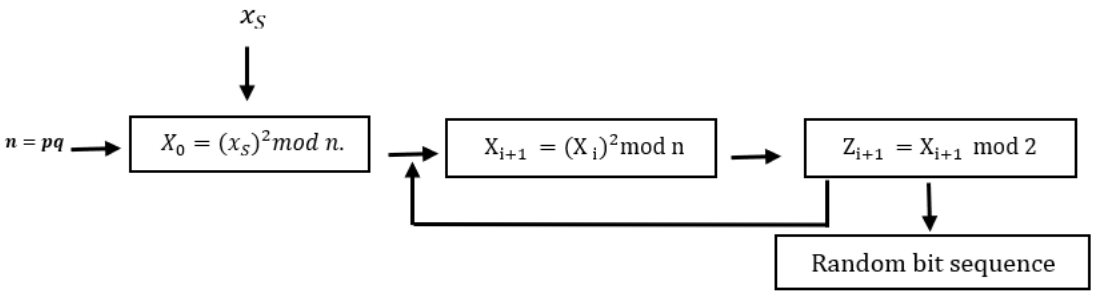

Figure 1: Original BBS

3 Randomness Hypothesis Test

The hypothesis test is a method to investigate whether the sequence of data is fulfilling the randomness observation or not. This randomness hypothesis test is depending on several steps [1,3]:
Step 1:Suppose the following hypotheses:
H_0 : The sequence of data is fulfilling the randomness.
H_1 : The sequence of data is rejecting the randomness.
Step 2:Calculate the number of runs where run is representing the number of observations that occur with the same pattern, for example, if the Head and Tail are the observations of a random sequence as:

H T H H T H T TT H T T H HH T TTTT H HH
The number of runs is (11) is determined as Table 1:

H	T	HH	T	H	TTT	H	TT	HHH	TTTTT	HHH
1	2	3	4	5	6	7	8	9	10	11

Table 1: The number of runs

Step 3: Determine the samples as the number of H as N1= 11, and the number of T as N2=12 so that if both or one of the number of samples are less than or equal to 20 then the test evaluation is the runs number and makes the decision according to the critical value table. On the other hand, if the number of samples N1 or N2 are larger than 20, the statistical technique will be taken.
Significance levelα**:** It is a probability of the error level that gives an indication whether the bit-stream is random or not. Sometimes, it gives us a rejection to accept the randomness even if the random generator has a good performance. Besides that, it sets before applying the test and its value often takes between ($\alpha = 0.01$, or$\alpha = 0.05$) [13].
p-value: It is a probability of giving a well-confidence indication for testing the randomness compared with the selected significance level (α). It means that if the p-value $\geq \alpha$ the randomness will be accepted, otherwise, the bit-sequence will be nonrandom if the p-value $< \alpha$ [13].
The statistical technique under consideration is relying on calculate the Z-score of the normal distribution by$Z = \frac{G-\mu_G}{\delta_G}$, where G is the number of runs, μ_G is number expectation calculated by $\frac{2N_1N_2}{N} + 1$, and $\delta_G$is the standard deviation calculated by $\sqrt{\frac{2N_1N_2(2N_1N_2-N)}{N^2(N-1)}}$.

After that, the critical value should be applied by $\mp Z_{\frac{\propto}{2}} = \mp 1.96$ with respect to the normal distribution by setting the significance level as $\alpha = 0.05$.

4 Methodology

In this study, several modifications have been developed from BBS algorithm:

1. Cubic BBS algorithm, which is demonstrated by the following steps:

a) Choose two distinct Blum prime numbers p and q for n such that $n = p \times q$.
b) Choose a scalar x as co-prime to n randomly.
c) Define a scalar seed X_0 as $X_0 = (x)^3 mod n$.
d) Generate the first iteration $X_1 = (X_0)^3 mod n$.
e) Convert X_1 to Z_1 such that parity(X_1) $= X_1 mod\ 2$.
f) Create binary sequence as output of Z_1 as $\{X_1 mod\ 2\}$

Figure 2 illustrates the algorithm:

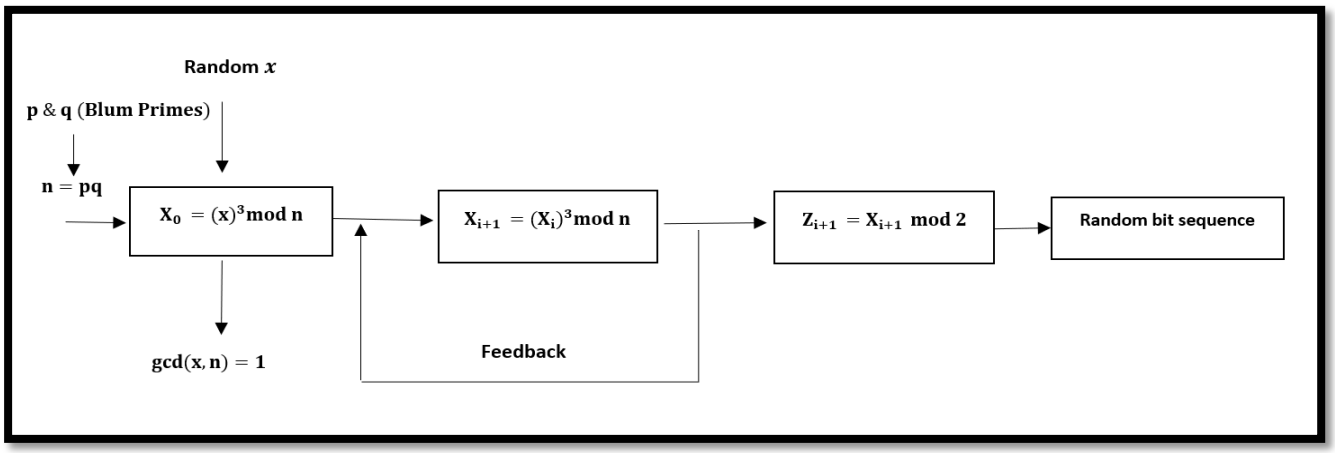

Figure 2: Cubic BBS

2. Quartic BBS algorithm is highlighted by the following steps:

a) Choose two distinct Blum prime numbers p and q for n such that $N = p \times q$.
b) Choose a scalar x as co-prime to N randomly.
c) Define a scalar seed X_1 as $X_0 = (x\ )^4 mod N$.
d) Generate the first iteration $X_1 = (X_0)^4 mod N$.
e) Convert X_1 to Z_1 such that parity(X_1) $= X_1 mod\ 2$.
f) Create binary sequence as output of Z_1 as $\{X_1 mod\ 2\}$.

Implementation of the modified algorithm is given in Figure 3:

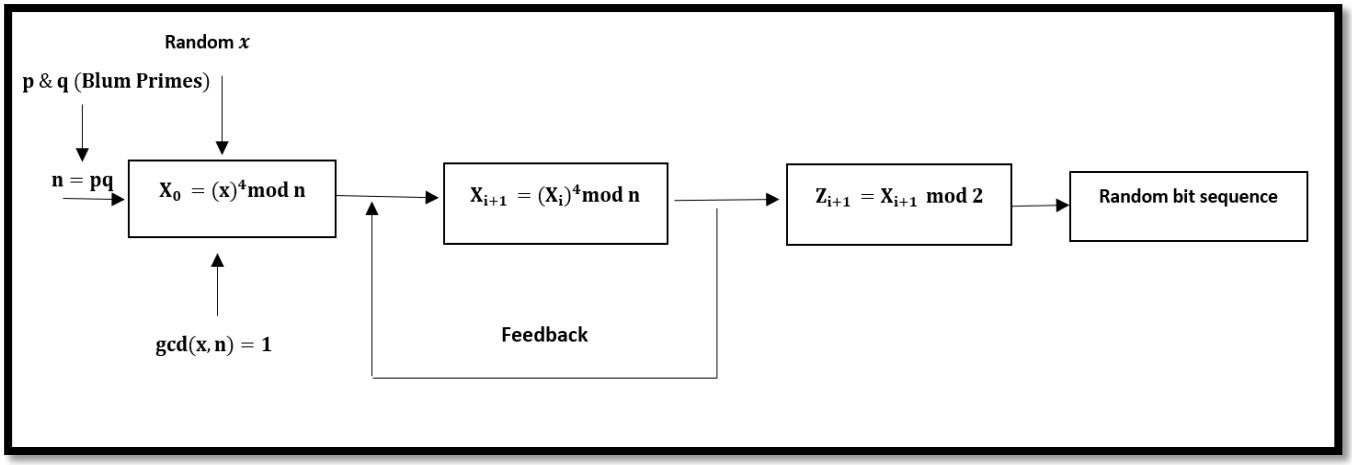

Figure 3: Quartic BBS

3. Sequence BBS algorithm,where the modification is based on arithmetic sequence is stated as below:

a) Choose two distinct Blum prime numbers p and q for n such that $N = p \times q$.

b) Choose a scalar x as co-prime to n with $0 < x < \frac{N}{2}$.

c) Create $Seq_k(x) = x + (k-1)\text{d}$, with $0 < k < \frac{x}{d}$ and $0 < d <$ *the smaller Blum prime* .

d) Define a scalar seed $S_{0\,Seq_k}$ as $S_{0\,Seq_k} = \{Seq_k(x)\}^2 mod N$.

e) Generate the first iteration $S_{1\ Seq_k} = \{S_{0Seq_k}\}^2 mod N$.

f) Convert $S_{1\ Seq_k}$ to $Z_{1\,Seq_k}$ such that parity($S_{1\ Seq_k}$) $= Z_{1\,Seq_k} mod\ 2$.

g) Create binary sequence as output of $Z_{1\,Seq_k}$ as $\{S_{1\ Seq_k} mod\ 2\}$.

Figure 4 summarizes the algorithm:

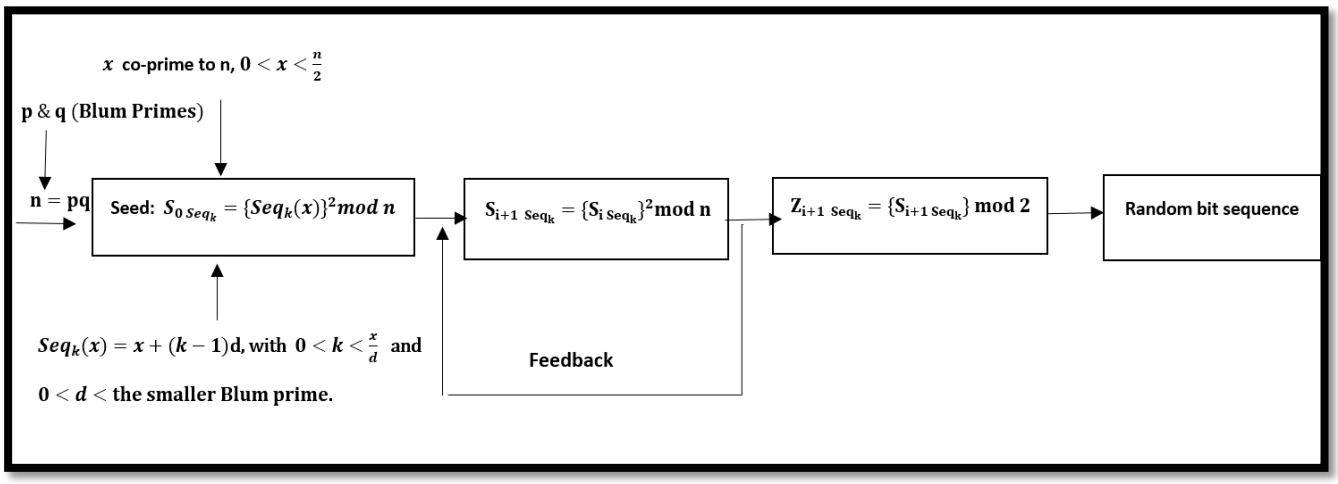

Figure 4: Sequence BBS

4. 3×3 matrix BBS algorithm, is an extension of 2×2 matrix BBS algorithm 10]:

a) Choose two distinct Blum prime numbers p and q for n such that $N = p \times q$.

b) Choose a matrix M of 3×3 as co-prime to n randomly with a condition of $\gcd(a_{ij}, N) = 1$.

c) Define a matrix seed $S_{0\,M}$ as $S_{0\,M} = (x_M)^2 mod\ N$.

d) Generate the first iteration $S_{1\,M} = (S_{0\,M})^2 mod\ N$.

e) Convert $S_{1\,M}$ to $Z_{1\,M}$ such that parity $(S_{1\,M}) = S_{1\,M}\ mod\ 2$.

f) Create binary sequence according to output of $Z_{1\,M}$ as $\{S_{1\,M}\ mod\ 2\}$.

The algorithm is illustrated by Figure 5:

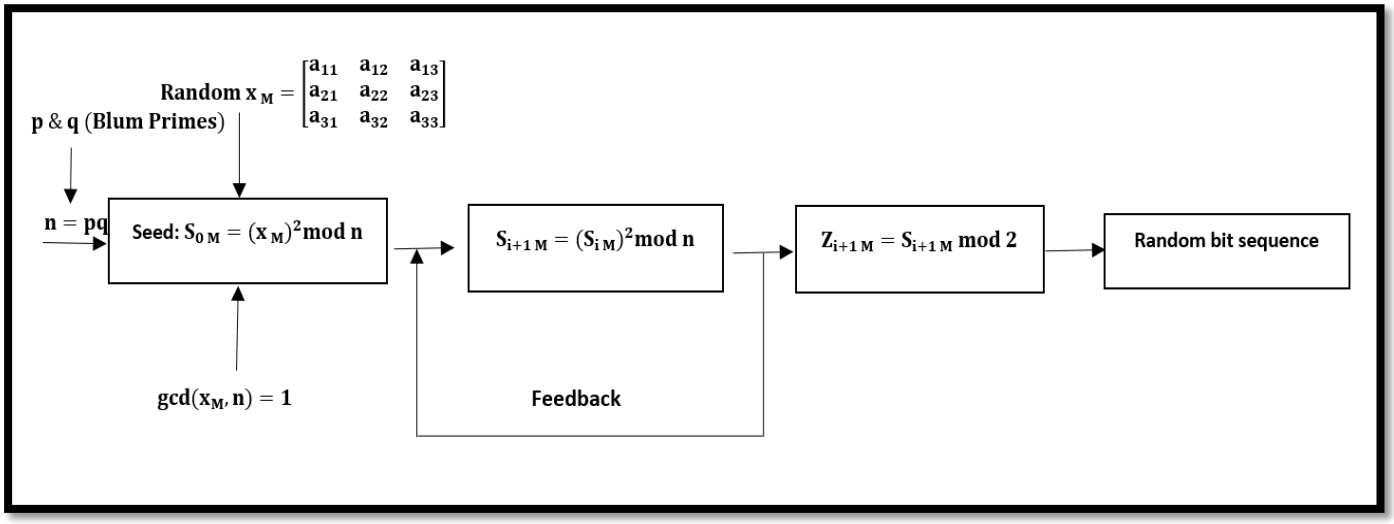

Figure 5: 3×3 matrixBBS

To understand the modified algorithm better, we give the following example on how a 3×3 matrix modified BBS algorithm can be constructed:

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

Step 2: Let a matrix M be $x_M = \begin{pmatrix} 13 & 20 & 11 \\ 17 & 5 & 10 \\ 16 & 19 & 8 \end{pmatrix}$.

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

$$S_{0\,M} = \begin{pmatrix} 13 & 20 & 11 \\ 17 & 5 & 10 \\ 16 & 19 & 8 \end{pmatrix}^2 mod\ 21 = \begin{pmatrix} 685 & 569 & 431 \\ 466 & 555 & 317 \\ 659 & 567 & 430 \end{pmatrix} mod\ 21$$

$$= \begin{pmatrix} 685\ mod 21 & 569\ mod 21 & 431\ mod 21 \\ 466\ mod 21 & 555\ mod 21 & 317\ mod 21 \\ 659\ mod 21 & 567\ mod 21 & 430\ mod 21 \end{pmatrix} = \begin{pmatrix} 13 & 2 & 11 \\ 4 & 9 & 2 \\ 8 & 0 & 10 \end{pmatrix}$$

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

$$S_{1\,M} = (S_{0\,M})^2 mod\ 21 = \begin{pmatrix} 13 & 2 & 11 \\ 4 & 9 & 2 \\ 8 & 0 & 10 \end{pmatrix}^2 mod\ 21 = \begin{pmatrix} 265 & 44 & 257 \\ 104 & 89 & 82 \\ 184 & 16 & 188 \end{pmatrix} mod\ 21$$

$$= \begin{pmatrix} 265\ mod 21 & 44\ mod 21 & 257\ mod 21 \\ 104\ mod 21 & 89\ mod 21 & 82\ mod 21 \\ 184\ mod 21 & 16\ mod 21 & 188\ mod 21 \end{pmatrix} = \begin{pmatrix} 13 & 2 & 5 \\ 20 & 5 & 19 \\ 16 & 16 & 20 \end{pmatrix}.$$

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

$Z_{1\,M}$ =parity $(S_{1\,M}) = S_{1\,M}\ mod\ 2$: $\begin{pmatrix} 13\ mod\ 2 & 2\ mod\ 2 & 5\ mod\ 2 \\ 20\ mod\ 2 & 5\ mod\ 2 & 19\ mod\ 2 \\ 16\ mod\ 2 & 16\ mod\ 2 & 20\ mod\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$.

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

5. Finally, the following two modifications are based on modified cubic and quadratic function with an embedded parameter:

I. Parameter modified cubic BBS:

a) Choose two distinct Blum prime numbers p and q for n such that $N = p \times q$.
b) Choose a scalar X_0 any random seed with $gcd(X_0, N) = 1$.
c) Define D as $D = {X_0}^3 mod g$ where$g = lcm(p-1, q-1)$.
d) Generate the first iteration$X_1 = (X_0 + D)^3 mod N$.
e) Convert X_1 to Z_1 such that parity(X_1) $= Z_1 mod\ 2$.
f) Create binary sequence as output of Z_1 as $\{X_1 mod\ 2\}$.

II. Parameter modified quadratic BBS:

a) Choose two distinct Blum prime numbers p and q for n such that $N = p \times q$.
b) Choose a scalar X_0 any random seed with $gcd(X_0, N) = 1$.
c) Define D as $D = {X_0}^2 mod g$ where$g = lcm(p-1, q-1)$.
d) Generate the first iteration by $y_1 = (gqD)^2$ with $X_1 = X_0 Y_0 mod N$.
e) Convert X_1 to Z_1 such that parity (X_1) $= Z_1 mod\ 2$.
f) Create binary sequence as output of Z_1 as $\{X_1 mod\ 2\}$.

In the following figure, the flow of thesetwo algorithmsare presented:

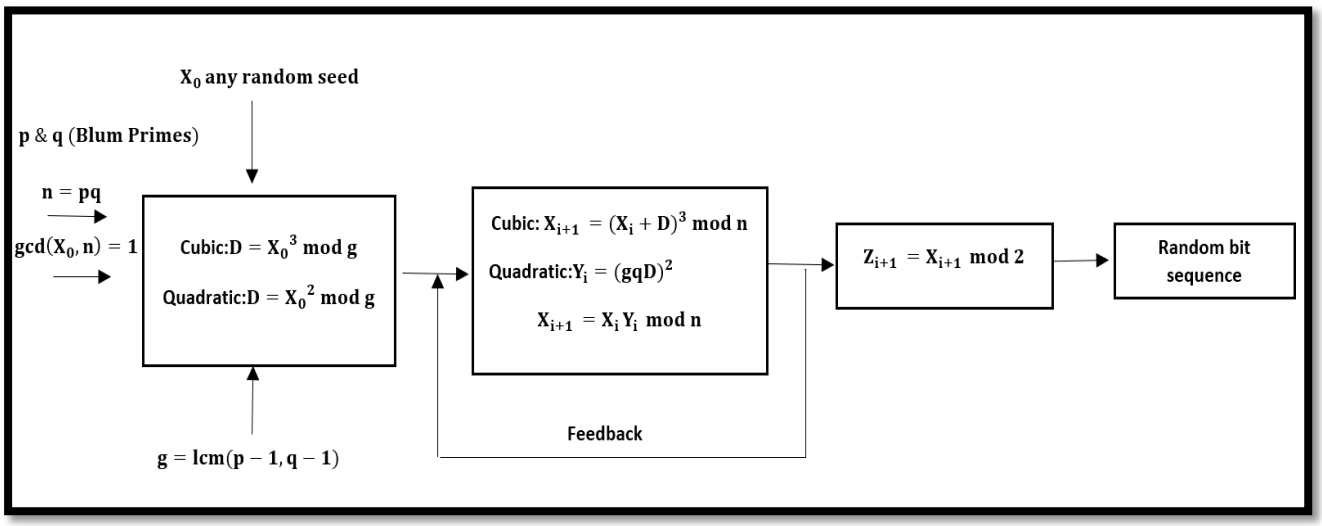

Figure 6: Cubic and quadratic function with an embedded parameter.

The following examples can explain the loop of the modified cubic and quadratic algorithms:

Example 1:

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

Step 2: Let $x = 100$.

Step 3: Define D as $D = {X_0}^3 mod g = 4$ where$g = lcm(p-1, q-1) = 6$.

Step 4: Generate the first iteration $X_1 = (X_0 + D)^3 mod N = 20$.

Step 5: Evaluate $Z_{1\,S}$ such that Z_1 =party $(X_1) = X_1 \, mod \, 2 = 20 \, mod \, 2 = 0$.
Step 6: Binary sequence: {0}.
Example 2:
Step 1 Let p = 3, q = 7, n = pq = 21.
Step 2: Let x = 100.
Step 3: Define D as $D = {X_0}^2 mod g = 4$ where$g = lcm(p-1, q-1) = 6$.

Step 4: Generate the first iteration by $y_0 = (gqD)^2$ with $X_1 = X_0 Y_0 mod N = 0$.

Step 5: Convert X_1 to Z_1 such that parity (X_1) $= Z_1 mod \, 2 = 0$.

Step 6: Binary sequence: {0}.

5 Results and Discussion

By applying the hypothesis randomness test for all above algorithms, the prime numbers $p = 203$ and $q = 603$ and the iterationnumber as 100 are fixed. Results of the hypothesis test on randomness are mentioned in the Table2:

	Algorithms	**Length of Bitstream**	**HypothesisTestResult**
1	Original BBS	100	fulfill the randomness test
2	Cubic BBS	100	not fulfill randomness test
3	Quartic BBS	100	fulfill the randomness test
4	Sequence BBS	1000	fulfill the randomness test
5	2×2 matrix BBS	400	fulfill the randomness test
6	3×3 matrix BBS	900	fulfill the randomness test
7	Parameter Modified Cubic BBS	100	fulfill the randomness test
8	Parameter Modified Quadratic BBS	100	fulfill the randomness test

Table 2: Results for hypotheses test on randomness generated by the algorithms

The data of all algorithms had been chosen as virtual collections that could be represented as samples in encryption system to check their algorithm's behavior. Despite the iteration is equal to 100 of each algorithm, the increment can be noticed in some bit sequence such as(Sequence BBS, 2×2 matrix BBS, and3×3 matrix BBS) with the same primenumbers $p = 203$ and $q = 603$.To illustrate that, the Table3 describes the output of bit sequence of all algorithms:

Algorithm	1	2	3	4	5	6	7	8
One Iteration (sequence)	1-bit	1-bit	1-bit	20-bit	4-bit	9-bit	1-bit	1-bit
100 Iterations (sequence)	100-bit	100-bit	100-bit	2000-bit	400-bit	900-bit	100-bit	100-bit

Table 3: The bit sequences of all algorithms

We can notice that the bit sequence of sequence BBS algorithm is 20 times greater than the original BBS per one iteration, and the bit sequence of2×2 matrix BBS algorithm is 4 times greater than the original BBS while the bit sequence of3×3 matrix BBS is 9 times greater than the original BBS.
Consequently, the evaluation of the hypothesis randomness test in the table 3 at significance level $\alpha = 0.05$ depends on the normal distribution formula that consists of run expectation number and standard deviation. On the other hand, the value of normal distribution Z illustrates the statistic test values for all data which are in between the critical interval $-1.96 \leq Z \leq 1.96$. It implies that the bit sequence of all algorithmsaccepts the null hypotheses and fulfill the randomness expected the cubic BBS algorithm. Therefore, the cubic BBS algorithm is not useful as a pseudo-random number generator.

6 Conclusion

In this paper, we have proposed seven modifications on BBS algorithm. Our randomness hypothesis test showed that they all fulfilled the randomness test except cubic BBS. Moreover, the extended bit sequence of arithmetic sequence BBS, 2×2 matrix BBS, and3×3 matrix BBSalgorithms can increase the randomness cycle despite lower iterations, which leads to improve randomness, but other modified algorithms still generate one bit sequence per iteration.

Acknowledgement
The author would like to thankAl-Nahrain University andUniversiti Putra Malaysia for the supporting this research.

7 Reference

[1] Laia, O., Zamzami, E. M. and Sutarman(2021). Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS). J. Phys.: Conf. Ser. 1898 012017.

[2] Jaffara,M. , Joey, F. (2022). Modification to Queueing System M/M/1 with Blum-Blum-Shub Generator. 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM2022), Mosul University, Mosul-Iraq.

[3] Stallings, W. (2011).Cryptography and Network security principles and practices. Prentice Hall, Fifth Edition.

[4] Pasqualini, L. and Parton, M. (2020). Pseudo Random Number Generation: a Reinforcement Learning approach. Procedia Computer Science, (170):1122–1127.

[5] Blum, L., Blum, M., and Shub, M. (1983). Comparisonof Two Pseudo-Random Number Generators. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds) Advances in Cryptology. Springer, Boston, MA..

[6] Julian, F. F., Philip, K. H., Volker, E. and Christine, V. R. (2015). Number Theory. Discovering the Art of Mathematics.

[7] Sidorenko, A. and Schoenmakers, B. (2005). Concrete security of the Blum-Blum-Shub pseudo-random generator.in: Lecture notes in computer science, Cryptography and Coding, Springer-Verlag, Berlin, (3796): 355–375.

[8] Blum, L., Blum, M. and Shub,M.(1986).A simple unpredictable pseudo-random number generator. SIAM Journal on Computing, 15(2):364-383.

[9] Kadhim,E., Hussein,U. and Hadi,S. (2017). AES Cryptography Algorithm Based on Intelligent Blum-Blum-Shub PRNGs.Journal of Engineering and Applied Sciences, vol. 12(10):9035-9040.

[10] Joey, F.(2023). Modification of Blum-Blum-Shub Generator (BBS) with a 2×2 Matrix and the First Digit Property of Generated Random Numbers and Bits" Applied Mathematics and Computational Intelligence,(12)2:120-129.

[11] Junod,P.(1999). Cryptographic secure pseudo-random bits generation: the Blum-Blum-Shubgenerator. (accessed 22.11.2016).

[12] Bujang, M., Sapri, F. (2018). An application of the runs test to test for randomness of observations obtained from a clinical survey in an ordered population. Malays J Med Sci., 25(4):146–151.