Indian Journal of Science and Technology, *Vol 10(27), DOI: 10.17485/ijst/2017/v10i27/113945, July 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Hybrid Approach for PRNGs using BBS and Dithering Technique

Estabraq Abdulredaa Kadhim, Firas Haqqi Ismael, Hassan Challob Mohsen and Alaa Abdul Hussein Jabbar

Computer Techniques Engineering, AL-Esra University College, Baghdad Iraq; Estabraq_ai_1989@yahoo.com, firasbajlan91@gmail.com, hasangalob@gmail.com, abda95947@gmail.com

Abstract

Objectives: The cryptographically secure pseudo-random number generator Blum Blum-Shub (BBS) is a simple algorithm with a strong security proof, all of number generators that produced by BBS will be converted to (0, 1) using simplest form called Parity (even parity bit and odd parity bit).This approach proposed new method for extending bit space acquired from BBS based on dither matrix instead of (Parity) feature with saving of strength and security of crypto-key. **Methods/Statistical Analysis:** We used dithering techniques (2*2, 4*44) that provide continuous image of higher colors on a display of less color depth, for improving numbers that generated by BBS. And used some of statistical tests for measuring crypto-key strength such (run, poker, serial and frequency tests).The proposal algorithm has been programmed using c#.net 2013 **Findings:** This approach provides high extended on number space that generated from BBS by using Dithering techniques (2*2, 4*4), with saving of crypto-key randomness and guarantee passing of statistical tests, dithering techniques (8*8) is unsuitable for this purpose. Although, it provides height extension but decrease the randomness of key numbers. **Application/Improvements:** This approach proposed to replace (Parity) property with dithering techniques. It's replaces max of gray level in dithering such (256) in 8 bit image with value of n , $(p*q)$ in BBS , this may convert the domain of any number sets to dithering domain to be suitable for purpose.

Keywords: Blum-Blum-Shub, Cryptography, Cryptography Key, Dithering, Pseudo Random Number generation (PRNGs)

1. Introduction

Pseudo Random numbers are critical in every aspect of cryptography such as encrypt e-mails. It's also known as a Deterministic Random Bit Generator (DRBG). The PRNG-generated sequence is not truly random, because it is completely determined by a relatively small set of initial values, called the PRNG's seed which may include truly random values[1]. BBS is a simple algorithm with a strong security proof; however it requires very large numbers to be secure, which makes it computationally heavy[2]. It is completely unpredictable even when a long sequence of bits has been generated. The underlying theory is based on quadratic residues, and cracking is equivalent to integer factorization[3]. Dithering is a technique added to any low-amplitude or highly periodic signal before any quantization or re-quantization process, in order to de-correlate the quantization noise from the input signal and to prevent non-linear behavior[4]. This approach proposed method for replacing (Parity) feature by dither matrix to increase the randomness of bit sequences and produce high key expansion from the same set of BBS numbers.

2. Previous Work

Estabraq Abdulredaa in 2015, proposed approach for enhancing BBS-integer values generated using Iterated Local Search (ILS) metaheuristic to create strong crypto key that passed through some nonparametric statistic test (Run, Sign and Wilcoxon Signed Rank tests). The experimental results shown that improvement ratio of generated crypto key numbers are ranging from [40%-90%][5].

*Author for correspondence

3. Blum-Blum-Shup

A Pseudorandom Number Generator Proposed In 1986 by Lenore Blum, Manuel Blum and Michael Shub that is derived from Michael O. Rabin's oblivious transfer mapping[6].Currently the generator which has the strongest public proof of strength is called the Blum Blum-Shub generator after its inventors it is also very simple and is based on quadratic residues. This is not a serious draw back if it is used for moderately infrequent purposes, such as generating session keys[7]. Simply choose two large prime numbers, say p and q, which both have the property that you get a remainder of 3 if you divide them by 4. The BBS generator, or x_2 mod N generator, is an example of a nonlinear congruently generator .The generator does not process any input after it is seeded once[8]. BBS pseudo random number generator shows in following steps:

Step1: Generate p and q, two big Blum prime numbers

Step2: n: =p*q

Step3: Chooses∈R [1, n-1], the random seed

Step4: x:=s2 (mod n) 0

Step5: The sequences is define as x:=x2 (mod n) and z: =parity(x)

4. Ordered Dithering

Ordered dithering is an image dithering algorithm. It is commonly used by programs that need to provide continuous image of higher colors on a display of less color depth. For example, Microsoft Windows uses it in 16-color graphics modes. It is easily distinguished by its noticeable crosshatch patterns. The algorithm achieves dithering by applying a threshold map on the pixels displayed; causing some of the pixels to be rendered at a different color, depending on how far in between the color is of available color entries. Different sizes of threshold maps exist[4].

¼ $*\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$

$\frac{1}{9}*\begin{pmatrix} 0 & 7 & 3 \\ 6 & 5 & 2 \\ 4 & 1 & 8 \end{pmatrix}$

<eq Problem>

It can re-map pixel values from specific grey level to a new range of dithering matrix by dividing the value by (256/17) (and rounding down).Replace each pixel by dots (binary pixels). If the remapped intensity is > the dither matrix entry, put a dot at the position (set to 1) otherwise set to zero[9].

5. Hybrid Approach

Blum-Blum-Shub (BBS) is one of important methods for secure pseudorandom number generator .It have several benefits such as each term x_i can be computed directly, without prior knowledge of x_{i-1} or any other terms besides x_0. All of number generators that produced by BBS will be converted to (0,1) using simplest form called Parity (even parity bit and odd parity bit). This consider weak point for any binary sequence, it's possible to predicate the weather of secret number (even or odd) from property. This approach proposed to replace (Parity) property with dithering techniques. It's replaces max of gray level in dithering such (256) in 8 bit image with value of *n* , *(p*q)*in BBS , this may convert the domain of any number sets to dithering domain to be suitable for purpose as shows in equation 1:

New key = (old key/ [(n-1)/threshold]) (1)

Where

n= (p*q) from BBS

Threshold= 5 if dither matrix is 2*2

17 if dither matrix is 4*4

This will provide more distribution, randomness then, and more complication to key sequence, Algorithm (1) shows the mains steps this proposal

Algorithm(1): Hybrid of BBS and Dithering Matrics

INPUT

p,q where p, q are prime number and p ≡ q ≡ 3 (mod 4)

Type of Dither-matrix (either 2*2 or 4*4)

OUTPUT:

BBS-Dither key

PROCESS

(BBS)

Step1: n =: p*q

Step2: Choose **s** ∈ R[1,n-1],the random seed

Step3: $x_1 := s^2 (\text{mod } n)$

Step4: The sequence is define as $x_2 := x_1 (\text{mod } n)$

Step 5: Generate BBS- key sequence

(DITHER)

Step 6:

1. **While(**BBS- key not met to length **) Do**
2. Remap key element by following equation

New key = (old key/ [(n-1)/threshold])

3. Rounding down of **New key**
4. Compare New key with dither matrix
5. Generate binary key sequence
6. End While

Note: threshold=5 if dither matrix is 2*2
threshold=17 if dither matrix is 4*4

Step 7:END

6. Implementation

This section illustrates the implementation of the proposed approach which programmed using Visual c#.net 2015. The following example explains in details steps of proposed algorithm work

BBS-Traditional

Let P=11 q=19 n=209 GCD (s, n) s=67 length=5

N=p*q→ N=11*19=209

S=Rand (0,208),→S= 67

X0=S^2 mod N

X0=67^2 mod 209 =100

X1=100^2 mod 209=177

X2=177^2 mod 209 =188

X3=188^2 mod 209 =23

X4=23^2 mod 209 =111

X5=111^2 mod 209 =199

BBS Key = [100,177,188,23,111, 199]

BBS Key (Parity) =[010111]

BBS- Dithering 2*2 Matrix

0	2
3	1

New Rang= (n-1)/ → (209-1)/5=41

Remap key element (100) = 100/41=2 → [1001]

Remap key element (177) = 177/41=4 → [1111]

Remap key element (188) =188/41=4 → [1111]

Remap key element (23) = 23/41=0 → [0000]

Remap key element (111) = 111/41=2 → [1001]

Remap key element (199) = 199/41=4 → [1111]

BBS Key (Dither 2*2) = [1001 1111 1111 0000 1001 1111]

BBS- Dithering 4*4Matrix

0	8	2	10
12	4	14	6
13	11	1	9
15	7	13	5

New Rang = (n-1)/17 → (209-1)/17=12

Remap key element (100) = 100/12=8 → [1010 0101 0010 0101]

Remap key element (177) = 177/12=14→[1111 1101 1111 0111]

Remap key element (188) = 188/12=15→[1111 1111 1111 0111]

Remap key element (23) =23/12=1→[1000 0000 0000 0000]

Table 1. Statistic Measurements for Multi-BBS-Keys

N	s	BBS methods	Frequency Test <=3.8	Serial test <=5.99	Poker test <=11.1	Runs test <=22.3
161	55	BBS-Traditional	0	0.33	5	2.98
		BBS-Dither 2*2	0.1	3.05	3.6	6.4
		BBS-Dither 4*4	0.2	1.8	7.5	12.8
209	25	BBS Traditional	0	1	7	2.3
		BBS-Dither 2*2	0	3.4	3	4.5
		BBS-Dither 4*4	0	3.6	15.1	7.6
713	312	BBS-Traditional	0.4	3.9	2.1	6.5
		BBS-Dither 2*2	0.45	2.3	6	11.2
		BBS-Dither 4*4	3.7	5	9.6	20
309	128	BBS-Traditional	4.5	3.9	6	2.9
		BBS-Dither 2*2	0.125	5.6	7.6	7.2
		BBS-Dither 4*4	3.7	17.09	20.4	16

Remap key element (111) = 111/12=9→[1110 0101 0010 0101]
Remap key element (199) = 199/12=16→[1111 1111 1111 1111]
BBS Key (Dither 4*4) = [1010 0101 0010 0101 1111 1101 1111 0111 1111 1111 1111 0111 1000 0000 0000 0000 1110 0101 0010 0101 1111 1111 1111 1111]

7. Experimental Result and Discussion

This section illustrates the results that obtained according to implemented proposed approach which explained in the previous sections, these result can be listed in following:

- Table 1 shows the results of applying statistical measurements of randomness on traditional BBS and improving BBS based on dithering 2*2 and 4*4 matrices. These tests are used to check randomness and distributed properties of several traditional BBS and BBS-dither samples. Useful statistical tests are four basic tests, including: Frequency test, Serial test, Poker test, Runs test[10]. The output of tests must be compared with passes values to decide if the outputs of randomness tests are good or not.
- Following chart in Figurer1 shows the differences in the expansion of key sequence generation between BBS and BBS-dither techniques
- According to most of experimental results, it's notice that implementing BBS with dither matrix 8*8 was implied to worst solution (i.e. generate key sequence with lowest randomness). This is not suitable for using to security requirement; Figure2 illustrates percentage of security between traditional BBS and BBS-Dither 8*8.

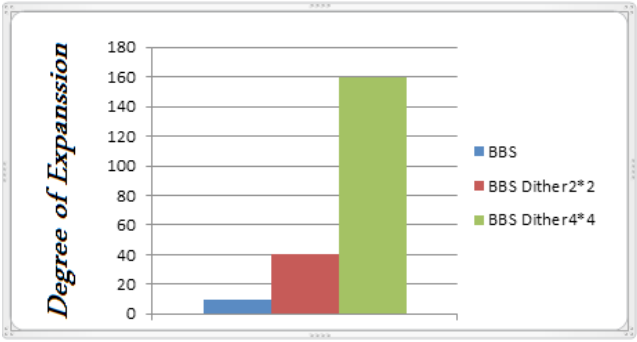

Figure 1. Key expansion between BBS and BBS-Dither.

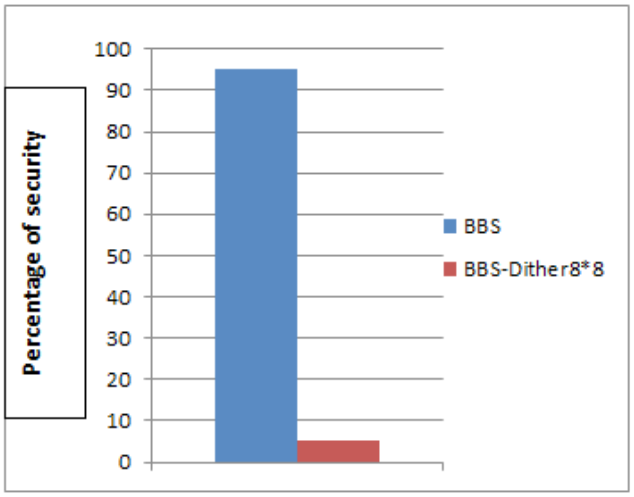

Figure 2. Percentage of Security Measurement between BBS and BBS-Dither 8*8.

8. Conclusion

We proposed a method for improving and expanding PRNGs based on hybrid of BBS and dithering technique that implied to save the secure of key and gave height length of extension. Improved key was passing through most of statistical measurements expect the sequence that generated by dither matrix 8*8. Also the size of the dithered key may be much larger. Since each key element in traditional BBS **either** replaced by4 × 4 array of bit, the key sequence becomes 16 times as large **or** becomes 4 times as large whenkey sequence replaced by 2 × 2 array of bit.

9. Future Work

Improving BBS with dither matrix 8*8 and benefit of having height-self generation of number spaces.

10. References

1. Neumann JV. Various techniques used in connection with random digits. National Bureau of Standards Applied Mathematics Series. 1951; 12:36-40.
2. Olsson M, Gullberg N. A performance comparison between a CPU bound and a GPU bound Blum Blum-Shub generator. School of Computing, Blekinge Institute of Technology, Sweden. 2012 January; p.101-10.
3. Barker E, Barker W, Burr W, Polk W, Smid M. Recommendation for Key Management- Part 1: General (Revision 3). NIST Special Publication 800-57 Part 1, Revision 3. 2012 July; p. 40-56. Crossref

4. Pettofrezzo AJ, Byrkit DR. Elements of Number Theory. 1970; p. 244.
5. Kadhim EAR.Number Generator Improvement Based On Artificial Intelligent And Nonparametric Statistic Methods. International Education & Research Journal. 2015 December; 1(5):1-6.
6. Menezes AJ, Oorschot CBV, Vanstone AS. Handbook of applied cryptography. The CRC Press series on discrete mathematics and its applications. USA. 1996. Crossref
7. Blum L, Blum M, Shub, M. A Simple Unpredictable Pseudo-Random Number Generator. SIAM Journal on Computing. 1986 May; 15(2):67-78. Crossref
8. Shannon C. Communication Theory of Secrecy Systems. Bell System Technical Journal. 1949; 28(4):656-715. Crossref
9. Barker E, Roginsky A. Recommendation for Cryptographic Key Generation. National Institute of Standards and Technology Special Publication 800-133. 2012 December; p.12-30.
10. Recommendation for Cryptographic Key Generation. Date Accessed: 2012/12: Available from: http://nvlpubs.nist.gov/nistpubs/ Special Publications / NIST.SP.800-133.pdf