INTERNATIONAL CONFERENCE ON COMBINATORICS, CRYPTOGRAPHY AND COMPUTATION

Iran University of Science &Technology

Proceedings of the 2nd International Conference on Combinatorics, Cryptography and Computation (I4C2017)

Implementation of Modified Chaos- based Random Number Generator for Text Encryption

Rahim Asghari[1,*]
Faculties of Information, Communication and Technology,
Malek-Ashtar University of Technology, Tehran, Iran.
Meisam.mathhome@gmail.com
Ali Jamalian[2]
Computer Science Department, University of Guilan, Rasht, Iran.
a.jamalian.math@gmail.com
Hadi Rezazade[3]
Faculty of Engineering Technology, Amol University of Special Modern Technologies, Amol, Iran.
rezazadehadi1363@gmail.com

ABSTRACT

In the present paper, a Modified Chaotic Henon Congruential Generator is implemented as a stream cipher cryptosystem and the new proposed cryptosystem is compared with the original chaotic henon congruential generator cryptosystem. The proposed stream cipher has been tested using some examples and the algorithm security analyzed by different methods and statistical analysis. The results of the analysis confirm significant improvement over the modified algorithm.

KEYWORDS: Cryptography, Stream cipher, Chaotic Pseudo random numbers, Security Analysis.

1 INTRODUCTION

Random numbers have a vital role in cryptography field. Such numbers are essential to encrypt documents for electronic communications. It is well known that true random numbers are very difficult to generate, noting that what we have computers that are typically designed to be deterministic, henceforth there is no way out of generating at least well behaved pseudo-random numbers for us. Although pseudo random numbers are typically very hard to be distinguished from true random numbers, but they are tractable which makes the fear of losing security. The importance of creating intractable pseudo random numbers is tied by the security as a general concept. A pseudo random generator (PRNG) is a deterministic algorithm that for a truly random binary sequence of length N, provides a binary sequence of length $m > N$ that "looks random". Due to the importance of PRNGs, a lot of researches attended to the issue to make or analyse proper algorithms for generating secure pseudo random numbers. Suitable PRNGs should have the following properties:

- Independency
- Uniformity
- Unpredictability

The meaning of pseudo random numbers in the context of cryptography differs from the meaning in normal programming contexts where we just need reasonably random-looking and good statistical properties [1,2, 3]. PRNGs are designed for different purposes including simulation, electronic gaming and cryptographic applications. The latter is called a cryptographically secure PRNG (CSPRNG). The design of CSPRNGs should consider that the seed do not play a role on strength of the output. That is, in order to distinguish the generator's output sequence from a random sequence adversary may not depend on seed. In summary all statistical tests that are restricted to polynomial time in the size of the seed should passed by a CSPRNG while in the case of PRNG some certain statistical tests are sufficient [3, 4]. The strength of proper CSPRNG should be supported by strong evidence in reduction of the problem into a hard one e.g. integer factorization. In general, years of review may be required before an algorithm can be certified as a CSPRNG. A lot of algorithms have introduced in recent years for different purposes. For cryptographic purposes, we can refer to papers. One of the suitable methods is A Modified Chaotic Henon Congruential Generator (MCHCG) which is introduced in 2016 by Behrouz Fathi Vajargah and Rahim Asghari [5] that is a modification on Chaotic Henon Congruential Generator (CHCG) [6].

The Chaotic Henon Congruential Generator (CHCG) Algorithm is proven to be proper PRNG for cryptographic purposes [5, 6] however predicting versions of the mainly Algorithm suffer from intractability of quadratic residuosity problem [5]. Though CHCG is proper PRNG, but it is improved for different purposes. As an example the uniformity of generated numbers is improved. In the present paper, the MCHCG generator is implemented for text encryption as a stream cipher system. Results of statistical tests and histograms are approved significant improvement was done in cryptography applications. Furthermore, some important statistical tests such as BSI and NIST; including Gap test, Poker test, and Run test are implemented to verify randomness properties [4, 9] and some methods is used for security analysis. The modified CHCG is constructed as a stream cipher cryptosystem and compared with the CHCG cryptosystem [7]. Security analysis is done over the cryptosystems based on the MCHCG and the CHCG generators. The analysis results are reported in section 4 and confirm our claims.

The paper is organized as follows: in section 2, the CHCG algorithm is introduced. In section 3, modified algorithm (MCHCG) is presented. In section 4, the modified CHCG is implemented and simulation results compared with the CHCG cryptosystem in section 5. Finally, the conclusion is drawn in section 6.

2. The Chaotic Henon Congruential Generator (CHCG)

As mentioned in many papers [1, 5, 6], a main problem of Linear Congruential Generator (LCG) is small period. That's why LCG is not suitable for cryptographic applications. In [5], Because of high sensitive to initial value and chaotic behaviors, to use a henon map in the LCG generator. Author's to combine the LCG and henon map, and named The Chaotic Linear Congruential Generator (CHCG) [5]. The author's purpose was to have a key stream with suitable period to generate an efficient key stream. The algorithm of CHCG has presented as following:

Algorithm 1: CHCG

select x_0

for $i = 1$ *to* n

$x_i = a x_{i-1} + b \pmod{m}$

for $j = 1$ *to* $i - 1$

if $x_i == x_j$

$x_{i+1} = 1 + \alpha x_i + \beta x_{i-1}^2$

end if
end for
end for

3. The Modified Chaotic Henon Congruential Generator (MCHCG)

Although the CHCG was tested by statistical tests and has showed is suitable for cryptographic applications, but the statistical properties can be improved. In [6], the uniformity as well as independence of the CHCG is improved . This is done by applying a suitable controller to regulate the distribution of PRNs generated by CHCG. First, the interval is divided into some subintervals and then produces random numbers; and at every step of the algorithm a subinterval is chosen randomly and then recall CHCG to generate PRN at this subinterval. The algorithm is presented as follows:

Algorithm 2: Modified CHCG

Start: Choose a seed S.
Preparation: Divide the interval I into some subintervals I_j .
Run: CHCG generator for generating $y_1, y_2, \dots$
$For\ j = 1, 2, \dots$
1. Choose a subinterval randomly and name it I^*.
2. Transform y_j into subinterval I^* put into x_j .
Return: $x_1, x_2, \dots$ As generated random numbers by MCHCG.

As noticed in [6], a modification on CHCG generator is done to improve statistical properties of the CHCG generator with proper scattering of generating random numbers is used. The statistical tests on CHCG and MCHCG showed improvements clearly [5, 6, 7].

4. Implementation of the MCHCG and simulation

In this section, the MCHCG generator is implemented based on a synchronous stream cipher system. For suitable evaluation, results of implementing CHCG and MCHCG algorithms are compared with an example. For comparisons, histogram analysis and correlation coefficient analysis, key sensitivity analysis and NIST tests are used.

4.1 Implementation and Simulation

The steps of encryption and decryption approach in proposed stream cipher are working as follows:

Step1: Read the plain text (original message) and converts it into its ASCII code.
Step 2: Convert the ASCII value into binary coding.
Step 3: Compute the key stream sequence produced by MCHCG.
Step4: Encryption of binary coding using the key stream.
Step 5: Decryption of encrypted binary coding using the same key stream.
Step 6: Get the decrypted message and comparing with original message.
Above algorithm is written by visual c#2012 and an example is presented as simulation result

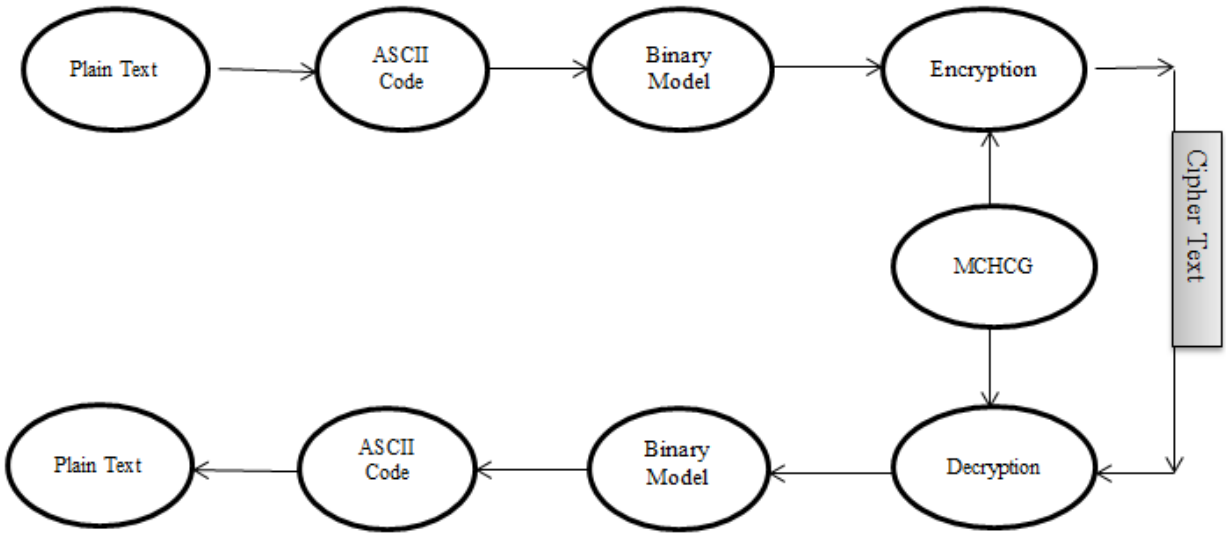

Figure1. Block diagram of the approach

Algorithm3: Encryption Process	Algorithm4: Decryption Process
1. Read the original message as plain text; 2. Convert the original message to integer data; 3. Convert integer data into binary model; 4. N=length of plain text; 5. for i=1 to N to do 6. Compute the key stream by MCHCG; 7. Encrypt the data digit by key stream; 8. end for; 9. Present to cipher text;	1. Read the cipher text; 2. Convert the cipher text into integer data; 3. Convert integer data into binary model; 4. N=length of cipher text; 5. for i=1 to N to do 6. Compute the key stream by MCHCG; 7. Decrypt the data digit by key stream; 8. end for; 9. Present to original message;

Example

In this example, a plain text indicated in table 1 is used and their encrypted messages by MCHCG generator and CHCG generator shown in table 2 and table3, respectively. By comparing table 1, table 2 and table3, very much different between the encrypted messages and original messages is proved. In table4, decrypted text by MCHCG and CHCG based stream ciphers is presented. Both of them are same. Note that, in this example the MCHCG and CHCG generators run by following coefficients:

$x(0) = 0.5, a = 75, b = 250, m = 2^{31} - 1, \alpha = -1.4, \beta = 0.3$

Table 1. Plain text

It is well known that true random numbers are very difficult to generate, noting that what we have computers that are typically designed to be deterministic, henceforth there is no way out of generating at least well behaved pseudo-random numbers for us. Although pseudo random numbers are typically very hard to be distinguished from true random numbers, but they are tractable this makes the fear of losing security.

Table 2. Encrypted text by MCHCG algorithm with k_1=44

Mc9lp>enIk!LgpfaWetozzSo8~EBD• q4Fh~xsXu"S\'[N[@PZWQV";P4tA,E_) -
"[2"` K″3<90!"u,V3"d
s• xq#iw+MfwB)|~bQVypiQ.|Nkl,E^E0hmXtp{zF• "VK1FJEL]@\UZa*Y41G. V%<9Y⁈ k^
S>6+6=1= p• |wf>{xin~~h/NVy5tD.oCd}~EXI~{4Ii3vsKuvY'CAK\XUETS1=Y50ZnY"
qd_+#_~□Mht@a?a|DViz;PofBeu,JYMryf[=rhs vK^+LLGE@BVCN)/N$tA,EZ)D,9I2.UQ-T.;(=o
bevh#j`~@'sFg{~bMxxyG|{&8fbm%wvwrfsB)kcnBWlwwG.|Nck,IMKuo4\uv:pOgpA$ADZ

Table 3. Encrypted text by CHCG algorithm with k_1=44

M|#Hj>yoN• 7A|rqa<x~da+`cVh0[QV|Kj)FoM}@sqM7~bC+TW]T"_H9:<`C5-.XU53[MJ‰ t
f:3=Gé$6H4'k s`bU9ik*JraO2~iblyb`gx4eKldQJ]spXsC~Im{H h+R CJH[&S (3<%E7W'TH$(Y-
0#:>7Uï8% p`fS|>gy}xe|• /syb%zm4vFcu[QLqJ`)In q{9_b~J|bMz=v|yyrj5yu• Gb}^MuFb{[:Am@!
SU5r!T"JWJQ@=C-$,`C5-+X 4(IMFQ?3=.:Väj bzlL9j|• G3eK|yib<bchwnfb!0

Table 4. Decrypted text by CHCG and MCHCG algorithms with k_1=44

It is well known that true random numbers are very difficult to generate, noting that what we have computers that are typically designed to be deterministic, henceforth there is no way out of generating at least well behaved pseudo-random numbers for us. Although pseudo random numbers are typically very hard to be distinguished from true random numbers, but they are tractable this makes the fear of losing security.

5. Security Analysis and Comparisons

In [6], proposed a cryptosystem based on CHCG generator. In this section, the new proposed cryptosystem based on MCHCG and CHCG cryptosystem are analyzed in more details and they compared together. The security analysis of the systems is included key sensitivity analysis, histogram analysis, correlation coefficient analysis and NIST tests. This is to prove that the new proposed cryptosystem is a suitable system in point of view of security and has more security according to CHCG cryptosystem.

5.1 Key Sensitivity Analysis

A good cryptosystem should be sensitive to the secret keys, which means a change of in a single bit in the secret key should produce a completely different encrypted message. The proposed cryptosystem was tested to the sensitivity of the keys and comparing to CHCG method. We encrypt the message shown in table1 with the secret key K_1= 44 and, we decrypt their encrypted messages shown table 2 with another different key; K_2= 45. The result is given in table 5. In other hand, this process is performed with the CHCG based cryptosystem and the result is given in table6. Then, the results compared together. So it can be concluded that the new proposed crypto system is highly sensitive to the key, because of this sensitivity a small change in main key lead to a different message according to the original message.

Compared to results is proved that MCHCG method has more sensitive to the secret key. For more celerity, histograms of tables 1, 4, 5, 6 are presented in figures 1 and 2.

Table 5: Decrypted message by MCHCG with K_2=45

Nf)LZ8gKBz>goVlyskVsRg@k/gPYExxII]uANojqIjl*bcbk fg =opc$ob zH‹bkl F9I~{+ pzhQ ou Fwhi!Ztz]rwRuEa]o{5PED7aJWUSvHPe ofXX}q*d%fgNwn x szlWL8y] xq,vXb7BrwhPrP`jgPCHyrFH{A]o_+t]SJzpb}``"-`n
y-"h ra Y|ZFykkikdHrgX'DG{Qab5_BLupATQeA–_rsR^tl*
nw}"ry=opcjnyvkX• l}Y a`fH lb[K6lmo]tzvZeSTf※"/wDC

Table 6: Decrypted message by CHCG and K_2=45

Nt!oJ!\nfe/e{{lg3|evu:CnFA,WGSi@@'LM^jlmZ7uPCZhc?mobhZu0uq!obl{,t ,Z fbZS0d0
h`rvN+bhyk5wtdc}yrsih[ExGOhY~RQPiesP7pFK~?})fdNSuusshfnq`·:=LhelWzq
shdt\!Bx*g`.bub)|}y7n|{VJiWGIdAJ'CLrs(@G• bfzi}4kzv}ix!hVrWKaHH`M HuQRzl?· cm\ arnkuY~o)}o{ptd3fxzc Eo?,GSI-[Eb[Rzl?]eu

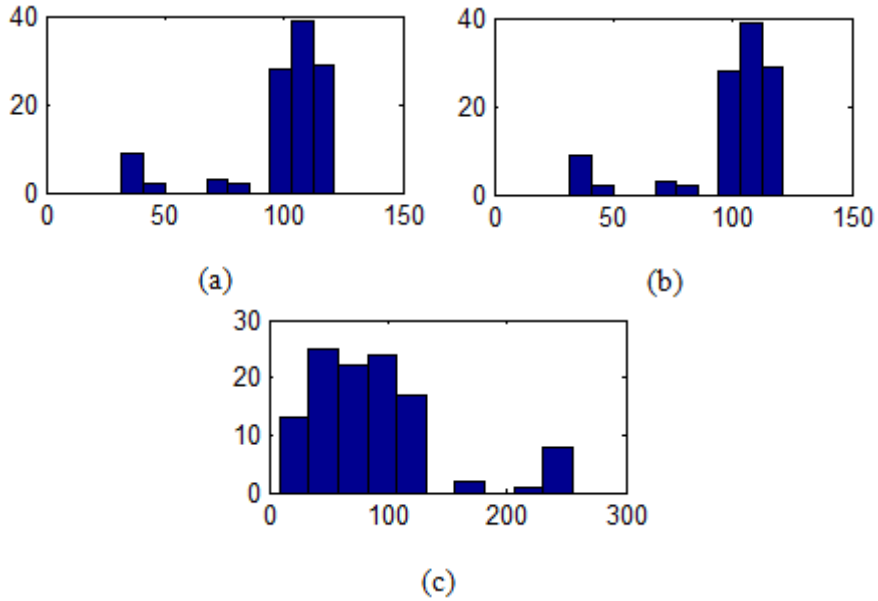

Figure 1. Sensitivity analysis: Frame (a), Frame (b), Frame (c) show histogram of messages in table1-4 and table5.

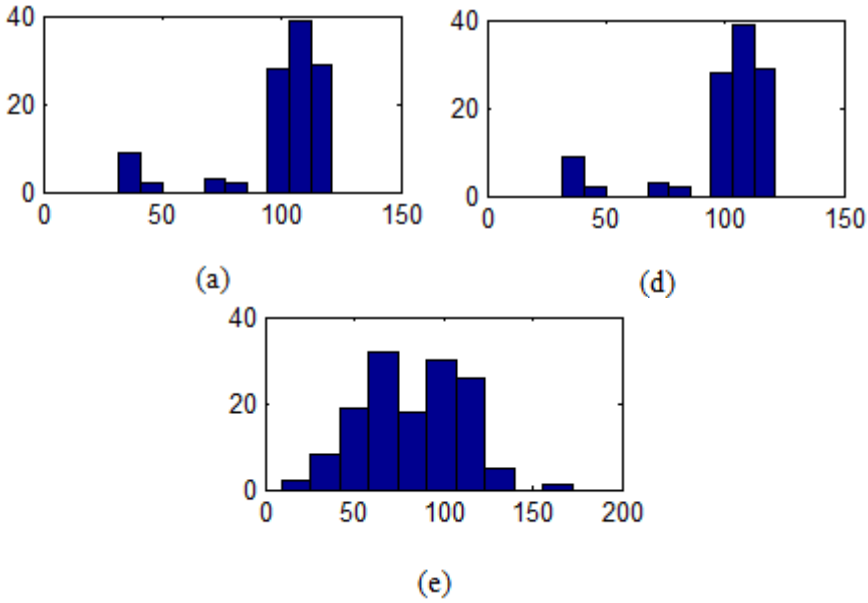

Figure 2. Sensitivity analysis: Frame (a), Frame (d), Frame (e) show histogram of messages in tables1, 4 and table 6.

5.2 Correlation Coefficient Analysis

Correlation coefficient results are presented in Table 7. Correlation-1, Correlation-2, and Correlation-3 are correlation coefficient between original message and encrypted message with correct key (k_1=44) and correlation coefficient between original message and decrypted message with correct key (k_1=44) and also correlation coefficient between original message and decrypted message with wrong key (k_2=45), respectively. Values of Correlation-1 shown in the Table 7 are proving that the original message and its encryption in MCHCG state has more different according to CHCG state and it's shown that in MCHCG state the plaintext and the cipher text has highly independent of the original message. Values of Correlation-3 shown in the Table 7 are proving that the original message and decrypted message with wrong key K_2 has more different according to CHCG stat. Therefore, MCHCG algorithm is very sensitive to change key and has more security. It's clear that Correlation-2 should be 1.

Table 7: Correlation Coefficient Analysis

Algorithms	Correlation-1	Correlation-2	Correlation-3
MCHCG based Algorithm	-0.0764	1.0000	0.2405
MCHCG based Algorithm	0.04691	1.0000	0. 1749

5.3 Histogram Analysis

The histograms of plaintext and their corresponding cipher texts are shown in Tables 1, 2, 3 are presented in figure 3. It is clear that the histogram of the encrypted message by MCHCG method is almost uniformly distributed, and significantly different from the respective histograms of the original messages. So, the encrypted messages do not provide any clue to employ any frequency attack on the proposed encryption message procedure, which makes frequency attacks difficult.

On the other hand, distribution of data in the new proposed method has more uniformity according to CHCG method. It's mean that, the security of the MCHCG based cryptosystem is more than CHCG based cryptosystem.

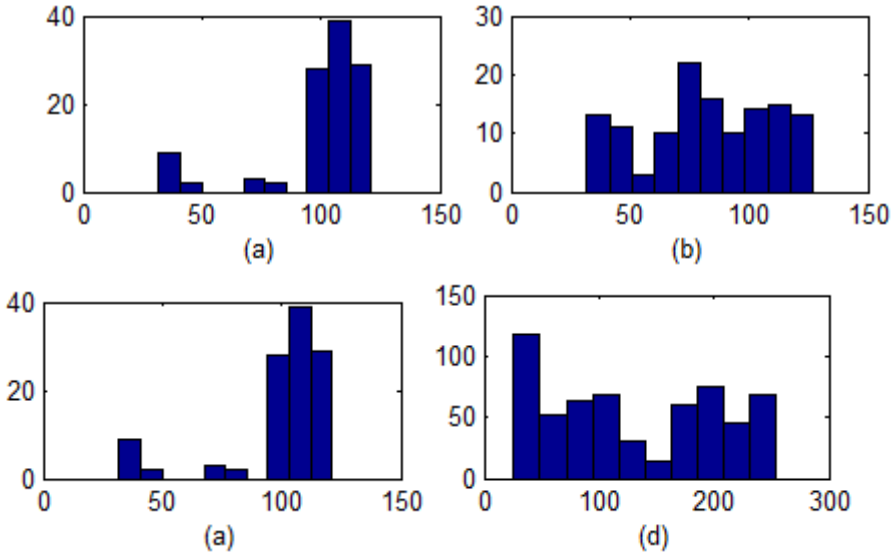

Figure 3. Histogram analysis: Frame (a), frame (b), frame (d), and respectively histogram of message given in Tables 1, 2, 3.

5.4 NIST Tests

In this section, the NIST tests are used on encrypted texts by the MCHCG and CHCG algorithms. The results are proved that encrypted texts by the MCHCG are better than the CHCG according to stochastic property. The NIST test results are presented in table 8.

Table 8. The NIST tests results

NIST Tests	MCHCG based Algorithm	CHCG based Algorithm
Frequency Test (within a Block)	Passed(0.91)	Passed(0.76)
Longest Run of Ones in a Block	Passed(0.84)	Passed(0.63)
Non overlapping Test	Passed(0.93)	Passed(0.88)
Monobit Test	Passed(0.69)	Passed(0.71)
Serial Test	Passed(0.72)	Passed(0.54)
Gap Test	Passed(0.87)	Passed(0.73)
Run Test	Passed(0.66)	Passed(0.65)
Cu Sums-forward	Passed(0.79)	Passed(0.72)
Cu Sums-backward	Passed(0.64)	Passed(0.68)
Binary Matrix Rank	Passed(0.90)	Passed(0.92)
Rank	Passed(0.90)	Passed(0.84)
FFT	Passed(0.48)	Passed(0.33)
Universal	Passed(0.59)	Passed(0.45)
Poker	Passed(0.77)	Passed(0.61)

6 Conclusions

In this paper, we implemented a MCHCG algorithm as a stream cipher in order to enhance the efficiency and security of the CHCG based cryptosystem. Statistical tests and some methods are performed over them and the results corned improvements. The proposed MCHCG is targeted stream crypto systems which are context based. The CHCG is a proper generator and proved CSPRNG and as mentioned it can be improved to act even better. In the present paper improvements in efficiency and

security are discussed, but for future researches we propose further improvements in implementation, or designing special improvements for special purposes.

References

[1] B. F. Vajargah, R Asghari," Application of Chaotic Maps in Designing Cryptographic Pseudo Random Number Generators ", Journal of Optoelectronics and Advanced Materials, Rapid Communications, **19** (1-2) , 109-116 (2017)

[2] B. Assa, M. Khaled and G. Lakhdar," Implementation of Blum Blum Shub Generator for Message Encryption", International Conference on Control, Engineering and Information Technology (CEIT14), 2014.

[3] A. Fouque, T. Vannet, "Improving Key Recovery to 784 and 799 rounds of Trivium using Optimized Cube Attacks" (PDF). Cryptology ePrint Archive. PP: 4-17, 2015

[4] A. Popov, "Prohibiting RC4 Cipher Suites" Internet Engineering Task Force (IETF), Vol (48), pp.: 1–6, 2015.

[5] B. F. Vajargah, R Asghari," A pseudo random number generator based on chaotic henon map (CHCG)", International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)", 5(15), 2026-37, (2015).

[6] B. F. Vajargah, R Asghari," A Novel Pseudo-Random Number Generator for Cryptographic Applications", Indian Journal of Science and Technology", 9(6), (2016).

[7] B. F. Vajargah, R Asghari," Implementation of Chaotic Henon Congruential Generator (CHCG) for Message Encryption", Journal of Theoretical Physics and Cryptographic, vol (9), 1-5, (2015).