JOURNAL OF THEORETICAL PHYSICS & CRYPTOGRAPHY

www.IJTPC.org/Publications

A Pseudorandom Number Generator using Irreducible Polynomial over GF(7^3)

J K M Sadique Uz Zaman[1], Ranjan Ghosh[2]

[1]Department of Computer Science, A.P.C. Roy Govt. College, Siliguri – 10, West Bengal, INDIA.

[2]Department of Radio Physics and Electronics, University of Calcutta, 92, A.P.C. Road, Kolkata – 9, INDIA.

Corresponding author e-mail: jkmsadique@gmail.com

ABSTRACT

The multiplicative polynomial inverse for all elemental polynomials exist under each of all irreducible polynomials over the Galois Field GF(p^m) where p is a prime integer and m ≥ 2. These multiplicative inverses can be innovatively used to design a pseudorandom number generator. A simple algebraic method, available in literature, finds all the multiplicative inverses of all the 342 elemental polynomials under each of the 112 irreducible polynomials over GF(7^3). A new pseudorandom number generator is proposed using the multiplicative inverses of all elemental polynomials under the first irreducible polynomial $x^3 + 2$ over GF(7^3).

Keywords

Extension field, Galois field, GF(7^3), Monic irreducible polynomial, Multiplicative inverse, Pseudorandom number, PRNG.

1. INTRODUCTION

In this paper multiplicative polynomial inverses under the first Irreducible Polynomial (IP) over Galois Fields GF(7^3), available in [1], are innovatively used to design a new pseudorandom number generator (PRNG) following the technique of randomly shuffling S-Box elements adopted in RC4 [2,3]. The 256 entries of initial identity S-Box of RC4 is replaced by a non-identity S-Box obtained from multiplicative polynomial inverses over GF(7^3). Other 87 multiplicative inverses are complemented at bit level and its decimal equivalent is sequentially put at the first initial 87 entries of the K-Box. The rest 169 spaces of the K-Box are filled by the given key. The output is tested using fifteen randomness tests proposed in the NIST statistical test suite and results are compared with that obtained by RC4. It is observed that the new PRNG is statistically random and quantitatively better.

The RC4 algorithm, where all additions are 256 modulo additions, starts from an 8-bit identity S-Box and the given key elements are repetitively stored in an 8-bit K-Box. It undertakes random shuffling of the S-Box elements first by using key elements stored in the K-Box and then without the key elements in order to systematically increase the arrangement of S-Box elements more and more random. A loop for Key Scheduling Algorithm (KSA) is executed 256 times where a sequential index (i) and a random index (j) both are initiated as zero. In each loop j is upgraded by addition of itself with S[i] and K[i], followed by swapping of S[i] and S[j]. After KSA, an infinite loop for Pseudo Random Generator Algorithm (PRGA) is executing in which both i and j starts from zero and in each PRGA loop i is upgraded by adding unity and j is upgraded by addition of itself with S[i] only followed by swapping of S[i] and S[j] – the result of addition of S[i] and S[j] is used as an index of random key byte. Many researchers [4–8] observed various types of weak keys. Even for good keys they also observed key bias [4,6] in few initial PRGA bytes and suggested many modifications in RC4 [7,8] in order to overcome the weakness. They also suggested that the conventional RC4, with no modifications whatsoever, would exhibit better performance without key bias if few initial PRGA bytes are discarded, possibly 256 as suggested by Roos [4]; but according to Preneel [6] the said amount should be at least 512 while it is 1024 as per Maitra [8]. Following a precise look, one would be convinced to notice two loopholes in RC4 algorithm behind the weakness of key bias: (1) considering initial S-Box with identity elements and (2) repetitive insertion of given key elements all through the K-Box.

One can overcome the weakness of key bias, if the initial identity S-Box is replaced by a non-identity S-Box and the given key characters are not repetitively inserted in the earlier part of the K-Box. In the present paper, the identity S-Box is replaced by a non-identity S-Box obtained from multiplicative polynomial inverses under the first irreducible

polynomial $x^3 + 2$ over Galois Field GF(7^3). Few elements of the initial K-Box are also obtained from some elements of multiplicative inverses and the rest are the repetition of the given keys.

In Sec. 2, a description of algebraic method to calculate multiplicative inverse over GF(7^3) is given shortly with two examples and algorithm. An overview of RC4 algorithm and the technique of getting a new PRNG based on the multiplicative inverse are presented in Sec. 3. Motivation of the NIST statistical randomness tests is briefly described in Sec. 4. The statistical randomness tests are undertaken on the output of the new PRNG and the results are described in Sec. 5. The conclusion is in Sec. 6.

2. DESCRIPTION OF ALGEBRAIC METHOD AND FINDING MULTIPLICATIVE INVERSE OVER GF(7^3)

The multiplicative polynomial inverse of each element for all the 112 monic irreducible polynomials [9–12] over Galois Field GF(7^3) can be calculated successfully by Algebraic method mentioned in reference [1]. The method is discussed briefly in Sec. 2.1; and two applications of the method are presented in Sec. 2.2 and 2.3. An algorithm to find all the multiplicative inverses under an irreducible polynomial over GF(7^3) is given in Sec. 2.4.

2.1 Description of Algebraic method to find the multiplicative inverse over GF(7^3)

Let $I(x) = (x^3+a_2x^2+a_1x+a_0)$ be a monic irreducible polynomial. One intends to find multiplicative inverse of $b(x) = (b_2x^2+b_1x+b_0)$ under this I(x). If $c(x) = (c_2x^2+c_1x+c_0)$ be the multiplicative inverse then it can be written as,

$$[b(x)\ c(x)] \bmod I(x) = 1$$

or, $[(b_2x^2+b_1x+b_0)\ (c_2x^2+c_1x+c_0)] \bmod (x^3+a_2x^2+a_1x+a_0) = 1 \quad (1)$

Here the target is to find the values for c_0, c_1 and c_2. One can get the values by solving eq.(1) as follows:

$$[b_2c_2x(x^3+a_2x^2+a_1x+a_0) + (b_1c_2+b_2c_1-a_2b_2c_2)x^3 +$$
$$(b_0c_2+b_1c_1+b_2c_0-a_1b_2c_2)x^2 + (b_0c_1+b_1c_0-a_0b_2c_2)x +$$
$$b_0c_0] \bmod (x^3+ a_2x^2+a_1x+a_0) = 1$$

or, $[\{(a_2^2b_2-a_1b_2-a_2b_1+b_0)c_2+(b_1-a_2b_2)c_1+b_2c_0\}x^2 +$
$\{(a_1a_2b_2-a_0b_2-a_1b_1)c_2+(b_0-a_1b_2)c_1+b_1c_0\}x + \{(a_0a_2b_2-$
$a_0b_1)c_2-a_0b_2c_1+b_0c_0\}] \bmod (x^3+a_2x^2+a_1x+a_0) = 1 \quad (2)$

The eq.(2) is solved detail in [1]. One can get the k-matrix from this equation as,

$$k = \begin{bmatrix} k_{00} & k_{01} & k_{02} \\ k_{10} & k_{11} & k_{12} \\ k_{20} & k_{21} & k_{22} \end{bmatrix} \quad (3)$$

where k-values are known and these are equal to,

$k_{00} = (b_2) \%7$ (4a)

$k_{01} = (b_1+6a_2b_2) \%7$ (4b)

$k_{02} = (a_2^2b_2+6a_1b_2+6a_2b_1+b_0) \%7$ (4c)

$k_{10} = (b_1) \%7$ (4d)

$k_{11} = (b_0+6a_1b_2) \%7$ (4e)

$k_{12} = (a_1a_2b_2+6a_0b_2+6a_1b_1) \%7$ (4f)

$k_{20} = (b_0) \%7$ (4g)

$k_{21} = (6a_0b_2) \%7$ (4h)

$k_{22} = (a_0a_2b_2+6a_0b_1) \%7$ (4i)

Let the inverse of k-matrix be k^{-1} matrix which can be written as,

$$k^{-1} = \begin{bmatrix} ik_{00} & ik_{01} & ik_{02} \\ ik_{10} & ik_{11} & ik_{12} \\ ik_{20} & ik_{21} & ik_{22} \end{bmatrix} \quad (5)$$

While calculating k^{-1} from k-matrix, one has to ensure that the determinant det(k) is non-zero. In the event det(k) = 0, the I(x) is not an irreducible polynomial, rather a reducible one and k^{-1} matrix for such a case does not exist. If det(k) is non-zero for all elements, the I(x) is irreducible and the multiplicative inverses of elements exist. By calculating k^{-1} matrix, one can get values for c_0, c_1 and c_2 as,

$$c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} ik_{02} \\ ik_{12} \\ ik_{22} \end{bmatrix} \quad (6)$$

.

Hence,

$$(b_2x^2+b_1x+b_0)^{-1} = (c_2x^2+c_1x+c_0) \bmod (x^3+a_2x^2+a_1x+a_0) \quad (7)$$

2.2 Finding Multiplicative inverse of x^2+3x+5

An example of application of Algebraic method to calculate the multiplicative inverse of $x^2 + 3x + 5$ under irreducible polynomial $x^3 + 2$ over GF(7^3) is given below.

Let the irreducible polynomial $I(x) = x^3 + a_2x^2 + a_1x + a_0$

$= x^3 + 2$

The given polynomial $b(x) = b_2x^2 + b_1x + b_0$

$= x^2 + 3x + 5$

One has to find, $b(x)^{-1} = c(x) = c_2x^2 + c_1x + c_0$ (8)

Here, $a_2 = 0$, $a_1 = 0$, $a_0 = 2$

and $b_2 = 1$, $b_1 = 3$, $b_0 = 5$

By using these a and b values in eqs.(4a) through (4i) one can calculate the k-values as,

$k_{00} = 1\%7 = 1$ $k_{01} = 3\%7 = 3$ $k_{02} = 5\%7 = 5$

$k_{10} = 3\%7 = 3$ $k_{11} = 5\%7 = 5$ $k_{12} = 12\%7 = 5$

$k_{20} = 5\%7 = 5$ $k_{21} = 12\%7 = 5$ $k_{22} = 36\%7 = 1$

Using these values, the k-matrix shown in eq. (3), and its inverse matrix k^{-1} will be,

$$k = \begin{pmatrix} 1 & 3 & 5 \\ 3 & 5 & 5 \\ 5 & 5 & 1 \end{pmatrix} \quad \text{and} \quad k^{-1} = \begin{pmatrix} 5 & 5 & 6 \\ 5 & 6 & 1 \\ 6 & 1 & 1 \end{pmatrix}$$

The values for c_0, c_1 and c_2 will be obtained from the last column of the k^{-1} matrix as shown in eq. (6). Hence, the solution for this problem is,

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 1 \end{bmatrix}$$

So, one can obtain the required multiplicative inverse as,

$b(x)^{-1} = c(x) = c_2x^2 + c_1x + c_0 = x^2 + x + 6$

Hence, $(x^2 + 3x + 5)^{-1} = x^2 + x + 6$

2.3 Finding Multiplicative inverse of $3x^2+6$

Another example to calculate the multiplicative inverse of a given polynomial $3x^2 + 6$ under irreducible polynomial $x^3 + 2$ over GF(7^3) is shown here.

Let the irreducible polynomial $I(x) = x^3 + a_2x^2 + a_1x + a_0$

$= x^3 + 2$

The given polynomial $b(x) = b_2x^2 + b_1x + b_0$

$= 3x^2 + 6$

One has to find, $b(x)^{-1} = c(x) = c_2x^2 + c_1x + c_0$ (9)

Here, $a_2 = 0$, $a_1 = 0$, $a_0 = 2$

and $b_2 = 3$, $b_1 = 0$, $b_0 = 6$

By using these a and b values in eqs.(4a) through (4i) one can calculate the k-values as,

$k_{00} = 3\%7 = 3$ $k_{01} = 0\%7=0$ $k_{02} = 6\%7 = 6$

$k_{10} = 0\%7 = 0$ $k_{11} = 6\%7=6$ $k_{12} = 36\%7 = 1$

$k_{20} = 6\%7 = 6$ $k_{21} = 36\%7=1$ $k_{22} = 0\%7 = 0$

Using these values, the k-matrix shown in eq. (3), and its inverse matrix k^{-1} will be,

$$k = \begin{pmatrix} 3 & 0 & 6 \\ 0 & 6 & 1 \\ 6 & 1 & 0 \end{pmatrix} \quad \text{and} \quad k^{-1} = \begin{pmatrix} 4 & 4 & 4 \\ 4 & 4 & 5 \\ 4 & 5 & 5 \end{pmatrix}$$

The values for c_0, c_1 and c_2 will be obtained from the last column of the k^{-1} matrix as shown in eq. (6). Hence, the solution for this problem is,

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 5 \end{bmatrix}$$

So, one can obtain the required multiplicative inverse as,

$b(x)^{-1} = c(x) = c_2x^2 + c_1x + c_0 = 5x^2 + 5x + 4$

Hence, $(3x^2 + 6)^{-1} = 5x^2 + 5x + 4$

2.4 Algorithm to find all the Multiplicative Inverses under an IP over GF(7^3)

In this paper the first irreducible polynomial $x^3 + 2$ over GF(7^3) is used to calculate all its 342 multiplicative inverses. Considering only the coefficients, the polynomial $x^3 + 2$ is also represented as 1002. The algorithm of related program is given below:

Initialization: p = 7 and m = 3.

Take an Irreducible Polynomial (ip) in septenary number system and store them in an array a[] defined in eq.(1) where a_0 is the least significant septenary digit.

for ep = 1 to 342 **do**

Convert the ep into its septenary equivalent and store them in an array b[] defined in eq.(1) where b_0 is the least significant septenary digit.

From arrays a[] and b[] form the 3×3 k-matrix described in eq.(3).

Calculate the determinant of k-matrix and its inverse as k^{-1}-matrix. Let the elements of k^{-1}-matrix be ik_{00} to ik_{22}.

Assign the c coefficients as $c_0=ik_{02}$, $c_1=ik_{12}$, $c_2=ik_{22}$.

Store the multiplicative inverse in c[] as $c_2c_1c_0$.

End of **for**

Computational results of the 342 multiplicative inverses under the first irreducible polynomial $(1002)_7$ are shown in Table 1 in septenary number system. The whole list of multiplicative inverses for all the 112 irreducible Polynomials over GF(7^3) is available online in academia.edu [13].

3. RC4 ALGORITHM AND A NEW PRNG USING IRREDUCIBLE POLYNOMIAL

In this paper a new PRNG is presented which depends on irreducible polynomial and adopts the shuffling technique like well known RC4 algorithm. An overview of the RC4 algorithm is given in Sec. 3.1 and design procedure of the new PRNG is described in Sec. 3.2.

3.1 Overview of RC4 algorithm

Ron Rivest translated the shuffling concept in two stages, Key Scheduling Algorithm (KSA) and Pseudo Random Generator Algorithm (PRGA). Design procedure of 256-byte state vector S which is used as the key-pool in RC4 is very simple. And let G be the given key of length *keylen*. In KSA, a K-Box of 256 bytes is created from given key and the S-Box is organized as follow:

The KSA stage

Initialization of S vector:

for i = 0 to 255

S[i] = i;

Generation of K vector:

for i = 0 to 255

K[i] = G[i **mod** *keylen*]

Permutation of S vector (key mixing):

j = 0;

for i = 0 to 255 **do**

j = (j + S[i] + K[i]) **mod** 256;

Swap (S[i], S[j]);

After permutation of S vector, role of the given key is end. Only the S vector is used in PRGA to provide a sequence of key stream Z as follow:

The PRGA stage
i = j = 0;
while (true)
    i = (i + 1) **mod** 256;
    j = (j + S[i]) **mod** 256;
    Swap (S[i], S[j]);
    t = (S[i] + S[j]) **mod** 256;
        Z = S[t];

Table 1. The 342 Multiplicative Inverses under the first Irreducible Polynomial $x^3 + 2$ over GF(7^3)

001, 004, 005, 002, 003, 006, 300, 616, 623, 214, 645, 241, 222, 500, 326, 343, 124, 315, 111, 142, 100, 365, 356, 252, 333, 264, 231, 600, 546, 513, 444, 525, 421, 412, 200, 635, 666, 462, 653, 434, 451, 400, 555, 536, 132, 563, 154, 161, 030, 413, 115, 643, 216, 346, 545, 461, 025, 636, 165, 363, 102, 135, 262, 335, 566, 433, 023, 463, 404, 521, 240, 063, 215, 604, 116, 310, 164, 665, 026, 201, 533, 236, 266, 324, 502, 440, 213, 065, 610, 416, 622, 066, 301, 510, 140, 113, 415, 050, 143, 245, 523, 446, 626, 325, 432, 655, 336, 153, 013, 133, 104, 234, 535, 016, 401, 353, 456, 436, 512, 036, 601, 340, 220, 243, 145, 131, 015, 556, 235, 633, 202, 255, 311, 420, 033, 445, 504, 246, 640, 614, 302, 120, 443, 035, 540, 146, 010, 162, 261, 651, 464, 352, 554, 136, 250, 351, 403, 652, 024, 660, 435, 552, 354, 360, 150, 206, 021, 625, 503, 524, 034, 514, 121, 212, 233, 551, 450, 022, 654, 560, 105, 526, 312, 305, 224, 322, 411, 032, 323, 621, 611, 114, 606, 031, 422, 060, 223, 425, 313, 126, 516, 615, 454, 355, 046, 101, 663, 166, 156, 251, 045, 366, 455, 553, 402, 465, 544, 602, 210, 123, 055, 320, 226, 152, 565, 656, 263, 043, 253, 204, 342, 056, 501, 620, 410, 423, 225, 641, 110, 053, 125, 304, 426, 520, 020, 452, 151, 331, 254, 532, 634, 163, 631, 230, 042, 334, 650, 405, 466, 130, 531, 203, 332, 044, 350, 646, 522, 505, 144, 542, 221, 062, 265, 632, 534, 550, 430, 106, 041, 543, 341, 321, 424, 306, 061, 242, 345, 603, 644, 064, 624, 441, 122, 040, 232, 431, 561, 134, 662, 364, 155, 362, 664, 630, 260, 406, 011, 453, 361, 160, 012, 564, 330, 205, 613, 511, 541, 244, 506, 051, 112, 256, 460, 661, 103, 562, 014, 530, 515, 303, 314, 054, 344, 211, 442, 316, 642, 605, 414, 612, 141, 052.

3.2 Design of a new PRNG using Irreducible Polynomial $x^3 + 2$ over GF(7^3)

As an application of irreducible polynomial in Pseudorandom Number Generator (PRNG) and following the idea of random shuffling of Knuth [10] a new PRNG is proposed in which the multiplicative inverses under the first irreducible polynomial $x^3 + 2$ over GF(7^3) are used in S-Box and part of K-Box of RC4. The proposed PRNG is called as GF7 in which the initial identity S-Box[256] of RC4 is replaced by an S-Box[256] which first takes zero in its 0^{th} index and then sequentially puts decimal equivalent of 255 values less than $(514)_7$ shown in Table 1. The decimal equivalent of other 87 values greater than $(513)_7$ available in Table 1 are complemented in bit level and sequentially put in a K-Box[256] as the first initial 87 entries. The rest 169 spaces of the K-Box are filled by the given key following RC4 algorithm. The initial S-Box proposed here are better shuffled than the identity S-Box used in RC4 and the organization of the K-Box indicates that the initial pseudorandom bytes created by the proposed generator GF7 would not carry the bias of the given key – rather these would carry the influence of irreducible polynomial. The rest part of GF7 algorithm is identical to the RC4 algorithm.

Merit of GF7: Following two points indicate the strength of GF7 algorithm.

1. The initial S-Box of GF7 is dependent on a mathematical function and it will be changed if the irreducible polynomial is altered. Hence the security of the generator will be increased if the irreducible polynomial is kept secret between the sender and the receiver.
2. The K-Box is partly dependent on the mathematical function due to which the key-bias of RC4 is removed in GF7.

It is intended to compare the random bytes produced by GF7 with that produced by RC4 from statistical randomness perspective. The Statistical Randomness Tests are undertaken on the two algorithms – their data are shown in Sec. 4.

4. STATISTICAL RANDOMNESS TESTS AND DATA OBTAINED FOR GF7 AND RC4

The motivation of statistical test is to check randomness of the proposed pseudorandom number generator GF7 based on NIST statistical test suite [14–17]. Calculation technique is presented in NIST publications [14,16] where the χ^2-value coupled with the degrees of freedom is transformed to a P-value instead of computing the probability value using the χ^2-distribution function only. Thereby it sets a passing criterion; P-value ≥ 0.01 (significance level). Using all the P-values obtained for a particular test, NIST also mentioned a statistical procedure to compute Proportion of Passing, which indicates uniformity or non-uniformity of P-values. The NIST test suite consists of fifteen statistical tests which are well reviewed recently [18]. The minimum length of a bit-sequence required for any particular test as recommended by NIST is given in Table 2.

It may be noted that 300 different 16-character key are used for the two algorithms GF7 and RC4; and each algorithm generates 300 different bit-sequences each of bit-length more than 1342400 bits.

In estimating the degree of randomness of an algorithm, the Threshold value (T-value) and P-value of P-values (POP) are considered as two important checking parameters. These are explained in sub-sections 4.1 and 4.2 respectively. The results of the two algorithms GF7 and RC4 are compared and presented in Sec. 5.

4.1 Computation of T-value: Observed Proportion Of Passing (OPOP)

To estimate the Observed Proportion Of Passing (OPOP) of a particular test, it is necessary to consider large number of

samples of bit-sequences randomly generated by an algorithm. If m samples of bit-sequences obtained from an algorithm are tested by a test producing one P-value, then the statistical average of T-value would be,

$$T_{value} = (1-\alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{m}} \qquad (10)$$

Here significance level (α) = 0.01. The size of m should greater than inverse of α. If m = 300, T-value = 0.972766. This means that such a test is considered statistically successful, if at least 292 P-values out of the 300 P-values do pass the test. If any test produced n number of P-values, then to calculate T-value in equation (10), one should consider m×n instead of m. With same values of α and m, the T-value is 0.983907 for n = 8 (test no. 14 in Table 2). Such a test is considered statistically successful if at least 2362 P-values out of the total 300×8 = 2400 P-values do pass the test. The status for proportion of passing a particular test would be a success if OPOP is greater than the corresponding T-value.

Table 2. Minimum required lengths and used lengths of bit-sequence for 15 statistical tests

Test No.	Name of the Test	Length of bit-sequence (n)	
		Minimum require-ment	Used in present software
1	Frequency Test	100	1342400
2	Frequency Test within a Block	9,000	1342400
3	Runs Test	100	1342400
4	Longest Run of Ones in a Block	128	1342400
5	Binary Matrix Rank Test	38,912	1342400
6	Discrete Fourier Transform Test	1,000	13424
7	Non-overlapping Template Test	10,48,576	1342400
8	Overlapping Template Test	10,00,000	1342400
9	Maurer's "Universal Statistical" Test	13,42,400	1342400
10	Linear Complexity Test	10,00,000	1342400
11	Serial Test	10,00,000	1342400
12	Approximate Entropy Test	100	1342400
13	Cumulative Sums (Cusum) Test	100	13424
14	Random Excursions Test	10,00,000	1342400
15	Random Excursions Variant Test	10,00,000	1342400

4.2 Computation of P-value Of P-values (POP): Distribution pattern of P-values

One can have an understanding about uniformity of distribution of P-values for a particular test from the series of obtained P-values. The P-values for a particular test are noted in 11 sub-intervals between 0 and 1 in Table 3(A) and Table 3(B).

Table 3(A). Frequency distribution of P-values of RC4

Test No.	0.0-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	2	37	24	33	41	33	21	24	34	29	22
2	2	30	29	30	20	22	28	28	41	33	37
3	4	25	24	29	36	29	29	23	36	36	29
4	2	21	36	29	29	25	29	31	34	32	32
5	2	19	34	26	26	30	40	30	38	28	27
6	2	33	36	30	28	32	23	24	39	27	26
7	1	22	32	37	30	30	32	33	30	30	23
8	4	21	28	34	36	31	25	33	28	28	32
9	3	31	35	32	27	30	26	35	27	24	30
10	2	24	23	26	33	28	38	30	26	31	39
11	8	58	68	52	72	63	58	57	58	60	46
12	3	33	29	25	42	36	23	28	29	33	19
13	7	62	52	60	57	62	52	68	66	50	64
14	20	211	213	235	248	246	235	236	249	274	233
15	45	419	544	522	541	535	584	559	540	551	560

Table 3(B). Frequency distribution of P-values of GF7

Test No.	0.0-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	2	26	24	31	29	29	29	31	34	36	29
2	3	23	33	35	35	32	23	27	37	21	31
3	3	37	31	27	26	27	30	25	33	34	27
4	3	25	29	29	34	35	28	29	23	27	38
5	2	30	26	31	23	22	31	37	23	39	36
6	5	30	39	24	20	28	30	39	35	24	26
7	3	26	36	29	27	34	13	32	35	34	31
8	5	31	24	31	23	30	35	35	24	33	29
9	3	23	38	34	29	32	23	31	30	30	27
10	6	27	18	31	34	35	28	30	36	23	32
11	7	52	52	55	75	54	55	56	60	64	70
12	1	32	26	31	32	24	27	31	32	31	33
13	7	63	59	65	51	66	67	49	60	67	46
14	26	220	242	227	240	247	231	255	205	245	262
15	74	443	527	531	521	523	550	563	563	532	573

For estimating χ^2-deviation of distribution of P-values, the first two groups of P-values are merged in one group and the rest in 9 groups, thereby considering 10 groups of P-values. The χ^2-deviation of distribution of P-values is computed as,

$$\chi^2 = \sum_{i=1}^{10} \frac{\left(s_i - \frac{m}{10}\right)^2}{m/10} \qquad (11)$$

where, S_i is the number of P-values in a group i, and m is the sample size. If a particular test produces n number of P-values, then m = n × sample size. Here the degrees of freedom v = 9. The two parameters in the gamma function, $\Gamma(a,x)$ are taken as, a = v/2 and x = χ^2/2 and the corresponding POP is obtained as,

$$POP = 1 - \frac{\Gamma(a,x)}{\Gamma(a,\infty)} \qquad (12)$$

The P-values are considered as uniformly distributed if eq.(12) gives POP ≥ 0.0001.

5. RESULTS: COMPARATIVE STUDY OF RC4 AND GF7

The test-wise results of statistical tests on RC4 are given in Table 3(A) and that for GF7, in Table 3(B). The P-value data are divided in 11 groups between 0 and 1. Depending on the result of P-value for a particular test, the count of an appropriate group is increased in which the particular P-value belongs. If the P-value < 0.01, then it will be considered as unsuccessful, the entry in column 1 indicates the numbers of unsuccessful P-value. The OPOP for a particular test is the ratio of sum of the last ten columns to the total sum of eleven columns. It is compared with the T-value calculated in eq.(10) to see whether the data set is statistically random or not. The POP of a particular test is also derived from Tables 3(A) and 3(B) using eq.(12). The distribution is considered to be uniform if POP ≥ 0.0001.

Following the procedure stated above, the test-wise OPOP and POP data for the algorithms RC4 and GF7 are calculated for all the fifteen tests. The OPOP data along with the T-value are shown in Table 4(A). It has been observed that both the algorithms pass all the tests and exhibit uniform distribution of P-values.

Table 4(A). Test-wise Observed Proportion Of Passing (OPOP) for RC4 and GF7

Test No.	T-value	Observed Proportion Of Passing (OPOP)	
		RC4	GF7
1	0.972766	0.993333	0.993333
2	0.972766	0.993333	0.990000
3	0.972766	0.986667	0.990000
4	0.972766	0.993333	0.990000
5	0.972766	0.993333	0.993333
6	0.972766	0.993333	0.983333
7	0.972766	0.996667	0.990000
8	0.972766	0.986667	0.983333
9	0.972766	0.990000	0.990000
10	0.972766	0.993333	0.980000
11	0.977814	0.986667	0.988333
12	0.972766	0.990000	0.996667
13	0.977814	0.988333	0.988333
14	0.983907	0.991667	0.989167
15	0.985938	0.991667	0.986296

The POP results calculated for RC4 and GF7 are presented in Table 4(B). It has been observed that both the algorithms exhibit uniform distribution of P-values. Regarding the uniformity of distribution of P-values, one can correlate the POP value shown in Table 4(B) with a corresponding visual histogram obtained from the right data of Table 3(A) or of Table 3(B). It is evident from Table 4(B) that, in GF7 the POP value for all the fifteen tests are in the order of 10^{-1} while in RC4 only thirteen tests have P-value within that order and the rest two is in the order of 10^{-2}. This indicates that GF7 produces more uniform data than RC4.

Table 4(B). Test-wise P-value Of P-values (POP) for RC4 and GF7

Test No.	P-value of P-values (POP)	
	RC4	GF7
1	8.733803e-02	9.527777e-01
2	2.209308e-01	4.434507e-01
3	6.786858e-01	6.924550e-01
4	8.676917e-01	7.597563e-01
5	3.345381e-01	2.327599e-01
6	4.685950e-01	1.426018e-01
7	7.918798e-01	1.509064e-01
8	8.930010e-01	6.093772e-01
9	8.623444e-01	8.043368e-01
10	4.814161e-01	3.838266e-01
11	4.527993e-01	4.878851e-01
12	1.153866e-01	9.642950e-01
13	6.059162e-01	2.780007e-01
14	3.990028e-01	3.859730e-01
15	5.549155e-02	6.448315e-01

From Table 4(B), test 12 of GF7 is seen as the best POP obtained from test 12 data of Table 3(B) – the same data is displayed in a corresponding histogram in Fig. 1(a). The uniformity of P-value distribution is visually evident. The histograms for worst uniformity of GF7 (vide test no. 6), best uniformity of RC4 (vide test no. 8) and worst uniformity of RC4 (vide test no. 15) are also shown in Fig. 1(b), 2(a) and 2(b) respectively. In all the histograms, there are ten columns: first column indicates the number of P-values lying between 0 and 0.1; second column indicates the number of P-values lying between 0.1 and 0.2, so on and so forth. It may be noted that histogram shown in Fig. 2(b) is based on 5400 P-values for 300 sequences (18 P-values in 1 sequence) while each of three other histograms have 300 P-values for 300 sequences (1 P-value in 1 sequence). For paucity of space in Fig. 2(b), the number of P-values above 400 is shown in the ordinate axis.

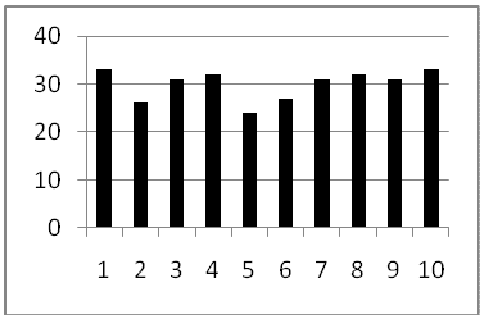

Fig 1(a): Histogram for Test no. 12 of GF7 (POP: 9.642950e-01)

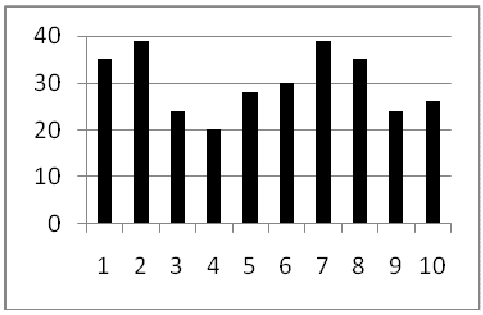

Fig 1(b): Histogram for Test no. 6 of GF7 (POP: 1.426018e-01)

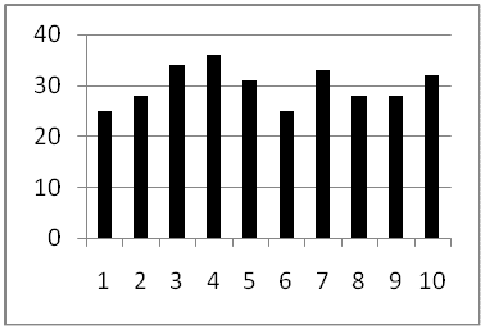

Fig 2(a): Histogram for Test no. 8 of RC4 (POP: 8.930010e-01)

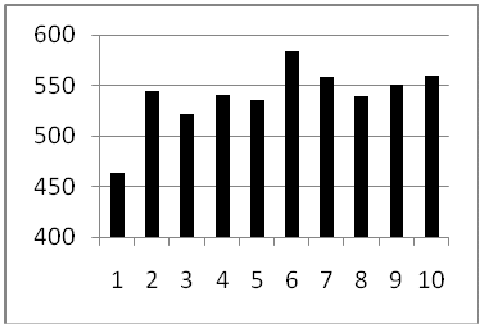

Fig 2(b): Histogram for Test no. 15 of RC4 (POP: 5.549155e-02)

6. CONCLUSION

A new pseudorandom number generator GF7 is proposed in the paper where the multiplicative inverses of 342 elemental polynomials under an irreducible polynomial over GF(7^3) is innovatively introduced. Both the initial S-Box and K-Box of GF7 depend on the irreducible polynomial which may be changed in various applications and the new algorithm does not have key bias in its initial random bytes unlike RC4. Thus keeping the irreducible polynomial secret between the sender and the receiver, security of the series of output sequence will be increased. The comparative study on the NIST data of GF7 and RC4 indicates that, from OPOP point of view GF7 is comparable with RC4, but from POP point of view GF7 is more uniform than RC4 as all the fifteen POP-values are in order of 10^{-1} while in RC4 only thirteen POP-values are of that order.

7. ACKNOWLEDGMENTS

We express our gratitude to the Principal of A.P.C. Roy Govt. College, Siliguri, Govt. of West Bengal for providing his moral support and infrastructural facilities towards research to the first author. We are also indeed thankful to the Head of the Department of Radio Physics and Electronics, University of Calcutta for providing necessary facilities to undertake research activities.

8. REFERENCES

[1] Zaman, J K M S., Ghosh, R., Multiplicative Polynomial Inverse over GF(7^3): Crisis of EEA and its Solution, Applied Computation and Security Systems, Springer, vol. 2, 87-107 (2014).

[2] Stallings, W., “Finite Fields,” in Cryptography and Network Security Principles and Practices, 4th ed., Delhi: Pearson Education, 95–133 (2008).

[3] Forouzan, B.A., Mukhopadhyay, D., “Mathematics of Cryptography,” in Cryptography and Network Security, 2nd ed., New Delhi: TMH, 15–43 (2011).

[4] Roos, A., "A class of weak keys in the RC4 stream cipher ", Sep. 1995.

[5] S. Fluhrer, I. Mantin, A. Shamir, “Weakness in the Key Scheduling Algorithm of RC4,” in Proc. Int. Workshop on Selected Areas in Cryptography, Berlin Heidelberg, LNCS 2259, 1–24 (2001).

[6] Paul, B., Preneel, “A New Weakness in the RC4 Keystream. Generator and an Approach to Improve the Security of the Cipher,” in Proc. Fast Software Encryption, Berlin, LNCS 3017, 245–259 (2004).

[7] Maitra, S., Paul, G., “Analysis of RC4 and Proposal of Additional Layers for Better Security Margin” , in Proc. Indocrypt, IIT Kharagpur, LNCS 5365, 27–39 (2008).

[8] Paul, G., Maitra, S., "RC4 Stream Cipher and its Variants," Boca Raton, Chapman & Hall/CRC (2012).

[9] Stinson, D.R., “The RSA Cryptosystem and Factoring Integers,” in Cryptography Theory and Practice, 3rd ed., Boca Raton: Chapman & Hall/CRC, 161–232 (2006).

[10] Knuth, D.E.: The Art of Computer Programming Seminumerical Algorithms, 3rd edn, vol. 2. Pearson Education, Upper Saddle River (2011).

[11] Church R., “Tables of Irreducible Polynomials for the first four Prime Moduli”, Annals of Mathematics, Vol. 36(1), pp. 198-209, January, 1935.

[12] Lidl R., Niederreiter H., Finite Fields, Encyclopedia of Mathematics and its Applications, Vol. 20, Addison-Wesley Publishing Company, 1983.

[13] https://www.academia.edu/6042515/ “Multiplicative Inverses of all the 342 Elemental Polynomials (EP) for all the 112 Irreducible Polynomials (IP) Over GF(7^3)”.

[14] Rukhin A., Soto J., et al, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST, US, Technology Administration, U.S. Department of Commerce (2010).

[15] http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf

[16] Rukhin A., Soto J., et al, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST, Technology Administration, U.S. Department of Commerce (2008).

[17] http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html

[18] Zaman, J K M S., Ghosh, R., Review on fifteen Statistical Tests proposed by NIST, Int. J. Theoretical Physics & Cryptography, 1, 18-31 (2012).